UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial

TESIS DOCTORAL

Confianza de grupo en sistemas distribuidos y sus relaciones con la seguridad de la información y la ciberseguridad

Group trust in distributed systems and its relationship with information security and cyber security

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Robson de Oliveira Alburquerque

Directores

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Madrid, 2016

Confianza de Grupo en Sistemas Distribuidos y sus Relaciones con la Seguridad de la Información y la Ciberseguridad

– – – – – – – – – – – – – – – – – – – –

Group Trust in Distributed Systems and its Relationship with Information Security and Cyber Security

Thesis by

Robson de Oliveira Albuquerque

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisors

Luis Javier García Villalba
Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, November 2015

Group Trust in Distributed Systems and its Relationship with Information Security and Cyber Security

Thesis by

Robson de Oliveira Albuquerque

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisors

Luis Javier García Villalba
Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, November 2015

Confianza de Grupo en Sistemas Distribuidos y sus Relaciones con la Seguridad de la Información y la Ciberseguridad

TESIS DOCTORAL

Memoria presentada para obtener el título de
Doctor por la Universidad Complutense de Madrid
en el Programa de Doctorado en Ingeniería Informática

Robson de Oliveira Albuquerque

Dirigida por:

Luis Javier García Villalba
Ana Lucila Sandoval Orozco

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Noviembre de 2015

Dissertation submitted by Robson de Oliveira Albuquerque to the *Departamento de Ingeniería del Software e Inteligencia Artificial* of the *Universidad Complutense de Madrid* in Partial Fulfillment of the Requirements for the Degree of *Doctor por la Universidad Complutense de Madrid en el Programa de Doctorado en Ingeniería Informática*.

Madrid, 2015.

(Submitted November 1, 2015)

Title:

Group Trust in Distributed Systems and its Relationship with Information Security and Cyber Security

PhD Student:

Robson de Oliveira Albuquerque (robson@fdi.ucm.es)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain

Advisors:

Luis Javier García Villalba (javiergv@fdi.ucm.es)
Ana Lucila Sandoval Orozco (asandoval@fdi.ucm.es)

This work has been done within the Group of Analysis, Security and Systems (GASS, http://gass.ucm.es), Research Group 910623 from the Universidad Complutense de Madrid (UCM) as part of the activities of different research projects. This research has been supported by the Ministerio de Defensa (MDE, Spain) through project UCM 321/2011, by the Agencia Española de Cooperación Internacional para el Desarrollo (AECID) of the Ministerio de Asuntos Exteriores y de Cooperación (MAEC, Spain) through project A1/037528/11 and by Safelayer Secure Communications S. A. through project UCM 307/2013, thanks to which part of this work was done during my stay in France at Supélec of Rennes.

This thesis is dedicated to my Family:
My Father - Eloi Albuquerque;
My Mother - Maria Feleciana;
My Wife - Sheyla;
My Daughters - Mariana, Maria Eduarda, Ana Clara and Antonella;
My Brothers - Amauri and Elon.

Acknowledgments

First, I would like to thank my supervisors, Luis Javier García Villalba and Ana Lucila Sandoval Orozco. Without their support, this work would not have been possible. Thank you very much for everything you did to make this come true.

Many thanks to the support of my family. Without them I could not have come this far. Dad and Mom, thanks for all the opportunities you have provided me.

Thanks to my dear wife Sheyla. Words cannot fully translate my feelings towards you. I just can say here thank you for your time and patience. I love you.

Thanks to my daughters. Mariana, you have become a lovely lady and very special to me. Maria Eduarda, you are beautiful and full of life. Ana Clara, you have the most beautiful hair I can image and a lovely smile. Antonella, you are so adorable and clever. I am very proud of being your Dad. I love you all. Thanks for making my life happier.

Thanks to my Brothers. You guys are terrific. Let's keep going forward.

Of course many many thanks to my friends. This work also has the support of them.

I also would like to say thank you to all the members of the GASS research group and to the Vicerrectorado de Innovación de la Universidad Complutense de Madrid for the all the facilities offered.

This research has been supported by the Ministerio de Defensa (MDE, Spain) through project UCM 321/2011, by the Agencia Española de Cooperación Internacional para el Desarrollo (AECID) of the Ministerio de Asuntos Exteriores y de Cooperación (MAEC, Spain) through project A1/037528/11 and by Safelayer Secure Communications S. A. through project UCM 307/2013, thanks to which part of this work was done during my stay in France at Supélec of Rennes.

Contents

List of Figures

List of Tables

Abstract

This thesis describes aspects regarding trust, reputation, information security and cyber security as connected subjects.

It is the belief of this research that without trust it is not possible to address security in computational systems properly. One fundamental aspect is to use information systems or to rely on them. One must trust the technology involved and consequently the entire system, without even knowing it in the first place. Due to human characteristics, there is the tendency to trust that systems will keep our data secure. However, one basic problem is that trust and reputation deals with subjective evaluations.

In many situations, information systems nowadays have distributed support. It means that there are a lot of parts that connect everything together, which is unknown to the ordinary users. Depending on the technology, it is even unknown to systems administrators when it comes to for example clusters, cloud and peer-to-peer systems.

Considering the first area of research - trust - from the perspective of this work, a group trust model for distributed systems is presented as an extension applied to conventional trust and reputation mechanisms, extension developed considering groups, which is herein defined as a collection of entities with particular affinities and capabilities. Broadening this perspective, the formation of groups is very common, but very few trust and reputation models studied deal with trust in the perspective of a collection of entities with common affinities. Thus, group trust is a way of representing the set of trust and reputation of their particular members. One aspect to be aware of is the fact that this set has pre-defined activities and common objectives.

In the second area of research - information security - this thesis also discusses that information, security and trust need to be considered from different points of view in order to protect information properly. From this perspective, this work additionally tries to aggregate value in the security information area with discussions that suggest there is the need to extend information security beyond confidentiality, integrity and availability in order to increase security to computational systems. In this sense, this work develops an architecture, which combines the view of fundamental areas in information security with trust as strong connected elements. The proposed architecture is divided into layers representing important parts of information security and its extensions as a way of going further than confidentiality, integrity and availability. After describing the components that are part of the architecture and its corresponding functions, the trust layer is presented. The trust layer ties together the view of information security with trust.

The third area of this research - cyber security - is seen as the application of the reviewed concepts and establishing trust and information security. It is clear that cyber security has an enormous impact on modern society. In fact, the cyberspace is considered as a critical infrastructure to many countries. Cyber security depends on the combined effect of information security measures together with explicit trust verification that these measures are operational and effective. In this sense, this thesis provides a view of information treatments related to trust and information security and discusses

how together they can counter advanced persistent threats and exploits that now plague the cyberspace.

Keywords: Cyber Security, Distributed Systems, Group Trust, Information Security, Reputation, Trust.

Resumen

Esta tesis analiza aspectos relativos a confianza, reputación, seguridad de la información y ciberseguridad, considerándolos elementos interrelacionados.

Este trabajo parte de la premisa de que sin confianza no es posible garantizar la seguridad de los sistemas computacionales de una manera correcta y eficiente. Para utilizar los sistemas de información es necesario confiar en la tecnología subyacente. Consecuentemente, toda la cadena productiva que forma parte de estos sistemas acaba siendo objeto de confianza, muchas veces sin saberlo. Por defecto se tiende a confiar en que los sistemas mantendrán los datos seguros. Sin embargo, el hecho de que la confianza y la reputación se basen en evaluaciones subjetivas es una fuente importante de problemas.

Esta tesis aborda la problemática de la confianza en grupos en sistemas computacionales, entendiendo por grupo una colección de entidades con afinidades y capacidades particulares y entendiendo por confianza en grupo la confianza y la reputación de sus miembros particulares. Un aspecto a tener en cuenta es el hecho de que este conjunto tiene actividades y objetivos comunes previamente definidos. En la literatura apenas existen modelos de confianza y reputación desarrollados para este contexto.

En multitud de situaciones los sistemas de información utilizados tienen una naturaleza distribuida. Esto implica la existencia de una gran cantidad de componentes interconectados, desconocidos para la mayoría de usuarios y, dependiendo de la tecnología, también desconocidos para los administradores de sistemas. Un claro ejemplo de este tipo de sistemas son los sistemas en la nube.

En primer lugar esta tesis propone un modelo de confianza y reputación basado en grupos para sistemas computacionales en ambientes distribuidos. Este modelo de confianza y reputación en grupos surge como una extensión de los mecanismos de confianza y reputación convencionales.

En segundo lugar este trabajo analiza que la seguridad de la información y la confianza deben ser consideradas conjuntamente con el fin de proteger la información correctamente. En este sentido esta tesis sostiene que es necesario agregar a la seguridad de la información un valor añadido como es la confianza, extendiendo así la seguridad de la información más allá de sus premisas básicas de confidencialidad, integridad y disponibilidad, para garantizar la seguridad de los sistemas computacionales. Esta tesis desarrolla una arquitectura que combina las premisas básicas de seguridad de la información con la confianza como elementos estrechamente relacionados. La arquitectura propuesta se divide en diversas capas que representan los diferentes elementos de la seguridad de la información y sus extensiones, presentándose la confianza como una capa adicional que interconecta a todas las anteriores.

Finalmente, este trabajo relaciona la seguridad de la información y la confianza con la ciberseguridad, aplicando la arquitectura multinivel de seguridad desarrollada anteriormente en el área de la ciberseguridad. Es ampliamente conocido que la ciberseguridad tiene un gran impacto en la sociedad moderna. De hecho, el ciberespacio es considerado una infraestructura crítica en la mayoría de los países, cobrando cada día más importancia. Elementos como exploits, amenazas persistentes avanzadas, etc., son

una constante diaria en Internet, representando amenazas reales para la seguridad de la información en el ciberespacio. Este trabajo plantea un marco para contrarrestarlas que hace uso de la citada arquitectura.

Palabras clave: Ciberseguridad, Confianza, Confianza en Grupos, Seguridad de la Información, Sistemas Distribuidos, Reputación.

List of Acronyms

APT	Advanced Persistent Threat
CIA	Confidentiality, Integrity and Availability
COBIT	Control Objectives for Information and Related Technology
CS	Computational System
DHT	Distributed Hash Table
ENISA	European Union Agency for Network and Information Security
FIRE	Fides and Reputation
GTRUST	Group Trust
HABIT	Hierarchical And Bayesian Inferred Trust
I2P	Invisible Internet Project
ICS	Industrial Control Systems
IoT	Internet of Things
ISMS	Information Security Management Systems
ITIL	Information Technology Infrastructure Library
MAS	Multi-Agent Systems
NATO	North Atlantic Treaty Organization
NIST	National Institute of Standards and Technology
P2P	Peer-to-Peer
PoC	Proof of Concept
QoS	Quality of Service
REGRET	Reputation Model for Gregarious Societies
RRAF	Reliability and Reputation Agent Framework
SCADA	Supervisory Control and Data Acquisition
SIEM	Security Information and Event Management
SLA	Service Level Agreement

TISA	Trust Information Security Architecture
TOR	The Onion Router
TRR	Trust-Reliability-Reputation
VPN	Virtual Private Network
XaaS	Anything as a Service
XML	Extensible Markup Language

Part I

Description of the Research

Chapter 1

Introduction

Trust in computation systems has been evidenced for a long time since the advances in informatics. Normally, people and organizations trust their governments or information systems to provide a safe and secure environment for them to carry out their business and personal lives. It does not matter if it is online or in the real world. Citizens trust that their data is protected from misuse by unauthorized parts. Most of it relies on laws that guarantee privacy and the vision that the justice will punish those who abuse it.

In the online world, people trust the technology they use. Of course, much of what is behind such technologies is unknown to most users. Concerns comes when there is the lack of trust in systems and those who are responsible for its maintenance. Problems of privacy are also major problem in such cases. Considering all these, there are a lot of hacks that show trust in the technology used to protect information systems has been misplaced. For instance, messing with Certificate Authorities undermines some foundations of secure online communications.

Considering the modern society, information is an important asset of any organization. Normally, in order to secure such assets involves assuring Confidentiality, Integrity and Availability (CIA) triad. But there is a lot of discussion in the security information field nowadays reasoning if only CIA is sufficient to guarantee security in technological systems. This research believes it is not. In order to discuss further issues, a trust approach to computational systems is proposed in this thesis and it also proposes what this work defined as Trust Information Security Architecture (TISA).

In information security field, a large number of vulnerabilities are seen, which affect a variety of systems and widely used services. Manufacturers and developers work hard in order to release, as fast as possible, patches that fix such vulnerabilities. But, on the other hand, there is a community, which is continually growing, producing malicious codes, exploits and malware that take advantage of known and unknown vulnerabilities in a short period of time.

Regarding trust, information security in computational systems still has many problems. In order to provide contribution to de discussions in the field, this work proposes a trust model for distributed systems based on groups of peers. It is important to understand that most trust models deals with a perspective of one-to-one (1:1) trust. This works extents this view with one-to-many(1:N) and many-to-many (M:N) approach

of trust. In a communication process, entities may have a trust and a reputation value of each other in the system. But, in many cases it may be necessary to trust the whole system instead of one particular entity. From the perspective of this work, group trust represents the trust of their particular members, so an entity that is not part of group, can decide to trust the group or not based on group trust information.

Many authors ([BMG02], [MHAH08], [YLDZ14], [LC12]) understand that information security is a risk management job. This research adds to the problem that current information security platforms do not deal with the different facets of information technology considering trust. There is a growing need to consider information and security from different points of view in order to protect it. Without the correct understanding of how information is organized, accessed, processed and stored, it is not possible to guarantee that it is secure from data theft, misuse by unauthorized parties or even from being destroyed.

Inside information technology infrastructure environments there are things such as virtualization, provisioning, cloud technologies, distributed processing, parallel processing, software-defined networks, internet of things, and much more. Still, we see the growing of information leaks, user data exposed, nation-state sponsored actions, basic missteps in information security. It seems that the more technological the world becomes, the more information security problems happens. It is the belief of this work that it is due to the lack of the correct understanding of information security and trust.

It is a common comprehension that distributed systems have become very complex environments. Hundreds of nodes have to collaborate in order to provide large-scale services and trust between entities becomes a crucial factor as a way of determining the reliability of the different nodes in the system, for example, to be able to detect and predict misbehaviors and security threats.

Internet has so many different types of things connected to it that, up to now, there is no known way of dealing correctly with the amount of information or data that is necessary to verify if systems are fully reliable. As seen in Verzion Data Breach Report [Ver14], for example, Security Information and Event Management (SIEM) systems are not trustworthy when it comes to protecting network computer systems. This was also discussed and presented by Yoran [Yor15] as a keynote section at RSA conference in 2015. This is just the opposite of what part of information security professionals tend to believe.

What normally is in question is that the amount of security threats increases because the complexity, distribution, channels, data, networks also increase, which makes the traditional security approach inefficient. This points the technology and researches towards new manageability and security challenges. According to this particular view, Saied *et al.*[SOZL13] considers that new trust management systems are necessary in order to accomplish security related to distributed system considering the amount and variety of connected systems.

It is also important to point out that management of trust relationships between different peers belonging to a distributed system can be done using different approaches. It can be manually established by each node in the network towards the rest of the nodes or it can be automatically calculated. For example, a model that provides data of how

much a node can trust other nodes using trust and reputation.

Typically, a given node follows a trust model to determine if a node is trustworthy or not. It relies on two different aspects: i) Trust values are locally calculated; and ii) trust values are provided by the rest of the nodes in the network (reputation). Considering a trust and reputation perspective, it is important to observe that when emphasis is given to trust, it may have different lines of understanding because of its subjective evaluation (social capabilities, time influence, context and behavior needs).

It is also important to consider further connections about information security and trust. Most of the literature reviewed about the relations of trust and information security points out small considerations about them. In order to fulfill this gap, an architecture was created to provide deeper relations in the discussion regarding trust and its importance when it comes to information security.

Furthermore, this work discusses three connected fields: trust, cyber security and information security. In cyber security, which is a subset of information security, there are hazards such as exploits and Advanced Persistent Threats (APTs), which are discussed in this work. Both fields (trust and cyber security) have an enormous impact on modern society, mostly because almost everything in our day-to-day activities deals with some information and communication technology that are connected to the computer networks or are dependent on information systems.

The following sections summarises the main research problems, what were the motivations and the objectives of this study. After these sections there is a short resume of the main contribution of this work and how this document is structured.

1.1 Research Problem

In order to provide the identification of the problem, this section serves as parameters for the discussions found in this work, which will be further detailed in the appropriated sections of this manuscript.

During the research investigation of the literature, it was realized that trust and reputation systems lack a solution to address a group trust view in computational systems. Also, the relations of trust with information security and cyber security are referenced as distant subjects. This presented a research opportunity with emphasis on trust and the connections with information security. Considering this opportunity, this work started with the idea of getting deeper in the subject of trust and information security. After the organization of the topics and the subjects, it was decided to divide the research problem into two main subjects.

The first subject is that trust models deal with one-to-one (1:1) relationship and not one-to-many (1:N) or many-to-many (M:N) relationships. In other words, current trust models are used to calculate one-to-one relationships between entities. This means that in a computational system, if an entity, let us say **A**, needs to determine the trust value for all the other entities available in the system, which it already communicated with him, entity **A** will have to ask another entity for this information. In large-scale scenarios, this fact is an important lack of scalability because **A** needs to calculate and maintain as many

trust relations as the number of entities necessary to accomplish its work in a distributed system.

There are other scenarios that consider several entities as a single one for trust purposes. For example, a distributed system could be seen as a single entity even though it is composed of several sub-entities. When **A** needs to communicate to a group of entities having the guarantee that the group itself is trustworthy, **A** wants to independently form an opinion about the whole group. If this is the case, it may be necessary to establish communication with every group member. This makes node **A** start an exhaustive process of discovering every member of the group. In most practical situations this is not a good approach because it will make **A** try to find maybe thousands of nodes in a network. This is resource and time consuming for **A** and may not be a good solution to achieve its objectives. So in this case, the first subject of the problem is to find a way of representing group trust in the perspective of a single entity (1:N) and representing trust to a group of nodes to another group of nodes (M:N).

The second subject of the problem is that most of the trust models do not address its relations with information security. It is understandable that managing information security is critical, but current information security frameworks do not deal with the different facets of information technology properly and do not have a trust approach. Besides, information representation, elements that treats them, operations and support components can be integrated to show the various risk sources when dealing with information, security and trust. In this case, the second subject is to create a framework or architecture for representing information security and its relations to trust and to address aspects such cyber security and its connections to trust and cyber threats.

1.2 Motivation

In general, trust management in computational systems is a mechanism used to decrease the complexity of the access control and authorization of operations, allowing an approach to cope with the decentralized structure of distributed and computational systems. In such systems, there is a dynamic behavior where arrival and departure of nodes or users happens constantly.

In most distributed environments it is desirable that they are able to organize themselves autonomously in order to provide a communication and processing structure to the participant entities, even though they do not have a previously determined structure nor a centralized control. Clearly, such activities are not a trivial task and they are usually presented in mobile ad-hoc networks, computational grids, peer-to-peer systems, agent communities, cloud systems, and so on.

One particular motivation for this work is that most reviewed trust models do not apply the group criteria in the process of calculation of trust and reputation. The models reviewed and referenced consider the perspective of trust of one node to another node (what is called one-to-one relationship in this work). Thus, this researcher realizes that there is a need of trust and reputation representation in groups and its application in practical scenarios.

Considering this situation, it is also a strong motivation to check the possibility of creating a group trust and reputation model and its implementation. Particularly, reputation criteria (relationships, social information, trust level, etc.) regarding a specific entity about a group (1:N), or the opinion of a group about another group (M:N) is subject of interest in trust and reputation mechanisms.

It is important to mention that trust and reputation models do try to consider typical human behavior in computational systems. This behavior deals with the notion of society, community and organizations. Such aspects are also interesting as challenges in creating such models as extension of trust and reputation use in distributed systems.

Besides that, it is essential to point out the relations between information security and trust. Most information security architectures do not show how information security and trust are connected together. Having such concerns in mind, this is also a particular incentive to research topics in information security and cyber security. This research tries to present architecture that connects trust to information security and pointing its relations to cyber security.

1.3 Objectives

This research has one main objective considering the state of the art in trust, reputation and information security. This main objective is to develop a trust model applied do groups in computational systems and after that to establish a relationship between trust and information security. This objective requires the process of reviewing a broad literature on trust and reputation models, besides its use in distributed systems and its relations to trust and information security.

Considering the challenges of this main objective, it was divided into two main subjects. The first subject is to find a way of performing group trust and reputation calculation in order to represent the group trust view of a particular computational system. The second subject is to research and develop information security architecture and show its relation with trust and cyber security.

In order to fulfill the main objective and its subjects, the following specific activities shall be conducted during this research:

1. Create a group trust model applied to computational systems;
2. Implement and show the results of the created model;
3. Create a framework or architecture that represents information security and trust;
4. Provides the explanation of such architecture and its relation with information representation, information treatment and its relation to trust; and
5. Describe cyber security and its relations to trust regarding threats in information security field.

1.4 Summary of the Contributions of this Thesis

Basically, this thesis provides five contributions to trust, information security, distributed systems and cyber security fields. These contributions are divided into 11 papers published and 1 paper accepted for publication. Figure 1.1 provides a short schema of the publications and the related fields of knowledge where the publications can be fitted.

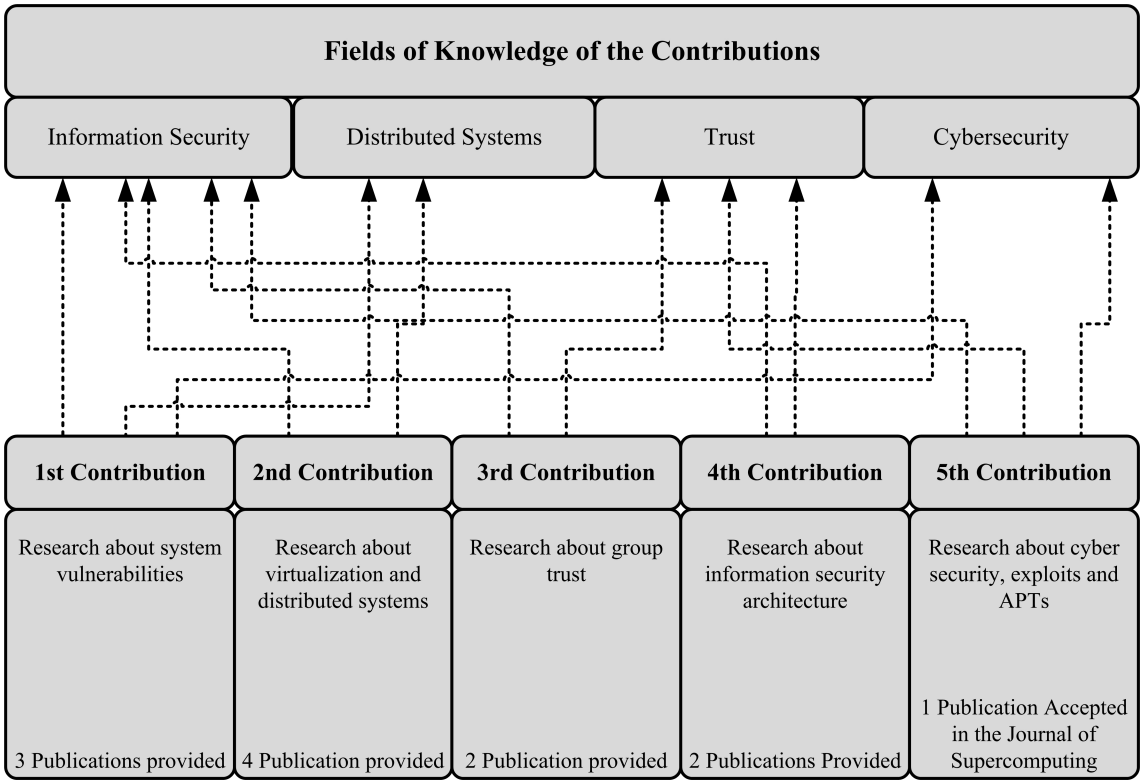


Figure 1.1: Schema of the contributions of this thesis

The first contribution came during the review of the literature. It was observed that parts of the information security field provided opportunities of investigation related to systems vulnerabilities. This opportunity was transformed in contribution throughout the publication of the articles "SisBrAV - Brazilian Vulnerability Alert System" [dOASALP+05], where the main idea of a system is provided based on the implementation of a vulnerability search and alert system using free software. Afterwards, the paper was published "Proposal of a System for Searching and Indexing Heterogeneous Vulnerabilities Databases" [dOAMBGV+06], where the main idea is an evolution of the first paper and spider mechanism for searching for online data. The last article related to this contribution was the publication of a solution to increase security processes in communications protocols. This publication was "Enhancing an Integer Challenge-Response Protocol" [dOAGVdSJ08].

The second contribution focused on the solution of virtualization techniques and research about distributed systems. The main idea of this contribution was to increase the efficiency of integrated computational systems. The first paper of this contribution was the publication of the article "Load Balancing and Survivability for Network Services Based

on Intelligent Agents" [dOAdSJdS+04], where the use of software agents is discussed to provide load balance solutions. The second contribution to distributed systems was the study of problems in Ad-hoc networks, with the publication of the article "An Overview of Open Issues and Preliminary Solutions Regarding Security in Ad-hoc Networks" [dSJdOAMBGV06], and the paper "MANET Auto Configuration with Distributed Certification Authority Models Considering Routing Protocols Use" [dOAdSJBAGV05]. To complete the contributions to virtualization studies, the work entitled "Virtualization with Automated Services Catalog for Providing Integrated Information Technology Infrastructure" [dOAVRTGdD11], was published where virtualization techniques are discussed and some results regarding automatization of processes are provided.

The third contribution is related to trust. This contribution published two works where a group trust and reputation model that can be applied to the computation of trust in a perspective of groups. The first paper published was called "Group Trust Model" [dOAGV12]. The second work was achieved through the use of a developed extension to conventional trust models that enables the support for calculating the trust values for groups of nodes, herein defined as Group Trust (GTRUST). In the developed mechanism, entities that need to interact with a given group are able to find a trust value for the whole group. This approach avoids the necessity of discovering the whole group members by an external entity. This contribution provided the publication of the work "Group Extension for Trust Models in Distributed Systems" [dOAGVK14].

As a result of the research in trust applied to address the group trust model, the fourth contribution of this thesis is the presentation of information security architecture supported by trust, defined as TISA. The purpose of the developed architecture is to extend and to discuss security information, its extensions and its relation to trust. This contribution provided the publication of the work "Arquitectura de seguridad multinivel: una guia para las organizaciones modernas" [dOABGV14] and the work "A Layered Trust Information Security Architecture" [dOAGVSO+14].

The fifth contribution is the presentation and discussion of cyber security considering strategies, exploits, APTs and its relations to trust. As results of this part of the contribution, this research sent a paper to the Springer Journal of Supercomputing, which was accepted for publication.

1.5 Outline of the Thesis

This thesis is organized as follows:

In Chapter 1 there is a brief explanation of this work constituted of a summary, the research problem and the motivation, followed by the objectives and the summary of contributions provided by this manuscript.

In Chapter 2 there is the review of the state of the art regarding trust. In this chapter there is the definition of trust, how it started to be used in computational systems, examples of its usage in distributed systems. There is also a review of current trust models used and its basic classifications accordingly to the literature.

In Chapter 3 the study about reputation and some of its relations to trust is presented.

Reputation is used as a means of acquiring trust. Some works are reviewed on the use of reputation in computational systems where trust is also used and referred. Also in this chapter some of the research done in the information and cyber security fields is summarised. Both subjects gained a lot of attention in research and development area recently due to the lack of confidence in computational systems and the Internet. This review shows some work and previous relations of trust and information security. The reviewed works about cyber security also aggregates value in the discussion regarding trust.

In chapters 2 and 3 there are the definitions of the main subjects used in the whole thesis. They serve as a basis of the discussions during the results in the following chapters.

Chapter 4 explains the group trust approach and describe the fundamentals of proposed group trust and reputation model developed. Besides it presents the proposed algorithm used to perform group trust calculations and the considerations applied to the simulation environment used to demonstrate the viability of the model. After this, this chapter also presents and discusses the results reached by the developed model and its analysis.

In order to provide a deeper view of trust and information security, Chapter 5 presents the proposed information security architecture and its relation to trust. In this chapter the developed architecture is detailed in terms of its pieces. Every piece is explained and a strong relation to trust and information security is established. Following this, this chapter extends the discussion considering information representation in digital environments and the relation of the proposed architecture can be understood.

Chapter 6 establishes a discussion about information assurance, information security and cyber security. In order to extend the subject, areas of interest applied to the creation of cyber security strategies are presented and reviewed. There is also a study about exploits and a market where values of vulnerabilities are sold by huge amounts of money. The study of exploits also presented an opportunity to detail discussions about APT. At the end of this chapter comes the establishment of the relation about information security, cyber security and trust.

Chapter 7 summarizes this research. It shows the conclusions that could be observed and points towards potential future work related to trust, reputation, group trust, information security and cyber security and its relations to trust.

1.6 Audience of this Thesis

The prerequisites to access this thesis material are not high. In order to make the work self-contained, we repeat several of the definitions and concepts available in the literature.

Trust is typically a human behavior. However from the perspective of this work, it can be measured in terms of probabilistic calculations using previous known models used to generate one-to-one trust. This work extends one trust model in order to use one-to-many and many-to-many trust dealing with the viewpoint of groups. The results give some information on how to deal with problems as contexts and threshold related to trust and reputation.

In some cases of this work, it is important that the reader has in mind that practical

and theoretical understandings of information security, cyber security and its connected elements may be needed. Information security deals with a lot of different matters and particular subjects. Some of the involved problems are herein discussed, while others are considered that the reader already has foundations to understand concepts discussed.

The reviewed bibliography provides the reader with further information where more details can be found regarding the study conducted by this thesis.

Chapter 2

State of the Art about Trust

This chapter reviews the main concepts of trust and its applications in computational systems. Bearing in mind a broad view, trust aspects towards human behavior will also be presented and discussed.

This chapter is organized in 6 sections. Section 2.1 reviews the concepts of trust. Section 2.2 gives details on the relation of context and information related to the third party in a communication process. Section 2.3 discusses some basic characteristics of trust. Examples of computational systems where researchers use trust to evaluate its behavior in such systems is the subject of section 2.4. Section 2.5 presents a review of trust models. Lastly, section 2.6 summarizes this chapter.

2.1 Definitions of Trust

Trust is the subject of study for many researchers. Some of the most well-known are Gambetta [Gam00], Lamsal [Lam01] and Dagsputa [Das00]. Most known definitions of trust are constructed by human behavior. They are based on people's relationships, which gives the sense of being protected from harm or being safe [Gam00], Lamsal [Lam01].

According to the online Merriam-Webster Learner's Dictionary[1], trust is:

a. To believe that someone or something is reliable, good, honest, effective, etc.;

b. To have confidence in (someone or something);

c. To believe that something is true or correct;

d. Somewhat formal: to hope or expect that something is true or will happen - often used to politely tell someone what you think they should do;

e. Trust in (someone or something) formal: to have a strong belief in the goodness or ability of (someone or something);

f. To have trust in (someone or something);

[1] Definition found at http://www.learnersdictionary.com/definition/trust

g. Trust to (something): to rely on (something you have no control over, such as luck or chance) to get what you want or need;

h. Trust (something) to (someone): to give the responsibility of doing (something) to (someone);

i. To allow (someone) to have or use (something valuable).

Considering the human perspective of trust towards comfort and safety, men make every-day decisions based on trust (including cross-relations). For instance, if you feel sick, the normal decision is to go to the doctor because you trust the doctor will help you with your sickness. In such decision processes, you also assume the responsibility of the action and the eventual effects of such decision.

If we ponder the scenario of the doctor, and consider you arrive to be questioned on issues such as your bank statements, your last trip destination, or even the color of your dog, the amount of trust in this particular doctor will decrease as such questions have nothing to do with your sickness. Or if the diagnostics of your sickness makes no sense, for example, you have a headache because your shoes are black. The amount of trust will decrease too.

A person decides to trust someone or something when the information to accomplish the desired action may be incomplete. In such cases, the person may search for other ways to fulfill their expectations, such as considering previous situations or the reputation of the other person in his social environment. To summarise, human trust depends on the context and the feeling of being safe to fulfill an expectation that there is a solution for the problem in question. This process is the result of much analysis, which together, gives the notion of trust ([Gam00], [Lam01]).

When the expectation that the problem has a solution with the help of another agent, the trust in this particular agent increases. The feeling of betrayal of trust arises when there is the feeling that the expectation was not fulfilled. In this case, the betrayal process analysis may consider the context in which it happened and even external variables (opinions, relationships, historical records, etc.).

The process to build trust in humans is a complex activity because it depends on a lot of things, such as emotions. However the process of losing trust is almost instantaneous. In this case, to rebuild trust, it may be very hard to acquire it again or it may not even be possible at all. Human trust is an abstract concept considering such perspectives and there is not just one single truth, but trust is present in daily human activities and behavior.

Considering trust in the computational perspective, Patel [Pat07], Suryanarayana *et al.* [ST04] and Marsh [Ste94] developed their research considering psychology in their models. When trust is used in the computational perspective, the most accepted definition is provided by Gambetta [Gam00], which is reinforced by Lamsal [Lam01]. This definition states that trust is a particular level of the subjective probability, with which an agent assesses that another agent will perform a particular action, both before it can monitor such actions and in a context in which it affects its own [Gam00].

In general, without trust there is no cooperation. Trust can be seen as a possible behavior that an agent will accomplish a determined action expected by another agent. In

this situation an agent may be able to verify the execution of the action (if the agent has this ability) and the desired action will affect its own behavior, which leads to a decision process. In such case, if the agent that executes the action is trustworthy, there is a high probability that it will execute the action accordingly, thus leading to a cooperation process. But if the probability is below an expected threshold, the cooperation shall be avoided. In this situation, the cooperation has a probability of being unsuccessful. For example, consider that there only exists distrust between agents, so they cannot cooperate with each other because they will not be able to fulfill each other's expectations.

Still, considering a computational perspective, the work of Marsh [Ste94] is among the first to represent trust in computation. Marsh says that trust can be represented throughout mathematics, so it can be implemented in a computer. Basically, the assumption of Marsh leads to the representation of trust as shown in Figure 2.1, where 0 represents distrust and 1 represents blind trust.

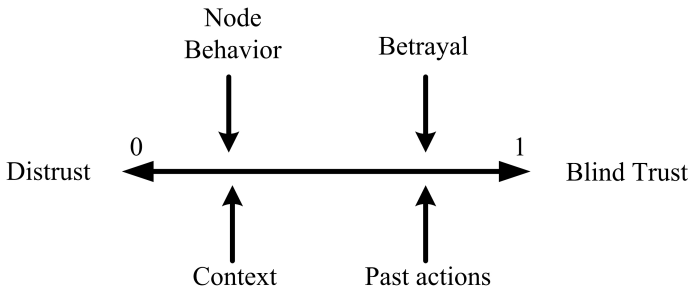


Figure 2.1: Trust representation (adapted from Marsh [Ste94])

In general, the blind trust assumption means that an entity believes so much in another entity that it forgets to assume that the entity in question may not be trustworthy anymore. According to this work, the blind trust perspective should be avoided, thus leading an entity to verify the trust situation of the other entity. Basically, blind trust eliminates any possibilities of suspicion of an entity in another entity.

When assuming that there is a trust mechanism involved, an entity may be able to generate its own trust, which is defined as direct trust (detailed in section 2.5.2). Direct trust is when the entity needs to use its own inferences in order to calculate a trust value. When the entity takes other aspects into consideration, such as relationships and recommendations, then it is assumed to be indirect trust (detailed in section 2.5.3).

Gambetta [Gam00] and Marsh [Ste94] consider that trust also suffers influence of competition and cooperation. Risk also influences trust, according to [Gam00], [Ste94] and [Pat07]. In this particular case, the risk can be seen as an expectation that the cooperation will not flow or it will not bring any benefits in the context in focus.

2.2 Context and Third Party Information

Context has direct influence in trust mechanisms. Context can be represented as a particular situation that has significant difference when compared to each other. Normally the context is directional and it is linked with a simple relationship. This can be seen in

Figure 2.2(a), where 1:1 means one-to-one relationship and Figure 2.2(b) represents the third party information.

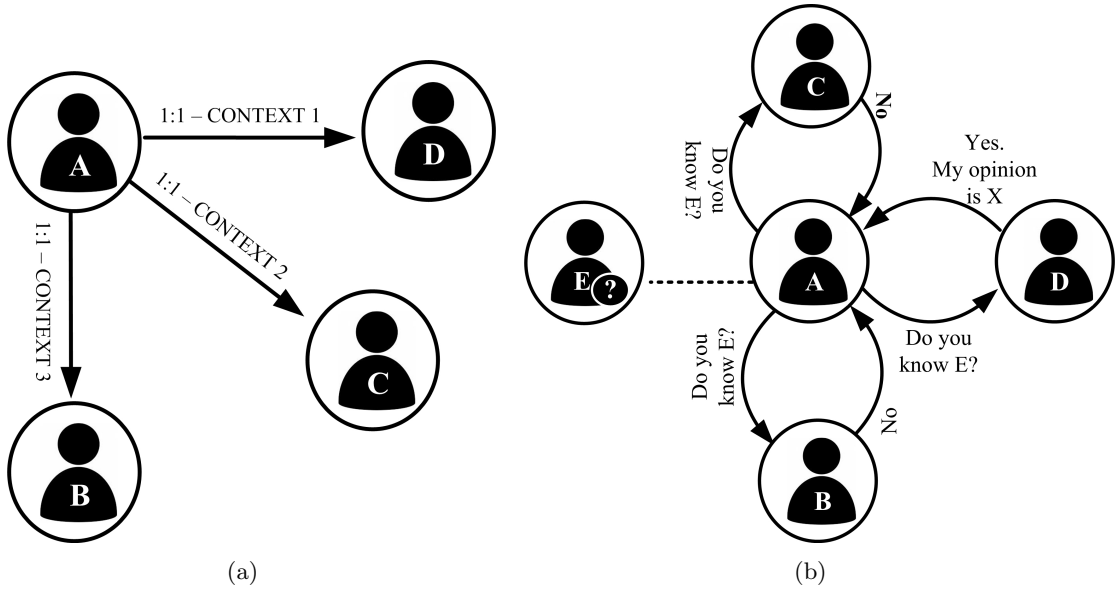


Figure 2.2: Context and Third-Party Information

Trust suffers influence of different mechanisms. For instance, consider that an entity (let us name it **A**) needs to communicate with another entity, it may be required a direct interaction. In this particular scenario, if entity **A** has no information about the other entity (**C**), it can be introduced to **A** by another entity, let us call it **B**. In this case **A** will receive some information about **C** through **B**.

In normal situations, an entity **A** may have as many relationships as it desires to accomplish a particular task or even solve new problems. According to Figure 2.2(a), **A** acquires or loses trust in **B** in context 1. The following has the same perspective to **C** and **D**. What is important in this evaluation is that the trust **A** has in **B** in context 1 belongs to **A** and to **A** only. In the case of **B**, **C** and **D**, as the contexts are different, the opinion of **A** may vary according to the development of the relationship.

Considering that a particular context may be a complex relationship, it might have different variable environments attached to it. For example, the variables could represent problem definition, time base history, the solution found or if the communication process had any failures, etc.) Considering this particular view, entity **A** is able to build a knowledge base representing the context itself, who was the entity in the context and the associated variables. Table 2.1 represents this particular knowledge base in the perspective of entity **A** and it could use this information to build his trust in **B**, **C** and **D**.

Table 2.1: Knowledge Base of A

Entity	Context	Variables
B	1	Va1, Var2, Var3, VarN
C	2	Va1, Var2, Var3, VarN
D	3	Va1, Var2, Var3, VarN

In order to build trust, there is also a perspective where **A** does not know a particular entity but it comes to a situation where **A** has to establish a relationship to this unknown entity. In Figure 2.2(b), **A** requests information about an unknown entity **E** to his parties **B**, **C** and **D**, which **A** has already knowledge of. In the particular example, only **D** knew **E** somehow and has information in his knowledge base. **D** answers **A**, so it can use this information to infer how entity **E** behaves in a particular situation. If the entity **A** so desires to establish a communication process with **E**, he may do so and add new information to its knowledge base of the new entity.

According to [SS02], [Pat07] and [Ste94] there is another situation where one entity might introduce a second entity to a third entity. Considering this perspective, an entity **A** presents entity **B** to entity **C**, where entity **A** has some trust information about **B** and **C**. This situation is an inversion of the ability to search information about other entities. It is useful, for example, when **A** knows **B** has a problem and **C** is candidate to help in the solution of the problem of entity **B**, for whom **C** is unknown.

2.3 Basic Trust Characteristics

The review of the common literature about trust ([Lam01], [SS02], [Pat07], [Ste94], [Gam00], [ST04]) shows that there are common aspects in almost all trust models found in the perspective of this work, which allows the creation of a common set of features related to trust. These features can are summarized in Table 2.2.

Table 2.2: Summary of basic trust characteristics

Characteristic	Example
Trust is context-aware	Entity **A** may trust **B** to download files but do not trust **B** to perform routing.
Trust can be measured	Entity **A** trusts more in entity **B** than **A** trust in entity **C**.
Trust changes with time	The amount **A** trusts in **B** may increase or decrease as interaction happens.
Trust is social-aware	Entity **A** may trust entity **C** because **C** was presented to **A** by entity **B**, and **A** already trusts **B**.
Trust may be directional	Entity **A** may trust **B**, but **B** may not trust **A**.

2.4 Computational Systems and Trust

Trust has been studied in many Computational Systems (CSs). From the perspective of this work, a computational system is any system that is able to process and represent information mathematically. When it comes to the study of trust and reputation, which is detailed in 3, a computational system has to be able to produce and calculate trust and/or reputation information, be it in a centralized or distributed approach.

The study of trust and reputation in CSs assumes that most of the behavior in open and distributed environment can be predictable. However there is no certainty the behavior of an entity when it comes to trust. Basically, the intentions of entities are unknown. Hence, due to the uncertainty of their potential behavior there is the requirement to control the interactions among parties through communication in a distributed environment. This characteristic also intends to separate good entities from fraudulent ones.

Pinyol and Sabater [PSM13] summarises that trust and reputation has three approaches which tries to address such problems. Table 2.3 resumes them.

Table 2.3: Trust and reputation approaches to computational systems

Approach	Summary of the characteristic
Security	Basic structural properties are guaranteed (authenticity and integrity of messages, privacy, identities, etc). There is the use of cryptography, digital signatures, electronic certificates, etc. The main problem is that there is no data on the quality of the information, although the established control is more than valuable.
Institutional	It assumes a central authority that observes, controls or enforces entities actions, and might punish them in case of non-desirable behaviors. This approach ensures a high control but it requires a centralized hub. Control is bounded to structural aspects of the interactions (allowed, forbidden or obliged), which can be checked and controlled. The quality of the interactions is left apart (a good or bad interaction has a subjective connotation).
Social	Reputation and trust mechanisms are normally focused in this approach. Entities themselves are capable of punishing non-desirable behaviors, for instance, by not selecting certain partners. To achieve such distributed control entities must model other entities behaviors, so trust and reputation mechanisms appears to be a good solution. This requires computational models of trust and reputation, which must cover not only the generation of social evaluations, but how entities use reputation information to select partners, how to communicate and distribute reputation information and how to handle it.

Considering such aspects, there are a lot of different CSs where trust studies have been applied to them. Examples of such systems includes cloud systems, grid systems, Peer-to-Peer (P2P) systems, software agents, internet and others. The following sections present and summarise some published works about the use of trust in some CSs.

2.4.1 Cloud Systems and Trust

In Canedo's paper [DCdSJdOAdM12], cloud computing is being progressively adopted in different business scenarios in order to obtain flexible and reliable computing environments. The representation of trust in computing systems has been widely discussed and applied in a lot of information technology scenarios. Their paper proposes the development of a trust model to ensure reliable files exchange among nodes in a private cloud. It also provides the calculation process of trust among them, according to the

established metrics, using a private cloud environment.

To Qu and Buyya [QB14], cloud computing is a utility computing paradigm that allows users to flexibly acquire virtualized computing resources using different model approaches. Their paper proposes a system that evaluates trust of clouds according to users' fuzzy Quality of Service (QoS) requirements and services' dynamic performances to facilitate service selection.

Ding *et al.* [DLG15] believes that trust evaluation research in cloud computing has been involved in the node security communication, security storage, resource allocation and many other aspects. As an effective replacement of traditional network security, trust mechanism has effectively solved some of the security problems of distributed computing, such as grid computing, pervasive computing and ad-hoc networks. Their paper proposes a trust model based on an evidence theory which considers factors such as time. The proposed model studies the qualification of the entity's recommendation trust. It also proposes a dynamic allocation method of trust weights and gives a calculation method to generate trust value. In brief, the work of Ding *et al.* [DLG15] discusses that there is a need to consider trust issues in the process of selecting a cloud service and considers trust problem within each entity in cloud computing.

The paper of Messina *et al.* [MPR+14] offers a trust model for a competitive federation of Cloud computing systems. A trust model helps the requester to properly choose the adequate provider on the basis of the reliability shown in the past in providing services, and the reputation the provider has with other users. The reference scenario used in [MPR+14] is a large-scale environment in which different cloud providers compete in offering their pay-per-use Anything as a Service (XaaS) resources to the various users and uses trust as a way of verifying what is claimed by cloud providers.

To Habib *et al.* [HRMV14] customers face problems identifying a trustworthy cloud provider based solely on Service Level Agreement (SLA), which is normally offered as parameter to users. Considering that it is important to support customers in identifying trustworthy cloud providers, their paper proposes a multi-faceted trust management system architecture for cloud computing marketplaces. Authors claim that the proposed system provides the means for identifying trustworthy cloud providers in terms of different attributes (compliance, data governance and information security). Habib *et al.* [HRMV14] also considers that trust and reputation systems are successfully utilized in numerous electronic marketplaces to support users in identifying dependable and trustworthy providers (i.e, eBay, Amazon, and App markets for mobile applications).

The process that deals with the establishment of trust is one of the most challenging issues in emerging cloud computing area according to Sidhu and Singh [SS14]. As contribution, their paper proposes a model to design and simulate a mechanism to calculate trustworthiness of service providers based on their compliance to the promised SLA parameters. Their results show that the model works and can be used to evaluate trustworthiness of service providers in a cloud environment.

Shaikh and Sasikumar [SS15] affirm that cloud computing has become a part of the competitive market and to analyze and measure a particular service based on its security properties is a challenge. Considering this view, their work presents a trust model that

measures the security strength and computes a trust value. This trust value contains various parameters that are necessary dimensions along which security of cloud services can be measured. Their model acts as a benchmark and ranking service to measure security in a cloud computing environment.

The role of trust in cloud computing services is explained in the work of Adjei *et al.* [ABB15]. Their paper answers questions regarding the role of trust in cloud computing service acquisition and what policies promote trusted cloud computing services. Some of their findings show that trust increases if users perceive that cloud computing service provider's act honestly and with the users' interest in mind, making trust a fundamental factor that helps the decision to acquire cloud computing services.

Sun *et al.* [SCSW11] provided a survey that discussed major security, privacy and trust issues in cloud computing environments. According to their paper, cloud computing has security as one of the major obstacles for computing as a utility. Their work pointed to further development of a complete security, privacy, trust evaluation and management framework on cloud environments.

In the paper published by Jam *et al.* [JKJA14] there is a discussion about trusted computing and security services. Their work describes the hadoop project and its present security mechanisms. There is an analysis of the security problems and risks, and a consideration of some methods to enhance its trust and security. According to the authors [JKJA14], hadoop platform is widely used in the business world, while the weakness of security mechanism has become one of the main problems obstructing its development.

According to Khan [Kha13], clouds typically place centralized and universal trust in all the cloud's nodes, thus data and computation integrity and security are major concerns for users of cloud computing facilities. As described in his work, the full-trust model has the negative consequence of amplifying potential damage from node compromises, leaving such clouds vulnerable to myriad attacks. In order to address this limitation, [Kha13] proposes and evaluates 5 paradigms (Hatman, Anonymous Cloud, Penny, CloudCover and Silverline) for decentralizing cloud trust relationships for stronger cloud security. Later, Khan [Kha13] concludes his work pointing future directions to every studied platform and its paradigms.

A survey related to single and multi-cloud security and addresses possible solutions for Hadoop data is provided by Devare *et al.* [DJGKS+15]. In their paper, the security of cloud computing is a major factor in the cloud computing environment, as users often store sensitive and secret information with cloud storage providers but these providers may be untrusted. Authors argue that there is a tendency when dealing with single cloud providers. It is predicted to become less popular with customers due to risks of service availability, failure and the possibility of malicious insiders in the single cloud. As a solution the use of multi-clouds reduces security risks and data inconsistency that affect the cloud computing user.

2.4.2 Trust Applied to Grid Systems

Grid is another computational system in which trust has been studied. Mainly the goal of computing grid is to offer users with access to resources they need and when they are required. A Grid solution gathers all resources together and makes them available to users and applications. It achieves such characteristics using middleware to dispense technological resources across the network [IBM04].

A Grid uses a set of open standards and protocols to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet. To Srivastava and Raperia [SR13], a grid computing system consists of machines that have different computational capabilities and it needs to be robust to deal with improbability once it operates in an environment where system performance is important.

Pernas and Dantas [PD05] argues that grid computing largely shares resources and services. It is considered an effective solution for many organizations to execute distributed applications to obtain high level of performance and availability. One problem according to the authors is that it is a complex task for an ordinary user, demanding a previous knowledge of the access requirements from a virtual organization.

Patel [Pat07] considers there to be problems in distributed systems when it comes to the use of trust. His work points that the use trust is able to solve problems related to intra-domain relationships in virtual organizations, and it can be used as an extension to increase the security of an environment besides the use of cryptography and/or certification authorities.

Basney *et al.* [BNO+05] proposes in their paper an extension to grid security infrastructure by adding facilities for dynamic establishment of trust between parties to provide better support for the dynamic and cross-organizational aspects of grid activities. Their work introduces the PeerTrust language for access control policies and shows how to use PeerTrust to model common needs in their environment.

A trust content model and content-included trust statement model are designed in the paper of Wang *et al.* [WWLY06]. Their work considers that grid computing is a dynamic system and trust changes over time. Authors argue that a trust model is defined with the objective of obtaining new trust value by the content of trust semantic and grid service semantic information, service content model, context model and situation model. Also, as results, the influence relationships between them are shown in their work.

Papalilo and Freisleben [PF04] consider that an important problem in grid computing is the management of trust among the entities involved in a computational process. Assuming this problem, their work presents a trust model supported by functionalities of bayesian networks to calculate trust based on the trust of the entity itself, its trust towards other entities and the trust of others towards the entity. It is also presented an implementation using the GridSim toolkit. According to the authors, the results illustrate how grid entities build up their trust values in selected interaction scenarios.

Tran *et al.* [TWHV05] proposed a trust model for grid environments based on their recommendation method, a fair-trading scheme, and an access control model. Basically

their model classifies grid nodes as good, bad or malicious. The proposed approach, according to the authors, preserves the grid's decentralized structure and participant's autonomy, but also enables secure service exchange.

According to the research of Nogoorani and Jalili [NJ15], the infrastructure offered by a grid permits researchers to solve various research problems through sharing their resources and establishing virtual organizations. The authors propose a framework for grid access control, which according to them, is the first trust-driven risk-aware access control framework. Their model uses obligations to seamlessly monitor users and mitigate risks, so trust evaluation and risk management are added to the base grid access control services.

Fadul *et al.* [FHAS14] debates the trust-management toolkit, which is a robust and configurable protection system extension, which can successfully function in the presence of an untrusted smart grid. The work presents a trust-management toolkit that is able to combine reputation-based trust with network-flow algorithms to identify and mitigate faulty smart-grid protection nodes. The toolkit assigns trust values to all protection nodes. When there is a faulty node, it is attributed to component or communication system malfunctions a lower trust value, which indicates a higher risk of failure to mitigate detected faults.

The paper of Shipman *et al.* [SHL15] applies a con-resistant trust mechanism to improve the performance of a communications-based special protection system to enhance its effectiveness and resilience. The con-resistant trust mechanism allows protection system nodes to make trust assessments based on the node's cooperative and defective behaviors. According to the authors, con-resistant trust is used to quickly identify malicious or untrustworthy protection system nodes to mitigate instabilities in smart grids. As a result, their model compared a simulated special protection system with a con-resistant trust mechanism to one without the proposed mechanism via an analysis of the variance statistical model.

2.4.3 Peer-to-Peer and Trust

Generally, P2P can be defined as a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session. This characteristic allows a P2P network to function as both client and server in a network. P2P has a distributed architecture that partitions tasks between peers equally, thus peers are both suppliers and consumers of resources.

Bandara and Jayasumana [BJ13] makes a further definition considering collaborative P2P systems. According to them, collaborative P2P systems goes beyond just being able to do similar things while sharing resources. Peers can bring in unique resources and capabilities to a virtual community, thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers and the overall results are beneficial to all peers in the community.

When the focus of P2P is towards trust, there are plenty of research work in the area. Tian *et al.* [TZWC08], proposed a recommendation evidence based trust model for P2P networks in their paper. The proposed model is able to avoid invalid recommendation

for building trust and dealing with uncertainty of information in the reputation-based trust model. They also proposed a feedback-based probabilistic searching algorithm with the objective of finding recommendation information. This algorithm, according to the authors, improves the searching success rate and lowers the network traffic. The results of their work show that their model has advantages in modeling dynamic trust relationship and aggregating recommendation information.

A mixed solution of grid and P2P is discussed in the paper of Junmao *et al.* [JSJM05]. Grid nodes provide service with QoS guarantee, while sharing computing resources of P2P nodes is the user's volunteer action without QoS guarantee. Considering such aspects, the authors claim that is difficult to establish the trust relationship among users with traditional trust mechanism. In order to provide a solution to the research problem, [JSJM05] designed a grid P2P trust model based on recommendation evidence reasoning to solve the problem by building a recommendation mechanism in Grid P2P and integrating the recommendation evidence with D-S theory.

Dou *et al.* [DWJZ04], argues that sharing is a voluntary action, thus peers are not responsible for their irresponsible bartering history. This leads to the problem that trust between participants cannot be set up simply based on the traditional trust mechanisms. To the authors, a reasonable trust construction approach comes from social network analysis, in which trust relations between individuals are set up upon recommendations of other individuals. Their paper presents a recommendation-based global trust model and gives a distributed implementation method. Based on analyses and simulations the model presented evolutions on trust security problems.

P2P trust models suffer greatly from dishonest feedback and trust metrics depending on subjective judgment, which excessively results in rough trust value, as reasoned by Wang *et al.* [WLM+02]. Based on this consideration, their paper proposed a versatile trust model based on multi-dimensional trust evaluation. The simulation and analysis showed that their model can calculate the trust value effectively and reasonably and discard the malicious peers from P2P system.

Udhaya and Basha [USB14] presented an exhaustive survey of existing trust and reputation models in P2P networks. Their review discusses trust problems like trust bootstrapping, trust evidence procurement, trust assessment, trust interaction outcome evaluation and other trust based classification of peers behavior into trusted, inconsistent, un trusted, malicious, betraying and redemptive.

Saini *et al.* [SSY14] considers the Internet, as it is nowadays, a medium of sharing an immeasurable amount of information for the extensive use P2P environments. Their paper argues that a fairly open structure of P2P network applications make peers exposed, thus the interaction with unfamiliar peer in the absence of a trusted third party makes them vulnerable to potential attacks. Their work evaluates existing collusion attacks and proposes a reactive defense mechanism against them by proposing a model that detects collusion based on underlying trust and reputation knowledge.

A peer based trust model defined as SuperTrust is presented to encourage peers to cooperate in a hybrid P2P system as shown in the work of Tian *et al.* [TYZL14]. According to their paper, without a sufficient number of transactions among peers, it is difficult to

build trust relationships. Their model divides peers into groups based on their sharing interests and a proposed algorithm filters similarities of peers' feedbacks such as fake, misleading and unfair referrals can be eliminated from the trust calculation. According to the authors, analysis and as the results demonstrate - the search failure rate of proposed model is low and requires little system resources and the model is robust and resistant to several popular P2P attacks.

Nithy and Balasubramanian [NB14] propose various trust metrics for the evaluation of trustworthiness among the peers in a P2P system. By evaluating the trust metrics based on the interaction between the peers, a trust relationship is constructed and it is measured based on the service provided by a particular peer and based on the recommendation collected from a peer about another peer. Authors argue that as a result of trust metrics evaluation, malicious peers could be separated from the network and good peers were able to communicate safely.

Building trusted relationships among peers in a large-scale distributed P2P system is a fundamental and challenging research area. A prevention and trust evaluation scheme called IRTrust is presented by Li *et al.* [LKYL14]. Their framework includes a strategy of identity authentication and a global trust of peers to improve the ability of resisting to malicious behaviors. According to the authors, IRTrust can defend against several kinds of malicious attacks, such as simple malicious attacks, collusive attacks, strategic attacks and sybil attacks [Dou02].

2.4.4 Trust in Software Agents

Trust and reputation (chapter 3) in some situations are used as complements while in others, they are used as single concepts. Software agents are an example of another computational system that uses trust and reputation mechanism in order to build more trustworthy solutions to distributed environments. Morreale [Mor98] believes that software agents have 5 characteristics (autonomy, learning, cooperation, reactivity and mobility), which makes them distinct to conventional software. To Nwana [Nwa96] software agents can be mobile or static, reactive or deliberative, and may have primary attributes such as autonomy, cooperation and learning capabilities.

Software agents execute tasks on behalf of a user or other computational system. Considering trust aspects in the communication process, it is desirable as software agents may be able to analyze and learn from interactions so it might help define metrics to build trust and reputation values. In such cases, trust applied to software agents may be able to help address problems of virtual communities' organization. These kind of organized environments help to simplify the search of information, what helps the interaction process of software agents. From this perspective, software agents may follow pre-defined orders or decide to follow new directions based on the interactions and information gathered using trust and reputation models.

To Pinyol and Sabater [PSM13] the cognitive dimension is important when classifying software agents and the ability to use trust and reputation mechanism. The cognitive elements (beliefs, goals, desires, intentions, etc.) are responsible for the differentiation of models that are able to have trust or to hold social evaluations. Software agents in

this perspective have the ability to explain their decisions and to reason about the trust structure itself.

Considering the paradigm of the intelligent or autonomous agents and Multi-Agent Systems (MAS) together with the spectacular emergence of the information society technologies, both are responsible for the increasing interest on trust and reputation mechanisms applied to electronic societies. Sabater and Sierra [SS05] provided a review to offer a panoramic view on computational trust and reputation models.

Still, to [SS05], the study of trust and reputation aggregates value to information and communication technologies, thus trust and reputation systems are recognized as key factors in electronic commerce because they are used by intelligent software agents as mechanism to search for trustworthy partners and to incentive decision-making about whether or not to honor contracts. In order to avoid cheaters and frauds, reputation is used in electronic markets as a trust-enforcing, deterrent, and as an incentive mechanism.

To Teacy *et al.* [TLRJ12] agents may be self-interested and, when trusted to perform an action, may betray that trust by not performing the action as required to achieve their goals. Since agents must be capable of assessing and identifying reliable interaction partners, even if it has no personal experience with them. Following this point, authors [TLRJ12] presented a hierarchical and bayesian inferred trust model, defined as Hierarchical And Bayesian Inferred Trust (HABIT), for assessing how much an agent should trust its peers based on direct and third party information. Their results demonstrate that their model is able to predict agent behavior in a simulated environment and based on data from a real-world webserver domain.

There are several models for representing both reliability and reputation in the view of Rosaci *et al.* [RSG12]. In their paper, authors extended a previous reputation model and introduce a new trust reputation model that considers interdependence among trust measures computed in the studied agent systems.

To Rosaci *et al.* [RSG12], it is important to understand that a trust relationship can involve multiple dimensions, depending on the particular perspective under which the interaction among agents takes place. Table 2.4 summarises some usual dimensions of trust.

Table 2.4: Dimensions of trust [RSG12]

Dimension of Trust	Example
Competence	A competent agent is capable of correctly and efficiently performing the requested tasks.
Honesty	An honest agent shows a trustful behavior, and it is not fraudulent or misleading.
Security	A secure agent confidentially manages private data and does not allow unauthorized access to them.
Reliability	A reliable agent provides reliable services. In other words, reliability measures the degree of reliance that can be placed on the services provided by the agent, including the efficiency.

Martínez-Sarriegui *et al.* [MSGGGS+12] proposed a trust and reputation model that estimates the reputation of an entity taking into account information from three trust

dimensions (the hierarchy of the system; the source of information; and the quality of the results). Their work is focused on hierarchical medical organizations, but according to the authors, their model is highly flexible and allows adaptations of the model to the peculiarities of agents systems.

Basheer *et al.* [BATG15] believes that modeling confidence of agents is important in heterogeneous agent communities. Therefore, confidence in multi-agent systems gives agents a form of control in making decisions and helps to improve the decision making process in such systems. Their paper presents an approach to agent-based confidence modeling, where two confidence requirements are integrated (trust and certainty), and to further strengthen their model [BATG15], it includes evidence as an additional requirement by which trust and certainty of an agent can be verified.

The proposition of Rodrigues *et al.* [RLO15] is to show a method to explore the flexibility of the decision support for the services' reconfiguration based on several pillars, such as trust, reputation and QoS models, which allows the selection of agents based on measuring the expected performance. The proposed solution includes the agents' intelligent decision-making capability to dynamically and autonomously change services selection on the fly, towards more trustworthy services with better quality when unexpected events happen.

2.5 Trust Models

Basically, trust models are applied to create a notion of trust or reputation (or both) in a computational system. This representation has the intention to picture an entity as good or evil, or as cooperative or malicious. In the following sections there is a discussion supported by the literature regarding trust and some of them also discuss details regarding reputation.

2.5.1 Basic Considerations about Trust Models

Dr. Paul English [Pau14], believes that there are four basic trust models, as summarised in Table 2.5:

Table 2.5: Basic trust models (adapted from [Pau14])

Basic trust model	Characteristic
Suspicious still	Don't ever trust anyone or anything, even after they have done something nice.
Suspicious until	Don't trust anyone or anything until they proved themselves.
Trust until	Trust anyone or anything until they make a mistake by your judgment.
Trust still	Trust anyone or anything even after they make mistakes, sometimes even when they hurt you.

Following this view, to Dr. English [Pau14], trust models can evolve according to basic assumptions, and this is represented in Figure 2.3.

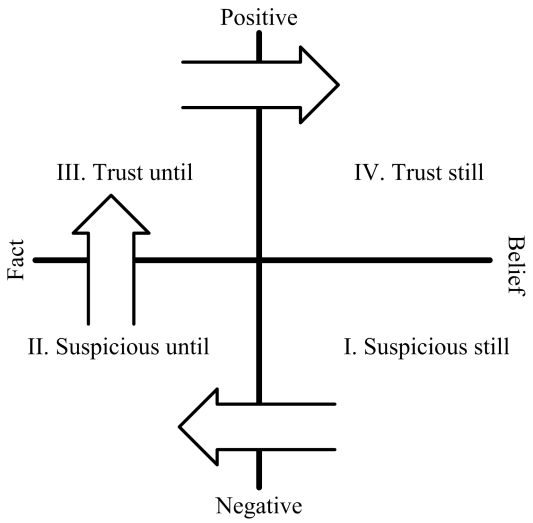


Figure 2.3: Representation of the four basic trust models (adapted from [Pau14])

Trust decisions considers expected benefits in a relationship. According to Marsh [Ste94], the expected benefits result in the use of others' ability through delegation. This leads to increased cooperation in an open and less protected environment.

A decision of an entity supported by a trust model is a multiple step process, which needs identifying and selecting the proper input data (trust indication, than performing some sort of computation (trust values), than making decisions based on the trust indicators and the notion of risk.

During the review of trust models, this work divided them in 4 different sections, which are commonly found in the literature as direct trust, indirect trust, cognitive trust and situational trust. It is important to mention that some models have in the same schema direct, indirect and reputation calculation. The following subsections describe some reviewed models.

2.5.2 Direct Trust

Considering the reviewed works, direct trust is when an entity is able to calculate its own trust value considering a direct interaction with another entity in a communication process in a specific environment. In the following subsections there are some direct trust models reviewed from the literature.

2.5.2.1 TRAVOS

Patel [Pat07] proposed TRAVOS as a distributed trust model. This model considers requirements such as scalability, trust and reputation evaluation, adjustment of non-trusted opinions, maintenance of interactions history, etc. Patel's model consists of a trust and reputation model for virtual organizations based on agents, in which trust is measured using probability. The evaluation of the amount of trust is based on past interactions and reputation obtained by other nodes.

In Patel's model, there are three methods to calculate trust. The first method is

based on direct interactions, using information gathered just by personal experiences. The second method is based on opinions of other entities located in the system. The third is a combination of both of the first two methods.

In the first method, the direct trust value is denoted as $\tau^{d}_{a1,a2}$ and is represented as equation 2.1.

$$\tau^{d}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}} \tag{2.1}$$

where $\hat{\alpha} = m^{1:t}_{a_1,a_2} + 1$ and $\hat{\beta} = n^{1:t}_{a_1,a_2} + 1$. $m^{1:t}_{a_1,a_2}$ indicates the number of successful interactions of a_1 with a_2 from time 1 to time t, while $n^{1:t}_{a_1,a_2}$ is the number of unsuccessful interactions of a_1 with a_2 from time 1 to time t.

The second method deals with information of reputation. The reputation value is calculated by $\tau^{r}_{a_1,a_2}$, represented in equation 2.2.

$$\tau^{r}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}} \tag{2.2}$$

where $\hat{\alpha} = M_{a_1,a_2} + 1$; $\hat{\beta} = N_{a_1,a_2} + 1$; $M_{a_1,a_2} = \sum_{k=0}^{p} \hat{m}_{a_k,a_2}$; $N_{a_1,a_2} = \sum_{k=0}^{p} \hat{n}_{a_k,a_2}$ and p=number of reports.

Patel's model relies on opinions from other participants of the community, so it needs to foresee inaccurate opinions, which may be consequence of not just malicious acts but also lack of information. Therefore, it is important to adjust or ignore non-trusted opinions before it combines with the total amount of reputation. For that adjustment, there is a metric of probability that measures the accuracy of the opinion based on the history of opinions provided in the past. This is represented by equation 2.3.

$$\rho_{a_1,a_3} = \frac{\int_{bin^{\Re^{t}_{a_3,a_2}}_{max}}^{bin^{\Re^{t}_{a_3,a_2}}_{min}} (B_{a_1,a_2})^{\alpha-1}(1-B_{a_1,a_2})^{\beta-1} dB_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}dU} \tag{2.3}$$

where ρ_{a_1,a_3} represents the accuracy of opinion provided by a_3 according to the truster a_1 at time t, B_{a_1,a_2} is the probability of a_2 fulfills its obligations.

The third method mixes direct trust and reputation, gathering information from itself and the other in the environment. The combined trust is denoted by $\tau^{c}_{a_1,a_2}$, represented in equation 2.4.

$$\tau^{c}_{a_1,a_2} = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}} \tag{2.4}$$

where $\hat{\alpha} = M_{a_1,a_2} + \bar{m}^{t}_{a_1,a_2} + 1$ and $\hat{\beta} = N_{a_1,a_2} + \bar{n}^{t}_{a_1,a_2} + 1$.

Patel's model also implements a metric defined as confidence, which measures the probability that the actual value of B_{a_1,a_2} may be a lie in an accepted level of error ε about $\tau^{r}_{a_1,a_2}$. This metric is calculated using the equation 2.5.

$$\gamma_{a_1,a_3} = \frac{\int_{\tau_{a_1,a_2}+\varepsilon}^{\tau_{a_1,a_2}-\varepsilon} (B_{a_1,a_2})^{\alpha-1}(1-B_{a_1,a_2})^{\beta-1} dB_{a_1,a_2}}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}dU} \tag{2.5}$$

2.5.2.2 BETH

Beth *et al.* [BBK94] proposed a model for direct trust considering a system that communicates trough data links. To the authors, direct trust can be expressed by equation 2.6.

$$A_{trust}^{seqX} \rightarrow B_x V_{p_{(v)}} \tag{2.6}$$

Their model considers that there is a direct trust if **A** knows that every experience with **B** are positive to trust class **X**. **Seq** is the sequence of entities that already measured the experiences except **A** and **B** and **V** is a relationship trust value which measures de probability that **B** will behave accordingly while he is considered trustworthy in the system. This value is based in positive experiences of **B** towards **A**.

So, according to Beth *et al.*' model, if **p** is the number of positive experiences of **B** towards **A** in the class **x**, then the trust value of such experiences is calculated by equation 2.7.

$$V_{z_{(p)}} = 1 - \alpha^p \tag{2.7}$$

The value $V_{z_{(p)}}$ reflects the probability that **B** is more trustworthy than a threshold α, based on information that **A** has about **B**. This trust relation measures the expectations that **B** is trustworthy to **A** related to one and only one task. This model depends of a correct definition of α, so according to the authors, the expectations can be fulfilled without security problems.

2.5.2.3 YAN

Yan [Yan14] proposed a trust model that is able to calculate direct trust based on interest of a group, where every group has a leader, which is responsible for the maintenance of information and global reputations of all nodes of his group. Following this assumption, the model proposes: i) a direct trust value that is calculated based on direct transactions between the nodes, ii) a reputation inner group value which is the global trust of the inner group and reflects the node ability and trust of supplying services to other nodes in one group; and iii) recommendation trust inter-group, which is the recommendation trust to nodes in different groups is the direct trust value of neighbor nodes. In general, to generate direct trust, his model considers equation 2.8.

$$D_{Tr(a \rightarrow b)}\Delta_t = \frac{tran_s}{tran_s + tran_f} * f(\Delta_t) \tag{2.8}$$

where direct trust D_{Tr} is the subject evaluated trust value based on the historical direct interactions between nodes. Assuming node N_a applied $tran_T$ count service requests to node N_b in time period t, using $tran_s$ to denote successful transaction times, $tran_f$ to denote fail transaction times. So the direct trust of N_a to N_b in this period is as 2.8.

To Yan [Yan14] the whole valid time span, used to calculate the direct trust of N_a to N_b is as equation 2.9.

$$DTr_{(a \rightarrow b)T} = \frac{\sum_{i=1}^{n} DTr_{\Delta_{ti}}}{n} * \omega^{\frac{\sum Tran_f}{\sum (Tran_f + Tran_s)}} \tag{2.9}$$

Reputation in group is denoted by G_{Rp} according to Yan [Yan14]. It is the global trust inner group, which reflects the node's ability and trust for supplying services to other nodes in one group. Within the group, every direct trust value between nodes after each of their transactions should be sent to the leader node and stored as a $m\ x\ m$ matrix, which stores the direct values. To generate reputation value is used equation 2.10. Recommendation trust inter-group is denoted by equation 2.11.

$$G_{Rp^i} = \frac{\sum_{n=1}^{m-1} DTr_{ni}}{m-1}, n \neq 1 \tag{2.10}$$

$$RTr_{uv} = \frac{\sum_{n=1}^{p} S_{ui} * DTr_{iv}}{\sum_{i=1}^{p} S_{ui}} \tag{2.11}$$

Yan [Yan14] considers that recommendation trusts from nodes with similar interests are more reliable which should be given higher weight. So his model chooses recommendations from the neighbor nodes with the similar trust when evaluating nodes trust in other groups. It is used to measure trust similarity between two nodes and it is expressed as the cosine similarity function as equation 2.12.

$$S_{ij} = \frac{\sum_{k=1}^{m-2} DTr_{ik} * DTr_{jk}}{\sqrt{\sum_{k=1}^{m-2}(DTr_{ik})^2}\sqrt{\sum_{k=1}^{m-2}(DTr_{jk})^2}}, k \neq i, j \tag{2.12}$$

Yan's model consider the decrease of trust as the time pass by. It is considering a valid period of time T, split in M_T time periods, so each period is calculated as equation 2.13 and the function of the decrease is expressed as 2.14.

$$T_p = \frac{T}{M_T} \tag{2.13}$$

$$f(t) = \rho^{MT\frac{T_c - t}{T_p}} \tag{2.14}$$

Finally, the trust of one node, according to Yan [Yan14] is given by the equation 2.15.

$$Tr_{a\to b} = \begin{cases} \alpha DTr_{(a\to b)} + (1-\alpha)G^b_{Rp} & \text{(same group)} \\ \beta DTr_{(a\to b)} + \gamma Rtr_{(a\to b)} + (1-\beta-\gamma)G^b_{Rp} & \text{(different group)} \end{cases} \tag{2.15}$$

2.5.2.4 KIEFHABER

A model of direct trust and reputation in organic computing systems is proposed by Kiefhaber [Kie14]. Basically, the author considers that the reliability of a node is an important basis for all interactions executed on the system. In a communication mechanism, if a node is not reliable, it cannot be reached due to a high rate of lost messages, the applications on it can be as good as they will, their calculations are unable to reach their interaction partner or only with a lot of delay due to error corrections.

Considering such aspects, [Kie14] developed an algorithm to evaluate trust based on acknowledge of information to detect node failure. His model considers that the calculated

trust value t_n for node n can be expressed as equation 2.16.

$$t_n = t(self).t(real)_n \tag{2.16}$$

where $t(self)$ represents the reliability value of the node, which is rating the node n, and $t(real)_n$ is the trust value of the node n, if the observing node would be completely reliable and not lose some messages itself.

[Kie14] considers that when a node is highly unreliable itself, it will eventually gain an appropriate estimation of the reliability of its interaction entities regardless of its own reliability. In this case a metric that rates older values higher than newer values is desirable. This is because older values actually depict the real reliability value of the interaction entity whereas newer values most likely are timed out negative experiences due to lost messages with acknowledgments. Based on these experiences a trust value for reliability t_r can be calculated. [Kie14] examined several metrics for this calculation, with n the amount of messages ordered by the time of their occurrence and x_i the experience of the interaction, where 1 is given for a positive experience while 0 for a negative one.

The following equations (2.17, 2.18 and 2.19) gives the author's perspective of direct trust calculation. Equation 2.17 is a simple arithmetic mean metric, while 2.18 and 2.19 are a weighted arithmetic mean metric that weight the experiences based on the time they occurred. Equation 2.18 weights newer experiences higher whereas 2.19 weights older experiences higher.

$$t_r = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.17}$$

$$t_r = \frac{\sum_{i=1}^{n} \frac{i}{n} x_i}{\sum_{i=1}^{n} \frac{i}{n}} = \frac{\frac{1}{n}\sum_{i=1}^{n} i\, x_i}{\frac{n+1}{2}} \tag{2.18}$$

$$t_r = \frac{\sum_{i=1}^{n} \frac{n-i}{n} x_i}{\sum_{i=1}^{n} \frac{n-i}{n}} = \frac{\sum_{i=1}^{n} (n-i) x_i}{\sum_{i=1}^{n} (n-i)} \tag{2.19}$$

Kiefhaber [Kie14] argues that equation 2.19 does not look like a reasonable metric considering the higher expressiveness of more current experiences at first place, but in case of an interaction entity with consistent, but fluctuating, behavior, weighting older experiences higher results in a trust value with less fluctuation. And this behavior is a result of an interaction entity with a specific reliability value but with a high variance around it (e.g., 20% of messages are lost by an interaction partner based on an observation of lost (0.0) and received (1.0) messages).

A mean value will average these losses to a reliability value of 0.8 (loss of 20% messages), but new experiences of 1.0 or 0.0 will create a high fluctuation around the mean value especially if current experiences are weighted high. Kiefhaber [Kie14] reasons that this effect is reduced by rating old experiences higher.

2.5.3 Indirect Trust

Normally, the concept of indirect trust is when one needs to define a value of an undefined trust relationship based on previous ones that are used to propagate trust

information. Indeed, an indirect trust models gets a trust path as input and generates an indirect trust value. Both models of Patel [Pat07] and Yan [Yan14] represent examples of how the calculation of indirect trust might be achieved, besides direct trust. Below there are more two examples of indirect trust models.

2.5.3.1 GAYEN & CHANDRA

To Gayen and Chandra [GC14] two entities, say i and j, can also share mutual trust (indirect trust) due to trust transitivity effects. The authors consider an indirect link trust of node$i$ on j as the sum of the direct trust of j's neighbors on j, weighted by the indirect trust of node i on the corresponding neighbor of j.

So, to [GC14] the estimation of indirect link trust of node i on j is expressed as I_{ij}. Considering there is k as a neighbor entity of j, then the indirect trust of node i on node j is expressed as equation 2.20.

$$I_{ij} = L_{ij} + \beta \sum_{k \in N(j)} I_{ik} * L_{kj} \tag{2.20}$$

where β is a damping factor that lowers the trust weight of entity i on entity k with increasing number of hops from i to k. Since this value can be greater than, the authors obtain the normalized indirect link trust as represented by equation 2.21.

$$\hat{L}_{ij} = \frac{I_{ij}}{\sum_{s \in N-i} I_{is}} \tag{2.21}$$

Gayen and Chandra [GC14] also argue that the effect of trust transitivity is that it leads to the formation of clusters. So, the authors consider that the clustering coefficient of order x for an entity i having k_i number of neighbors can be defined as the ratio of the number of neighbor pairs who has a path of distance x among them (through any entities other than i) and the number of all possible pair of neighbors.

If $E_i(x)$ denotes the number of such neighbor pairs of entity i then the clustering coefficient of any entity i of order x is given by equation 2.22. Then the 1st order clustering coefficient of entity i indicates the fraction of neighbor pairs of i that are directly connected to each other.

$$C_i = \frac{E_i(x)}{\binom{k_i}{2}} \tag{2.22}$$

2.5.3.2 RAZAVI

Razavi *et al.* [RRM09] proposed a trust model with indirect trust calculation. The authors proposed a calculation model where different recommenders have different weights that can be mentioned as their trust values. Their model judges recommenders according to their honesty and context, where recommenders which are more trustworthy have more effect in computing the trust value of the recommended entity (represented by equation 2.23 as the initialization function of a indirect trust value).

$$Rt = \sum TRV_i/n, \text{with } 0 \leq i \leq n \tag{2.23}$$

where Rt is the recommender's trust value, n is the number of records in trust records database that their service provider identification is the same as the recommender's id, and TRV_i is the trust value of the i^{th} trust record. Also recommenders will be updated after each interaction with the corresponding recommended entity. The similarity distance between the provided value and the real value is computed as in equation 2.24.

$$\delta = \sum |PV_i^{norm} - RV_i^{norm}| / n, \text{with } 1 \leq i \leq n \tag{2.24}$$

where δ represents the similarity distance (the shorter the distance means the more accurate recommender). PV_i^{norm} is the normalized provided value. and norm RV_i^{norm} is the normalized recommended value for i^{th} attribute and n is the number of attributes.

The update factor for a recommender's trust value is computed as equation 2.25, where $error_{acpt}$ is the acceptable error between provided and recommended values. UF is the update factor for the recommender's trust value. Considering this, a recommender's trust value is as in equation 2.26.

$$UF = 1 - (\frac{\delta}{error^{acpt}}) \tag{2.25}$$

$$RT^{new} = \begin{cases} +1 & \text{if } (1+UF).RT^{old} \geq +1 \\ -1 & \text{if } (1+UF).RT^{old} \leqslant -1 \\ (1+UF).RT^{old} & Otherwise \end{cases} \tag{2.26}$$

where RT^{old} is the old and RT^{new} is the updated value for a recommender trust. UF is the update factor and a recommender's trust value will be increased in the case of having less error, considering that an unaccepted error causes the recommender trust value to be decreased. Finally, the indirect trust value for a recommended entity is computed as in equation 2.27.

$$It = \frac{\sum(RT_i.TRV_i)}{\sum RT_i}, \text{with } 1 \leq i \leq n \tag{2.27}$$

where It is the indirect trust value for the recommended entity, n is the number of recommenders for that recommended entity, RT_i is the recommender's trust value corresponding to i^{th} recommender, and TRV_i is the trust value which is recommended by the i^{th} recommender.

2.5.4 Cognitive Trust

Cognitive trust is a confidence to rely on other's competence in providing a service. Normally, it comes from an accumulated knowledge that allows one to make predictions, with some degree of confidence, that a focal entity is likely to fulfill his obligations.

Although cognitive trust is knowledge-driven, the need to trust presumes a state of incomplete knowledge. If one considers that a state of complete certainty regarding an entity's future actions implies that risk is eliminated, thus trust is redundant [JG05].

Bellow there are some examples and further discussion about cognitive trust.

Bellow there are some cognitive trust discussions.

2.5.4.1 FALCONE & CASTELFRANCHI

Falcone and Castelfranchi [FC01] provide a definition of trust as a mental state and as a social attitude and relation considering MAS. They believe there is a relation between trust and the mental background of delegation, thus trust is a bet and implies some risks. Their model analyzes more complex forms of social trust, based on a theory of mind and in particular on morality, reputation and disposition, and authority (three party trust).

Falcone and Castelfranchi [FC01] consider that promises, contracts, authorities can increase trust by modifying mental representations and presents a principled quantification of trust, based on its cognitive ingredients, and use this "degree of trust" as the basis for a rational decision to delegate or not to another agent. Considering such aspects, the authors then present a cognitive model of trust in term of necessary mental ingredients (beliefs and goals) and decision to delegate.

Falcone and Castelfranchi [FC01] agree with the trust definition given by Gambetta [Gam00], but they argue that it is also quite a poor definition, because it ignores the competence dimension of trust. The subjective probability definition of trust does not consider many important parameters and beliefs, which are very relevant in social reasoning, thus to cognitive trust.

Trust considering a cognitive approach [FC01], require 3 elements: belief, wish and intention. Thus, the authors proposed a model where an entity **A** trusts entity **B** about a particular action, which create a state **S**. This model considers that trust is a mental state or attitude, which results in the action of delegating part of **A**'s plans to **B**. So, basically, there is a division in two distinct components. The first one creates a characteristic of the trusted entity (internal trust). The second one considers that there is an evaluation about the probability with external factors' consistency, barriers and opportunities that generate an external trust, among other external factors.

2.5.4.2 JOHNSON & GRAYSON

Cognitive trust has a tied relation to trust, considering the view of Johnson and Grayson [JG05]. Authors work with the proposition that consumer trust in service providers has distinct cognitive and affective dimensions with unique antecedents and consequences. Thus, their work [JG05] evaluates whether these dimensions are empirically supported, and what practical insights can be gained on managing interpersonal aspects of service relationships.

Cognitive trust provides a base for affective trust, so it should exist before affective trust advances. But as affective trust matures, the potential for decoupling of trust dimensions and reverse causation increases. [JG05] model cognitive trust as a positive antecedent of affective trust (relationship between cognition and affect in attitude formation is bidirectional). Their study has produced evidence in favor of conceptualizing trust as having cognitive and affective dimensions. Though the two dimensions are highly correlated, they are empirically distinguishable, and both dimensions of trust have unique antecedents.

2.5.4.3 DUNN

The report presented by Dunn [Dun00] shows the results of two experiments on the importance of three trust indicators on establishing interpersonal trust. A cognitive element of trust is the result of a rational calculation by the trustor about how the trustee will behave in the future. To Dunn, most social relations are based on cognitive trust, where as emotional trust is the basis for intense personal relationships (i.e love and friendship).

Dunn's study [Dun00] focuses on three cognitive-based cues that engender trust within a dual relationship. The first is the frequency with which the trustee and trustor interact. The second is the competence of the trustee. The third is the consistency of the trustee's previous behavior. In his experiments, trust is measured as the perceived reliability of the trustee's information and the estimated time for the trustor to complete his task and the results reveal that the consistency of the trustee's previous behavior is the most important element in engendering cognitive-based trust within a dyad.

2.5.4.4 ALI & LUDWIG

To Ali *et al.* [ALR05] the socio-cognitive trust is drawn by characterizing the known motivations of the other entities. This involves forming coherent beliefs about different characteristics of these entities and reasoning about these beliefs in order to decide how much trust should be put in them. So, the action of deciding to trust is the cognitive process of selecting a course of action from among multiple alternatives mutually exclusive actions, where one and only one choice can be made.

Considering this point, Ali *et al.* [ALR05] argues that each of these choices might have one or more possible consequences that are beyond the control of the decision maker, which again are mutually exclusive. The rational procedure in such a situation is to identify all possible outcomes, determine their values (positive or negative) and the probabilities that they will result from each course of action, thus multiply the two to give an expected value, where the action to be chosen should be the one that gives rise to the highest total expected value.

The cognitive process of making a judgment to assign a set of qualities for the experience that the user had with the service in a transaction is defined as evaluating a transaction with a service, so when a user **A** requests a task **T** from a service **S**, the quality of completing the task reflects **A**'s experience with **S**, and as an experience, the qualities drawn from that experience are important to **A**. In the context of Ali *et al.* [ALR05] model, the quality of experience and the quality of compliance are metrics used. The quality of experience is used as a feedback to the user's own experience registry while the quality of compliance is used as a feedback to a public compliance registry.

In short, the model proposed by Ali *et al.* [ALR05] considers a trust management model constituted of four phases regarding trust management: i) capturing user's trust disposition; ii) verifying trustworthiness level; iii) making trust decision; and iv) evaluation after a transaction has taken place.

2.5.5 Situational Trust

Situational trust is basically understood as a situation where x trust y in context α and it can be expressed as a function $T_x(y, \alpha)$ with values [-1,+1). Nowadays, there are several formalizations of trust in the sense of cooperative situations. In short, given the situation between cooperation or non-cooperation, an entity decides whether to cooperate with a particular entity or not. Below there is a review of some situational trust models.

2.5.5.1 MARSH

A metric defined by Marsh [Ste94] towards situational trust is considered one of the first computational formalization of trust. In order to simplify his approach, Marsh considers there are two or more entities involved in a cooperation process and an entity decides in which to trust in the particular situation. His formulation uses the estimation of situational trust as first parameter and a cooperation threshold as second parameter, as expressed in equation 2.28.

$$T_x(y, \alpha) = U_x(\alpha) * I_x(\alpha) * \widehat{T_x(y)} \tag{2.28}$$

where $\widehat{T_x(y)}$ emphasises that x can use previous trust-based knowledge in y in this calculation, whether related to the particular situation or not. This gives the perspective that x has this much trust in y. One could not forget that if it is for a different situation, where the situation is more important to x, the values may be very different. To Marsh [Ste94] the situation extends when x has to decide how much trust in y to start the cooperation. This generates the cooperation threshold expressed by equation 2.29.

$$CT_x(\alpha) = \frac{PR_x(\alpha)}{PC_x(y, \alpha) + \widehat{T_x(y)}} * I_x(\alpha) \tag{2.29}$$

where $CT_x(\alpha)$ represents the cooperation threshold of x in the context α, $PR_x(\alpha)$ is the perceived risk seen by x in the context α and $PC_x(y, \alpha)$ is the perceived competence seen by x about y in the context α. According to Marsh, this gives a means of seeing what is necessary for x to accept any cooperation with the help from y in the situation α. Then, if $T_x(y, \alpha) \geq CT_x(\alpha)$ this implies that x will cooperate with y in context α.

Still, to Marsh [Ste94], when trust is encouraged then it shall be strengthened, while when betrayed the trust shall be weakened. Then, if x helped y in the past, and x responded at this time by defecting, the trust y has in x will be reduced by a considerable amount. Otherwise, if x helped y in the past, and x reciprocated at this time with cooperation, then the amount of trust y has in x will remain the same or increase only by a small amount.

2.5.5.2 REHAK

RehÃ k *et al.* [RGP06] created a model for situation representation to be used by software agents. They argue that general trust models are inappropriate and their use for advanced decision-making is limited due to the lack of detail to be used in complex

environments, thus the authors proposed a framework for context representation in trust model.

In the proposed model, there is a set of agents in a community (A, B, C, ...). The general trust, without context, is represented as $\Theta_A(B)$. A context C is a metric of spaces contexts with distance function as $d(c_1, c_2)$. The context of each trust observation $\tau_A(X|c_i)$ or trusting decision $\delta_A(X|c_i)$ represented by exactly one point in the space, but several distinct observations may come int the same point c_i in C. A represents the observing or deciding agent, X denotes the evaluated agent and c_i is a context of the decision.

To RehÃ k *et al.* [RGP06], the context space is formed considering: **i)** identify all relevant features of the environment; **ii)** define the Q-dimensional context space where each dimension q matches a relevant feature of the trusting environment; **iii)** for each dimension q, define its quantification (either discreet or continuous) and appropriate distance metric d^q that correctly represents the feature; and **iv)** defines a joint metric d on the full space C, taking into considerations the domain characteristics and marginal metrics d^{i1}.

Authors also consider that the general update approach proposed by their model ensures that several trustfulness relative to different contexts are updated simultaneously. They assent this as a critical feature for any trust model, once it reduces the time before the model can provide meaningful results. Also the decision-making process proposed gathers data from all relevant contexts, increasing the quantity of the information the trusting decision is based on.

Based on all the metrics proposed, the authors [RGP06] concludes that specialized agents can successfully rely on general trust models, once there is the considerations of the tradeoffs of specific decisions. In their model this is addressed using a generic trust update method and trustfulness aggregation method using the general set of reference contexts.

2.5.5.3 SCHULTZ

In the perspective of the work presented by Schultz [Sch06], a trust model for situational trust is used to communicate a clearer and better understanding of trust. His work includes a situational trust model, a trust transaction, and a trust equation. The situational trust model relates to the trustor, the trustee, the trust object, and the trust environment.

When considering a trust approach, the trustor decides cooperatively to accept an exposure by relying on the actions of a trustee, and the decision to trust depends primarily on previous experiences formed by interaction and communication. Schultz considers Figure 2.4 as the single transactional display of any trust situation, with the reference of trust formation to prior trust situations.

Considering such aspects, trust is situation specific and depends on the experience gained over time, thus it be expressed in its most elemental form as equation 2.30.

$$V_{t+1} = v(V_t) = v(s, x, t) \tag{2.30}$$

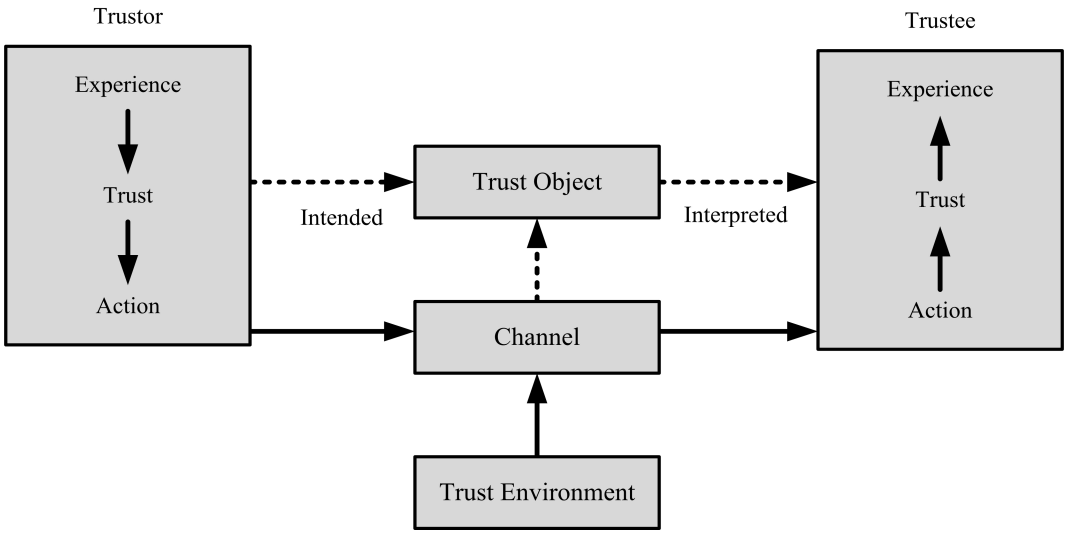


Figure 2.4: Transactional display of any trust situation (adapted from [Sch06])

where the trust V of a trustor is formed by the trust decision process v according to the given situation s and the experience x obtained over the time t. Than the components of the trust model were included into the trust equation transforming the expression into the equation 2.31.

$$V_{t+1} = v(s, x, t) = v(T_{s,x,t}, O_{s,x,t}, E_{s,x,t}) \tag{2.31}$$

where the trust V of a trustor is determined by the trust function v operating on the trustee T, the trust object O, and the trust environment E in accordance to the situation s and the previous experience x up to the point in time t. Following, equation 2.32 shows the association between the general trust function and the different type of trust formation.

$$v(s, x, t) = d(x, t) + i(s, x, t) + p(s, x, t) \tag{2.32}$$

where the trust function v can be broken down into dispositional trust d, institutional trust i, and interpersonal trust p. Then, the complete trust equation can be expressed as equation 2.33.

$$V_{t+1} = d(E_{x,t}, O_{x,t}, T_{x,t}) + i(E_{s,x,t}, O_{s,x,t}, T_{s,x,t}) + p(E_{s,x,t}, O_{s,x,t}, T_{s,x,t}) \tag{2.33}$$

where the range of the functions d, i, and p are not limited to positive values, and the trust equation can account for an increasing, a decreasing or a constant level of trust development from t to $t+1$. Schultz [Sch06] understands that the complete trust equation (2.33) represents the gradual development of trust over the course of previous transactions.

2.6 Synthesis of This Chapter

This chapter reviewed trust and its applications in CSs. Basically, trust is widely used in distributed systems to help address some problems that lead to information security and to the reliability of such systems.

When emphasis is given to trust, one realizes that it has different lines of understanding because of its subjective evaluation. Trust is subject to social evaluation, suffers influence of time, and it is context and behavior dependant. Considering the perspective of the review of this work, Table 2.6 provides a summary of the basic requirements of trust model applied to CS ([Pat07], [Ste94], [Gam00], [Lam01], [Das00]).

Following the organization of this chapter, section 2.1 presented some definitions of trust and some important work in the literature that describes trust in different perspectives. Discussions about context, third party information and trust's basic characteristics were also discussed (sections 2.2, 2.3). After that, in section 2.4, it presented a review of trust applied to computational systems, such as cloud, grid and P2P. Afterwards, some trust models used in CSs were reviewed and presented in section 2.5.

Table 2.6: Basic requirements applied to trust models

#	Requirement	Description
1	Be scalable.	It shall not have performance issues and shall not be restricted to the number of entities that are part of the process of generating trust or reputation.
2	Be decentralized.	It must be robust and function even in failure situation in the network.
3	Be able to distinguish between an entity from environment.	It must be able to make distinction between the role of an entity and the role that is seen by an entity in the environment.
4	Calculate direct trust.	It must allow an entity to perform trust calculation of another entity supported by past experiences.
5	Calculate reputation.	It must allow an entity to perform trust calculation based on the opinion of other entities about a particular entity.
6	Find reputation sources.	It shall offer tools to help identify sources of general opinion besides specific protocols.
7	Encouragement the sharing of opinions.	To address requirement #5, an entity shall provide reasons so another entity may share their opinion about a particular entity.
8	Adjust opinions.	It shall provide tools to make opinions' adjustment in order to avoid false opinions.
9	Store historical data.	An entity shall be able to store past interaction information so it can adjust the trust values based in new and old trust and reputation information about a particular fact.
10	Use social information.	Depending on the environment, the use of social information may help in defining trust values.
11	Dynamic trust value.	The trust value shall be adjustable depending on the information of an entity about another entity. If the opinion changes, so shall be the trust value depending on the situation.
12	It shall be able to generate consensus about trust level.	It shall allow a group of entities to reach consensus about the trust level of a particular member of a group, be it internal or external.
13	Group trust level.	In some situations, it shall be necessary evaluate or calculate a group trust level.
14	Effective opinion exchange.	It shall permit entities to rapidly exchange information about another entity.
15	Share trust information.	Considering requirement #2, trust information must not be kept in only one central point.
16	Context dependent.	It shall allow entities exchange information about a particular context and the trust values related to it.

Chapter 3

Review about Reputation and Information Security and Cyber Security

This chapter provides the review of reputation, information security and cyber security. As viewed in the previous chapter, the literature points a lot of trust models. Mainly trust and reputation can be analyzed from different points of view and can be applied to many situations. Such characteristics are important because one must consider particularities of the environment where trust and reputation can be developed as a computational process.

To be able to sort or classify trust and reputation models, it is important to consider aspects such as the source of information, if it is acquired by direct or indirect interactions, if it is used in social network evaluation or even if an entity, based on given properties, is part of a recognizable group. With regards to information security and cyber security, it is understandable that these subjects are interlinked, however each has its own meanings and implications as presented by the review in the following sections. The following sections are an important basis for the discussions regarding the architecture proposed and the discussions about exploits and APT.

This chapter is organized as follows. In section 3.1 the definition of reputation is presented, while section 3.2 discusses the reputation in CSs. Some reputation models found in the literature are presented in 3.3. In section 3.4 some of the relations of trust and reputation considering information security are discussed, followed by some review and related work about information security (section 3.5). The discussions and a short review about cyber security are the subjects of sections 3.6 and 3.7. Finally this chapter is summarized in section 3.8.

3.1 Definition of Reputation

Basically, reputation relies on information to choose partners to cooperate with and it has been studied from different approaches. According to the online Merriam-Webster

Learner's Dictionary[1], reputation is:

a. The common opinion that people have about someone or something;

b. The way in which people think of someone or something.

This basic definition points to what is generally said or believed about a person or something. Generally, it implies that reputation is formed by an accumulation of opinions, which makes reputation part of the concept of trust, which in some cases, may even be more objective than trust.

This work regards reputation as directly related to trust. Reputation is an opinion that an entity has over another entity. As it is an opinion, it has many subjective evaluations depending on several factors like:

- Situation observed (context);
- Own trust value inference;
- Information received by others in a social relationship;
- Etc...

The reputation value evolves with the time and depends directly on the behavior of the entity being observed. In some cases, reputation may also represent indirect trust. It includes asking for the opinion of other parties whom the entity has previously had interaction with regarding a third entity.

Reputation can also be defined as the common opinion of others regarding an entity [Pat07] which may be used in the absence of trust formed from personal opinions. Reputation values take time to be acquired, but it can be easily lost in social aspects.

Calculation of reputation values is done using past information that has been obtained over time and based on the information that was received from trusted parties. These variables enable an entity to form an idea about an unknown entity. It could be considered a social evaluation of an individual or group of individuals.

To Dagsputa [Das00], reputation is supported by trust. Reputation should be acquired over time considering behavior under known circumstances. Trust and reputation in this case are subject to expectation, which influences the behavior of an entity before it can even monitor other parties in a communication process.

As stated by Moyano *et al.* [MFGBH14] a good approximation to the relationship between trust and reputation was suggested by Jøsang [JIB07], who made the following statements:

1. 'I trust you because of your good reputation';
2. 'I trust you despite your bad reputation'.

[1]Definition found at http://www.learnersdictionary.com/definition/reputation

Following this analysis [MFGBH14] supports the idea that reputation can be considered as a building block of trust. But reputation does not have the final say as stated by the second statement. This view of reputation allows one to trust someone with a negative reputation or to distrust someone with a high reputation. This is because there may be other elements that shall have greater impact over the trust decision. Such elements can be either the trustor's disposition to believe in the trustee, or the trustor's opinions, or the trustor's personal familiarities with the trustee.

According to Jøsang et al. [JIB07], reputation also corresponds with the view of social network, which states that reputation is a quantity derived from the underlying social network which is globally visible to all members of the network. Following this assumption, reputation can be considered as a collective measure of trustworthiness based on the recommendations or ratings from members in a community.

Reputation can also be related to a group. A group's reputation can be modeled as a view of all its members' individual reputations. In this case the reputation of a group is how the group is perceived as a whole by external parties. Tadelis' study [Tad03] shows that an individual belonging to a given group will inherit an a priori reputation based on that group's reputation. If the group is reputable all its individual members will be perceived as reputable and vice versa.

The work described by Wiseman [Wis08] points that an effective reputation is one that weakens over time. This means that the reputation must be constantly updated by other means (i.e. external components). To Wiseman [Wis08], if reputation is not updated, it will not provide good information, so outsiders or competitors can gain advantages over the situation in context.

3.2 Discussions about Reputation

To Jøsang [JIB07] reputation is understood as a collective measure of trustworthiness, in the sense of reliability, which is based on the referrals or ratings from members in a community. In general, an individual's subjective trust can be created based on a combination of received referrals and personal experience.

Basically, the architecture of a computational system determines how reputation is generated and how it is shared among entities that are participants in a reputation system. This gives two types of reputation mechanisms: **i)** centralized; and **ii)** distributed.

In the first model, information about a given entity is collected as ratings from other members in the community who have had direct experience with that participant. After applying calculation criteria depending on the particularity of the reputation model, the information is then stored in a central repository, where it can be consulted by any participant of the community.

In the distributed approach, there is no central location for submitting ratings or getting reputation scores. Basically, there are distributed repositories where ratings can be submitted, or each entity simply records the opinion about each experience with another entity, and provides this information on request from relying parties.

Following this discussion, there are several models for representing reputation in the

view of Rosaci *et al.* [RSG12], and a trust relationship can involve multiple dimensions, depending on the particular perspective under which the interaction among entities takes place.

According to the review provided by Hendrikx *et al.* [HBC15], reputation is context dependent. It also relies on contextual information to give meaning to the data. One problem is that context in reputation systems is not easily defined because additional information, such as contextual dimensions, attributes or facets, are used to provide greater meaning and usability to the information generated during a transaction that uses reputation. Normally, when reputation systems use more than one context, it often adds additional domains of information.

To Vavilis *et al.* [VPZ14], reputation has an important role in the process of trust establishment and management. Usually, when an entity needs to make a trust decision whether or not to engage in an interaction with another entity, he relies on reputation systems as a source of information. Therefore, a reputation system becomes a fundamental component of the trust computational systems.

In general, reputation may represent indirect trust (section 2.5.3) because it involves asking for the opinion of other parties whom the entity have previously interacted with about a third entity. Reputation is also the common opinion of others regarding an entity, which may be used in the absence of trust formed from personal opinions.

Reputation is rarely extreme (all or nothing). It takes time to be acquired, but it can easily be lost in social aspects. Calculation of reputation values may be done using previous information that has been obtained and is based on the information that was received from trusted parties. These variables enable an entity to form an idea about an unknown entity.

If trust and reputation models can be characterized as a cognitive approach, they are supported by beliefs and they are a function of the degree of these beliefs. Towards this cognitive approach, if an entity has the ability to trust another entity and assign a reputation value, there are consequences based on the state of both entities and the act of relying on each other. Otherwise, if the probabilistic approach (A expects that B will perform a given action) is considered, trust and reputation are not just the result of a state of the agent in a cognitive sense. Rather it is seen as the result of a more pragmatic approach that is able to consider past interactions.

3.3 Reputation Models

In a larger view, reputation is a social evaluation of an individual or group of individuals in direction of one entity or another group that have impact in trust formation. The following subsections provide some reputations models found in the literature.

3.3.1 Online Reputation

Normally, these are the reputation models mostly used in e-commerce solutions. Online markets, online stores and e-commerce sites such as *eBay*, *Amazon* and *OnSale* use

reputation mechanism to qualify their online users. Online reputation is also used to track or monitor information about an entity.

The normal approach in e-commerce considers that after a transaction is complete, the buyer has the possibility to rate the seller (give its opinion about it). The normal procedure constitute of rating possibilities which vary giving good or positive information, neutral, or bad or negative. Such values are processed by the online site and this provides a general reputation value, which is publically available to any interested entity in the system. Thus, it may provide users with unreliable information and this may generate unwanted relations. Such online reputation systems have a very simple implementation and are designed to have a very intuitive understanding, making them ideal for human-based application. But, considering this simplicity, they also lack measures to infer strength against false information or even cheating. This problem may lead to misinformation to users with no immediate effect on the system.

It is also important to consider that the increasing of reputation in online transactions, systems that use reputation as a parameter has become a widely used method for creating trust among the transactors and providing incentives for good behaviors in online transactions. The main problem is that false reputation affects the effectiveness of such systems, because online environments are more open to attacks on reputation systems where, i.e, a single comment can have a huge impact on an entity's online reputation.

When dealing with online reputation things are difficult to measure in terms of impacts considering negative information, since the spread of bad comments can happen anytime and anywhere and the free flow of endless unfiltered information makes it nearly impossible to control the content. This is particularly dangerous for organizations' online reputation, once the spread of unfavorable information is harmful and is hard to control ([Kiz15]).

3.3.2 FIRE

Fides and Reputation (FIRE) is a trust and reputation model proposed by Huynh *et al.* [HJS06]. It was conceived to a multi-agent environment where agents are assumed to be benevolent and honest in exchanging information. FIRE model takes in account information about the direct agent's experience, the agents' relationships; the verification about the behavior of an agent and the reputation about an agent that is witnessed by a third-party suggested by the agent itself.

Shergill and Kaur [SK15] considers that FIRE defines certified reputation as an additional source of trust information once it integrates sources of information and is able to provide trust metrics in a wide variety of situations.

The basic functioning of FIRE stats when agent **X** is trying to estimate the reliability of agent **Y**, but when there is no information that can use in its trust computation, **X** asks **Y** to provide ratings from its earlier experiences. **Y** then provides **X** with a set of proficient ratings, which it has collected from asking others to estimate its performance at the end of an interaction. This means that **X** can ask **Y** to supply it with ratings of **Y**'s past exchanges without having to look for a huge social network. It is assumed by the model that a security mechanism is present in it that prevents agents from tampering with these certified ratings ([SK15], [HJS06]).

3.3.3 REGRET

Hendrikx *et al.* [HBC15] argues that the Reputation Model for Gregarious Societies (REGRET) [SS02] is a modular reputation and trust system for cooperative agents exploiting impressions about other agents derived by both direct trust and a reputation model. REGRET uses the concept of a compositional value where three proportions of reputation (ontological, social and individual) are considered in the process of generation reputation.

To REGRET [SS02] trust is a function of direct trust, only calculated through direct experiences and reputation. The incorporated reputation model uses transmitted information, social networks analysis, system reputation and prejudices, which are used to infer reputation values of unknown agents from their belonging group. The model also incorporates a credibility module to evaluate the truthfulness of witness information, which takes into account the reputation and trust of the information provider. It offers reliability measures for trust, reputation and credibility values. [HBC15] consider that REGRET can be adopted to produce good results because the model deals with lots of issues of trust and reputation in virtual societies.

3.3.4 TRR

Rosaci *et al.* [RSG12] developed Trust-Reliability-Reputation (TRR), where a software agent is able to represent both the reliability and the reputation of another agent, and to merge these two measures into a global trust evaluation. According to the authors, the TRR is an evolution of previous mode Reliability and Reputation Agent Framework (RRAF) and TRR uses a dynamically computed weight, representing how the agent **X** considers the reliability important with respect to the reputation when it computes the trust in agent **Y**. The weight considered as a metric depends on the number of interactions between **X** and **Y** and the expertise that **X** has in evaluating the **Y**'s capabilities.

Authors argue that an important feature of TRR is what represents the mechanism for computing the reputation. In the case of TRR reputation perceived by an agent **X** about another agent **Y** is based on the global trust that each other agent **Z** of the community has in **Y**, such as the overall trust measure θ_{XY} will depend on the other trust measures θ_{ZY}. Authors consider this approach more efficient than that used in RRAF because it uses the actual trust measures of the other agents as suggestions for the agent that is computing a trust measure, instead of suggestions arbitrarily provided.

3.3.5 SPORAS

Another review of a reputation model called SPORAS was done by Shergill and Kaur [SK15] as a way to improve online reputation models. SPORAS [ZM00] considers that when a newcomer enters the online community it starts with a minimum reputation value which is updated as a result of their activities in the entire system and an individual's reputation cannot gets below the point of a newcomer. This characteristic of SPORAS considers that an individual never has to leave the system and re-enter under a new

distinctive identity. The minimum reputation level is then updated after each interaction in the system.

In SPORAS, each entity can only rate another entity just once, and when it happens that an entity has rated another more than once, the most recent rating is used. Also the amount of an entity's reputation level is not only dependent on feedback, but also on the present reputation level. Thus, entities with very high reputation values experience smaller changes in the rating mechanism after each update than entities with low reputation [SK15].

Another characteristic of SPORAS is that the ratings used to compute reputation are discounted over time. This means that the latest ratings have more weight. Authors [ZM00] argues that it is an easy and effective way to avoid problems related to trust and reputation, exhibited by the dynamic behavior of entities. This way, by increasing the weight of more recent ratings the value of reputation obtained will represent a truer illustration of an entity's current behavior.

3.3.6 QADE

Zupancic and Trcek [ZT15] believe trust is essential to economic efficiency and points out that one way is to deploy trust and reputation management systems that are based on collecting feedback on entities' transactions. But the presence of unfair ratings is a problem in trust and reputation solutions. Based on these assumptions, they proposed *QADE* as a trust and reputation model for handling false trust values considering subjectivity, which assumes the existence of unfairly reported trust assessments.

To them [ZT15], the nature of differently reported trust values do not necessarily mean false trust values, but it can also imply differences in dispositions to trust, based on the considerations of the subjective characteristic of trust. In their proposed method a trust evaluator finds other entities in a society that are similar to it and takes into account pairwise similarity of trust values and similarity of entities' general mindsets. As defined by the authors, their model reduces the effect of unfair ratings by the exclusion of non-similar entities from the trust computation.

3.3.7 TORMO

The proposition of a dynamic and flexible selection of a reputation mechanism for heterogeneous environments corresponds to the work of Dólera *et al.* [TMP15]. Basically, authors consider that current trust and reputation management approaches usually offer rigid and inflexible mechanisms to compute reputation scores.

Most systems provide certain parameters that are configurable or tunable, but authors argue this is not enough for such heterogeneous and dynamic environments as the ones introduced by Internet of Things (IoT). Thus, [TMP15] designed and prototyped a TORMO as flexible mechanism to select the most suitable trust and reputation model to apply on-the-fly, amongst a pool of predefined ones, considering both the current system conditions and the selected performance measurements of the environment.

Dólera *et al.* [TMP15] claims their solution guarantees a smooth transition between

different computation engines and it avoids abrupt changes in the computed reputation values. The experiments' results show the proposed solution is able to identify and start up the most suitable trust and reputation model depending on the current system conditions and expected performance measurements.

3.4 Relations of Trust and Reputation and Information Security

The discussions about the relations between trust and information security is not new. Still it is subject to researches and definitions, which tries to make better assumptions about security in computational systems. The view of information security as a risk management activity has to deal with a lot of different aspects where trust is understood as a manner of helping to protect information.

The Zero Trust Model for Cybersecurity [Con13] is an information security architecture that claims network packets should always be considered untrustworthy. The basic view is that the data and information that goes across a network should be protected the same way, be it internal or external data in the networks.

Information security and trust are pieces of computational systems that influence interactions and behavior between entities. Trust also deals with the sense of safety and security [Kie14]. Safety in the sense that a system has no possibility to enter a state or to create an output that may impose harm to its users, the system itself or parts of it, or to its environment. Security in the sense of there is very low or even complete absence of possibilities to defect the system in ways that it disclose private information, or the system may change or delete data without authorization, or to misuse the authority to act on behalf of others in the system.

Security mechanisms are used to provide protection against malicious actors, be it human, hardware or software. What is seen in literature is that traditional security solutions will protect resources from malicious parties by restricting access to only authorized users. One basic problem of the relation of trust and information security, is that the view should be reversed. This is because information providers can act deceitfully by providing false or misleading information.

Seen by this perspective, traditional security mechanisms are unable to protect against this type of threat, but trust and reputation systems can be used in this situation to provide means of enhancing security. This therm is also called soft security because it uses social control mechanisms (trust and reputation) to increase the perspective of security in the systems [JIB07].

Trust is important regarding information security because it can be used to ensure secure and reliable communications. The accuracy of trust evaluation depends whether a trust system can function properly under all situations and be capable of handling unfair rating attacks in a satisfactory manner [WMI+15].

In order to accomplish security related tasks in distributed systems trust is also a possible solution because the management of trust relationships between different peers can be done using different approaches to help increase the security perspective of a

computational system. Considering a general view, information security must support business objectives by minimizing risks and developing trust.

Even though humans use trust every day in social aspects, work, home, etc., trust is not widely used as a computational resource in network and information systems yet. Normally, trust is applied in a small set of systems and particular environment. The process of trusting depends on the knowledge the trustor gathers about the trustee and it is not a good approach to trust something without being able to measure it or to fully understand it.

Anyway, trust is indeed an expectation. It is a probability that things will work and keep working as they are supposed to. But when it comes to security, trust should be zero or one, i.e., one trusts an information system, network, etc. or does not. If security is important, the "maybe" approach should be avoided.

If something is to be trusted, it must be clearly identified and operate exactly as planned and expected. It must also not do anything it was not supposed to do and must be able to operate continuously. If it is acknowledged that a system has been or may be compromised, this is enough to make it suspicious or untrustworthy, and consequently not secure.

3.5 Review and Related Work Regarding Information Security

A lot has been said about norms, compliance, standards and frameworks regarding information security and the importance of following some of those. Commonly, security standards are guidelines to develop and maintain Information Security Management Systems (ISMS), as seen in works like [BMG02], [Pel13], [Dis13], [Sta11] and [PVS04].

Information security must support business objectives by minimizing risks and developing trust. In addition, it is important to understand that information security requires continuous improvement in a cyclic approach [Dep07]. Standards such as BS7799 and ISO 27000 Family [WM13] are well known guides in the information security business. But they are more related to telling what to do instead of how to do it.

Frameworks such as Information Technology Infrastructure Library (ITIL) and Control Objectives for Information and Related Technology (COBIT) ([POK13]) are also used in information technology governance in order to help organizations in reducing costs and increasing productivity, and in some aspects, aiding security in terms of organization and methodologies [Dep07]. The review of such methodologies shows that none of them presents the importance of trust.

However, ensuring compliance supported by standards does not guarantee security at all. To deal with information security it is required to go beyond compliance or best practices. They do help, but they are not the most effective way of getting things secured. Up until now and as far as this research is aware, there is no known proven technology or framework for developing application and information systems without security issues, such as exploitable flaws, configuration errors or system misuse.

Discussions and references precedes to the eighties considering the information security

CIA triad. The main discussion about confidentiality dates back to 1974, maybe earlier, with Bell and LaPadula [BL73] and maybe earlier. Regarding integrity, references quote Biba [Bib77] and his model that describes a set of access control rules designed to ensure data integrity. Regarding availability, some references are found in the National Institute of Standards and Technology (NIST) according to Burrows [Bur83]. The exact origins of the 'CIA Triad' expression appears to be unknown, but apparently they date back to the nineties NIST publications.

Regarding information security architectures, the Zero Trust Model for Cybersecurity [The13] conveys a very simple message: stop trusting packets as if they were people. The idea behind this model is that the concept of internal and external networks should be changed because one should assume that all network traffic is untrustworthy. In practice, Zero Trust claims that one should protect internal data from insider abuse as one protects external data in public networks.

In [JYJ10], the authors discuss the general effect of risk management that is limited to security activities regarding an information security architecture on information assets for an organization.

The work proposed in [Tan14] describes an information security architecture with three levels: domain, component and control. Their target is to align security and business, and customer service and manage security, and to create traceability between them, proactively and predictively.

According to [ASS07], it is needed to develop and improve information security in both administrative and organizational levels. Then, a holistic view of information security is proposed based on a survey on literature along with a specific Swedish model. According to the authors, they have extended the Swedish model based on concepts of the semiotic theory and on the perspective of an information system that included technical, formal and informal sectors.

Behavioral information security shifts the way people are related to information security ([CJL+13],[AP12]). People have become the center when talking about and dealing with information security, a standpoint that this research agrees with.

Much more has been discussed when it comes to the ability of representing information. In technology, if information can be retrieved it is because it has somehow been digitally represented before [Chu03]. Despite the amount of effort, it is still rather difficult to represent and retrieve the correct information for the correct user at the correct moment considering the amount of variables that may be included or that could be related to the subject being searched.

The web provides information from a variety of data sources that have great potential for knowledge discovery [NSSV13]. One of the long-standing problems is that search engines retrieve plethora of data, however, most of it is useless depending on what the user is interested in or searching for. Thus, giving meaning to data is still rather difficult. Yet, when it comes to protecting such information during transferences, mechanisms such as Secure Socket Layer [FKK11] or Transport Layer Security [Die08] are used very often.

Considering such aspects and in order to protect information, it is very important to understand how it is digitally represented and treated. For example, from a users'

perspective, a piece of information can be text, image or both formats. The internet has redefined the way information is represented and how it is retrieved [DL07] and as a consequence, the way it is treated significantly changed too. Furthermore, information representation needs extra structural or semantic complements, which transform raw data into something intelligible to people. Researches on the subject such as ontology, semantic web, etc. are evidences of that.

Now that we went over these issues, it is concluded that existing security architectures fail to manage risks, policies, people and assets correctly to secure information. As a contribution, this work proposes a layered information security architecture as a way to see how the components are connected regarding information security. In addition, this layered view is important to depict how all the security architecture elements interact with one another.

3.6 Discussions about Cyber Security

Cyber security nowadays is a fundamental challenge to any country or organization. It has to deal with threats that are mostly unknown to ordinary people. It is strongly connected with strategic interests and information security. It also has to consider communications technology as Internet, cellular networks, satellite communications, critical infrastructure systems and so on. In order to manage such a huge area it is important to have security guidance, policies, collaborative work and continually reviewing processes that are subject to such activities.

In a wider perspective, cyberspace should be protected from abuse, misuse and malicious activities. Governments also have a major role in guaranteeing rights online in such environments. But, at the same time, the Internet should maintain reliability and interoperability where most of it is controlled by private sectors of society.

When the attention comes to trust in the cyber security, as noted by Geer [Gee14], trust will simply end as surveillance activities takes place and keep growing in scales that only Government actors are able to tell. Still, after Snowden's revelations [GMP13], much work has to be done and one cannot deal with information in cyberspace naively.

According to Menn [Men15], the cyber security industry is a US$ 71 billion market in 2015. It is fragmented along geopolitical lines because security companies chase government contracts, but they also share information with security agencies. If it is considered only the technical aspects regarding information security, the power behind security agencies and information security systems only experts were able to understand before. There is new technology being developed and deployed rapidly. There are new companies concentrated in protecting user data from surveillance. There are frameworks being revised and new ones being developed with intentions to guarantee that user data is safe and cannot be understood by third parties if captured.

Regarding compliance standards, it is important to remark that some sectors think that if you are information security compliant you are safe. From the perspective of this research it is not. It means that because you are in compliance with a particular security standard, it does not mean your information is safe at all. Upon further consideration, it

might allow you to be more mature in terms of information security, and that is all.

The world is dealing with cyber security strategy from a long time. European Union Agency for Network and Information Security (ENISA) [Eur15] has information that some countries are considering cyber strategy planning since the year 2000.

Thus the review of documents provided by ENISA [Eur15] shows that the cyber security field is no place for isolated actions. It requires strategic planning and specific goals. It considers that the study of technology is mandatory and it is in constant evolution. In such cases, past plans may not be appropriate anymore and should be constantly revised.

The cyber security field is moving constantly, be it by the development of new technologies or by way of protecting them or simply by way of deceiving them. Anyway, the cyber security research field needs focus, analysis, constant timepiece, advanced training and skilled people. And this is all part of a strategic view.

As an example of how the cyber security is being held with deep interest of nations, news [Mil15] show that the Central Intelligence Agency is planning to change its structure to put emphasis on cyber security and cyber espionage. It is no secret that countries such as China, Israel, France, Germany, Spain, England, Russia, North and South Koreas, etc. also pay close attention to cyber security.

Another important point to mention is that much of this attention has relevance not only in protecting themselves, but also dealing with offensive measures in cyber space. Cases such as BelgaTelecom, Sony and American health companies are examples of that. Thinking about offensive security, it is more about using hackers to pentest boundaries and defense perimeters to test your own systems. But it is also about using skilled people to gain control over your adversaries' systems. It is a double edge sword.

3.7 Review of Cyber Security

According to Shah and Mehtre [SM13], cyber security is a bigger problem now than it used to be in the past because of the growth of connectivity through the Internet, the extension of information systems and its complexity. This work adds to that the importance that is given to critical infrastructure systems and the impact that it may have if such systems are misused or abused by unauthorized parties.

If one just considers that hackers will keep hacking and breaches in such systems will keep cropping up, the analysis is quite simple. But things are far beyond that. There are real threats in systems that deal with people's lives in such manners that are beyond control or measure by simple analysis. There is no doubt that efficiently dealing with threats and attacks to critical systems can prevent bigger damages to society and much more efforts are being placed in the cyber security environment.

Still, things are happening in a cyclic view. Users keeps being targeted by phishing attacks and falling on them, not properly locking down sensitive information internally, allowing their users too much access without control measures, unnecessary network services up, and so on. All these things make it easy for targeted attacks to take place with high rates of success. Indeed, taking required measures to safeguard information from

possible cyber attack might not be enough because no matter how much effort you spend in security measures, it may not avoid data theft, systems exploitation, and other offensive measures. It has to be considered that it also depends on the attackers capabilities and resources.

The point is not just about investment either. It is about dealing with security from different points of view, it is about people, it is about technology, it is about continuous monitoring and so on. There must be a balance between perimeter protection, internal controls, auditing and constantly monitoring the technological infrastructure. Most of the time, attackers are already inside and there is not much that can be done on the perimeter to defend yourself. Without humans the most sophisticated technology applied to information security and cyber security may render useless.

If critical infrastructure is on focus, much of Supervisory Control and Data Acquisition (SCADA) systems relies on proprietary networks and hardware [BL04] and thus it has been considered immune to cyber attacks for a long time. As researches keep going, this affirmation is no longer supported and is absolutely unthinkable not to take cyber security measures to protect such systems. SCADA systems are highly customized, so it is unlikely that there are two exactly the same system in production at different sites. So, what is done to protect one system may be completely different to protect another. But it also leads to the point that hacking such systems required specific knowledge and a lot of investment, i.e duplicating the infrastructure, having specific controllers available for testing, and so on.

Safe and reliable operation of SCADA in critical infrastructure systems is a major concern as stated in [TAS+10]. In their work it is studied that schemes based on high measurement redundancy fail in the presence of intelligent and skilled attackers, where they can, for example, send false information to the control center. Their research shows that the more accurate model the attacker has access to, the larger deception attack they can perform undetected. The developed tools can be used to further strengthen and protect the critical state-estimation component in SCADA systems [TAS+10].

In [Har14] there is a conclusion that states that hacking activities, active defense measures and everything in between, have lawful and unlawful impact and all associated risks with deceptive practices, misattribution, and escalation. Tools, technologies, partnerships, and information sharing between corporations, governments, vendors, and trade associations are promising, effective and improving, all that considering a cyber security perspective.

The work discussed in [dOAVRTGdD11] shows that virtualization is a major area when cyber infrastructure takes place. With virtualized systems it is possible to raise and automate the availability in an information technology infrastructure. Aligning server virtualization concepts and infrastructure management tools provide gains in time saving, costs and management when compared to systems without such automation steps.

Thinking about attacks against cyber infrastructures, virtualization is the path used as a way of gaining control over the hypervisor throughout the guest host machine, thus making it possible to compromise a whole infrastructure. According to Wojtczukif [Woj14], if a hypervisor isolates untrusted code that runs in a virtual machine from the rest of the

system, an attacker that exploits vulnerability successfully in the hypervisor breaks this isolation, thus gains access to all the resources available to the hypervisor. Evidence of that can be seen in [Elh11] and [SD15]. Even so, virtualization helps reducing costs, permits scalability [dOAVRTGdD11] and is widely used to build cyber infrastructures.

According to [FSSF15], an APT is a deliberately slow-moving cyber attack. It is applied to compromise interconnected information systems without revealing itself. APTs use a variety of attack to get unauthorized access and then spread itself throughout the network. Also, APTs cannot be detected by protection systems that rely on signature-based methods [GBS14]. So users and cyber infrastructure need proactive defense systems, which have the capability to make intelligent decisions.

When dealing with exploits, sometimes it is not possible to perform comparative approach to rate exploit's authors because normally they will not show the full scope of their knowledge. Basically, an exploit will carry out the minimal necessary actions to accomplish the task of infecting the target with malware [Sza15]. To work with exploits, technical knowledge is required. Once the target is infected, reverse engineering, malware analysis and network behavior, all of them are tools used to try to mitigate and understand what a malware does.

3.8 Synthesis of this Chapter

This chapter reviewed reputation and some common models applied to CSs. Basically, reputation is used in distributed systems to help address some problems of reliability in such systems. The same way as trust, when emphasis is given to reputation, it has different lines of understanding because of its subjective evaluation. Reputation too is subject to social evaluation, suffers influence of time, and it is context and behavior dependable.

Following the review, this chapter also provided some details that serves as fundamentals aspects of understanding information security and cyber security, which are areas of interest in chapters 5 and 6 of this manuscript. Such areas are important in modern society because a lot of productive sectors rely on Internet, thus the implications of keeping the information secure and dealing correctly with cyber space and cyber security has great significance.

Chapter 4

Group Trust Model

This chapter presents the group trust model developed during this research. The proposed model is seen as an extension for supporting groups in trust models. According to the view of this work, this extension enables the definition of trust values over groups and it can be informed as single trust values that represents the group view, so it can be used internally or externally to the group.

This chapter is organized as follows. In section 4.1 there is the initial considerations of the developed model, in section 4.2 there is a discussion about leadership with regards to trust in groups. Section 4.3 presents the developed trust model and 4.4 shows the proposed algorithm used to perform group trust calculations. Then, in section 4.5 there is the testbed characteristics and the description of the environment where the results were achieved and in 4.6 the results and analysis of the proposed model are presented. Finally, in section 4.7 there is a summary of this chapter.

4.1 Initial Considerations

In order to propose a group trust model, first there should be some initial discussions and proper definitions about groups, trust and reputation applied to groups.

Definition 1. A group can be defined as a collection of entities connected together with common goals.

Definition 2. Entities that are part of a group are able to perform specific works in a common context, like service search or service offering.

Definition 3. Entities part of the group are able to perform trust and reputation calculations of each other in the system, considering any given interaction.

Definition 4. Entities may use any trust and reputation model that attends the system needs, and if it is a system requirement or characteristic, the model for trust calculation can be different to the reputation model. In the case that they are different, they have to interact somehow. This means the reputation model may use the values generated by another trust model.

Definition 5. From the perspective of this work there is a trust and reputation algorithm that can perform trust and reputation calculations (1:1), since this work is worried about representation of trust and reputation in the perspective of groups.

Definition 6. Trust and reputation are considered as expectations that entities will fulfill their obligations.

Definition 7. The role of the leader is critical to the system, so it must be defined properly.

Definition 8. The chosen leader must know the contexts the group offers.

Considering the definitions above, group formation has to address situations where an entity, let us say **B**, that belongs to a group, let us say **1**, is able to trust another group **2**. To entity **B**, the members of group **2** are unknown, but entity **B** needs the services provided by group **2** considering a trust approach. The same view is valid if group **1** has to develop trust in group **2**, so group **1** can use the services provided by **2**. It is important to verify that entities must have some sort of unique identification and also has the group, so it can be distinct from each other. Once managing IDs in digital environments goes beyond the problem of this research, this model considers that IDs of the group and of its entities are unique. Figure 4.1 illustrates this perspective.

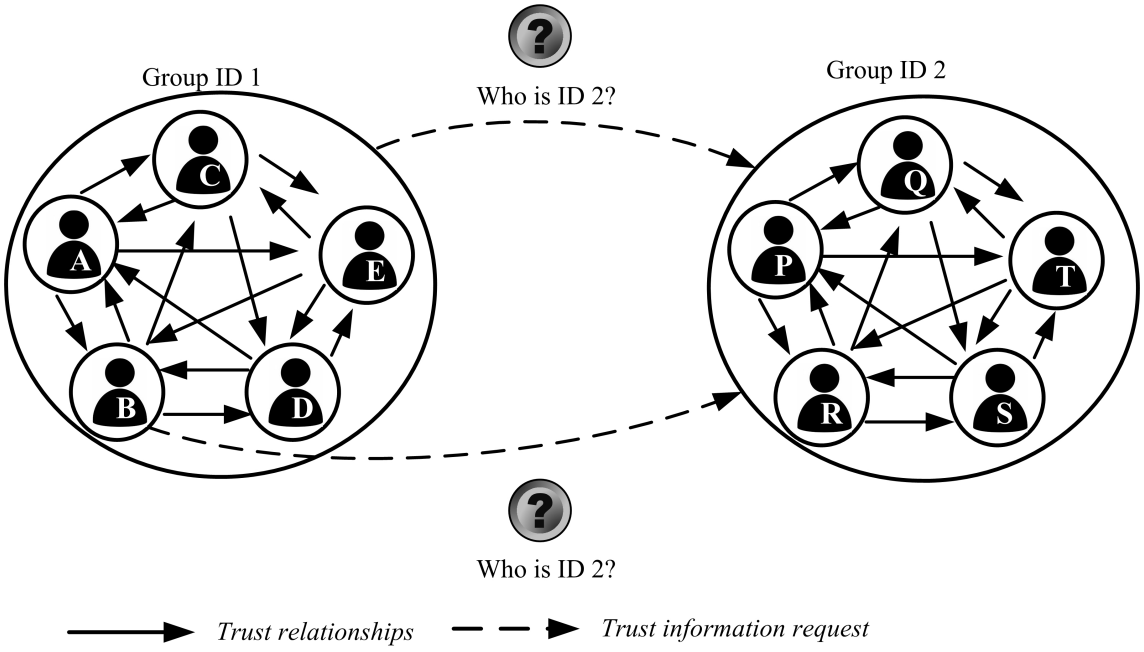


Figure 4.1: Group trust perspective

The normal representation of trust and reputation considers the one-to-one approach (1:1). The perspective proposed by this work brings the view of one single entity to trust a whole group (1:N) and a group trust another group (M:N).

One problem of group trust is that once it is a collection of entities, the trust value of the group has to be a function of trust of each of its members. This brings the problem of collecting such values, processing it and sharing it with whom it may concern. From this, entities have to agree that one of its members can represent the group externally. This work considers that the representation of the group is leadership role and once the leader is selected, this information has to be publically available.

4.2 Group Leadership

Considering group trust and that a leader to represent the group has to be selected somehow, the normal option could be by voting or consensus. If it is considered that a leader in a group is necessary and it must be chosen using trust, there is the need of a process that can check trust characteristics of every member of the group.

If a leadership election process based on trust is needed, then is it desirable to have a trust consensus process because entities may express their opinions and it is not restricted to one vote process. If trust leadership is achieved by trust consensus, than trust leadership should consider options, historical contexts and aspects that lead to a better decision-making process regarding trust.

If the context approach is taken into consideration, the entity that has the highest trust value could be the leader. The problem is that when multiple contexts are used, the highest value for a particular context shall not represent the better leader, i.e. the entity that has more capacity in the system may serve only one context so it will not represent the better choice to others entities, when multiple context is needed.

There is a consideration where one prerequisite to be the leader is that the candidate entity has trust and reputation values in the group. Well, any entity may announce its preferred candidate also depending on the context. That makes the leadership choosing process more complicated than just voting. For example, an entity could be responsible for taking on the opinions of other entities and announcing in the network who has the most elevated trust value among $n + 1$ members, where n is half of the whole group. Well, in large groups that can be a problem because this process may flood the network with messages for just choosing one entity as leader and packets may also not reach its destination, etc.

If distributed system approach is considered in a group of entities, a leader may be a router, a middleware broker, a dispatcher, an authentication server, a cloud replication node, or any piece of component that messages can go through in a computational system.

Considering that such problems are not part of the proposed solution, and in order to simplify the development of the group trust model, which is one of the objectives of this research, developing a trust leadership algorithm or a trust leadership model will not be considered. This work will select any entity that is part of a group and give it the role of leadership.

4.2.1 Trust Consensus

Leadership in trust is consensus dependent in many aspects. In the trust leadership problem, any entity on the group may be the potential leader of the group. Another point of view is that entities may infer that a trust value is acceptable or not thus generating a trust threshold, which in first instance, can be accepted as consensus if entities are above the defined threshold value. But if an entity does not overcomes a specific trust value then another entity may assume by his own means that the entity in focus may not be the leader for it. Well, that maybe not the case for another entity that believes that he has enough trust to be the leader.

The assumptions discussed lead the problem to a quantitative trust measure and not a qualitative approach. In some computational system or even in distributed system that is not adequate. There are few works in literature that deal with trust consensus, but fewer works deal with it in a computational perspective.

It is not an easy task to define a group leader in a trust manner. The trust threshold is difficult to address in large systems to select a leader because every entity doesn't have the same trust value about any other entity once trust is calculated individually making use of its own inferences. Even though entities may agree on an ordinary value of trust and also agree that this value is enough to assume that one specific entity can represent the group. This assumption would transform the chosen entity in the leader of the group. However, in real distributed system scenarios, where every node can perform its own decision, such agreement is very difficult because entities must exchange trust and reputation information that should have been defined previously considering, for example, security aspects such as availability.

In this case, if we assume that an entity should not trust another entity if its trust value in not inside a specific range, this entity could avoid exchanging information with others in the system thus leading to a complete failure of acquiring enough trust information to create trust consensus. Besides, this should consider that some basic factors are commonly known and used by every entity in the system and that may not be true in most distributed systems.

Also, if only voting schemas are considered for a leadership choosing process, this may not represent consensus because what an entity does is just to vote. In other words, it is to choose one among many options. Voting schemas, in general, do not consider consensus in a distributed manner, what contradicts the trust aspects regarding consensus.

Considering the same perspective of trust leadership and in order to simplify the development of the group trust model, it is not the objective of this work to develop a trust consensus algorithm or a trust consensus model. This work will assume that the defined leadership role will be accepted by entities in the group.

4.3 Group Trust Model

In the approach of this research it is considered that a leader already exists in the group and entities in the group agreed that it is the representation of the group for new members and for the outside world as well. The review of trust indicates that any entity may have lots of relationships in many contexts.

As trust is calculated by considering each interaction, the final trust of one entity over another inside a context in a particular period of time can be calculated as the average result of all interactions already done for that context. In this case it is important that the leader of the group knows every context the group has in the system, i.e., service offering.

In order to calculate the group trust this work moved towards the use of reputation values that an entity has over another. This is because the leader may not be able to use just his own trust values, but he needs the opinion of every entity in the group about each other. This way the group trust will be the representation of all the opinions of every

entity stated by each other in the group. It means that every entity individually evaluates each other according to its own knowledge and shares this opinion with the leader of the group.

Supported by this analysis, the trust added value is a consequence of the individual trust values of the group members viewed as an opinion. The added value represents a point of information for external entities so they can use it to infer the whole group trust value.

Thus, after these assumptions, the following defines how to calculate the trust value of the groups in the approach of this work.

4.3.1 Initial Context Representation for Group Trust

In large groups, it is normal that entities may not be able to actuate in every context available in the system related to services. For example, one entity may be able to upload files but may not be able to perform matrix calculation. So, as stated by definition 8, it is a requirement of the model that the leader in the group knows every context in order to be able to calculate the trust information of the group for all contexts available within the group.

Considering that a group may have one or many contexts, firstly, the reputation that entity **B** has for entity **A** in a particular context **C** is represented in equation 4.1.

$$\delta^{C}_{a\to b} =_i V^{C}_{a\to b} \tag{4.1}$$

where ${}_iV$ is the reputation value calculated for the interaction i using the available reputation model in the system.

In this case ${}_iV^{C}_{a\to b}$ is one record that represents the expectation that **B** will fulfill **A**'s requests in context **C** for the interaction i. Trust and reputation values may be stored as many individual records of every contextualized interaction with the same entity or different entities.

4.3.2 Final Context Representation for Group Trust

Once it is possible to store and compute data for trust and reputation values for each interaction, the system has to be able to process the final context reputation value for the available entity.

Considering the context approach, an entity may have a collection of different reputation values for other entities in the system. Then, in the case of a particular entity **A** may calculate the final reputation about an entity **B** for a particular context **C** using equation 4.2.

$$\overline{\delta}^{C}_{a\to b} = \frac{\sum_{i=1}^{j} {}_iV^{C}_{a\to b}}{j} \text{ , for } j > 1 \tag{4.2}$$

where $\overline{\delta}^{C}_{a\to b}$ is the value that represents the final reputation entity **B** has in **A**'s perspective in context **C** and j represents the amount of interactions that **A** and **B** have done in context **C**.

4.3.3 Representation of all Contexts for Group Trust

Basically, in a group there are as many contexts as the computational system design wants. With this in mind, an entity may be programmed to offer one particular service, or download file, or route packets, etc. It may also achieve all of these things together. In this case, this particular entity has as many contexts as needed to accomplish its function in the system.

Then, considering that one entity may have as many contexts that it is programmed to, the final reputation regarding all contexts of one entity about another is given by the expression in equation 4.3.

$$\overline{K}_{a\to b} = \frac{\sum_{i=1}^{x} \overline{\delta}_{a\to b}^{C}}{x} \text{ , for } x > 1 \tag{4.3}$$

where $\overline{K}_{a\to b}$ is the final reputation value that entity **A** has about **B** for all contexts they collaborated before; x is the amount of all contexts that **A** knows about **B**.

In this way entity **A** is able to store all the reputation information about **B** in a given time period for all contexts that **A** knows about **B** as one value. Then, this value is used to help in the calculation of the trust value for the group. It is important to remember that reputation evolves with time, so this value can go up or down as times goes by considering that **A** interacts with **B**.

4.3.4 Initial Value for Group Trust

From the perspective of this research work, what best represents the trust value of a group is the reputation that every entity within the group has about all entities of the group that it is part of. Then, the trust value can be calculated as an average reputation of all members inside the group. Considering this, the leader of the group receives and organizes the reputation values of the rest of the entities and computes the trust value of the group.

For example, let us consider an example of a group G as a collection of 5 entities E represented as $G = \{E_1, E_2, E_3, E_4, E_5\}$ and E_5 is the agreed group leader. In this case, E_5 asks E_1 about the reputation information it has about E_2, E_3, E_4 and the leader himself E_5.

After that, E_5 asks E_2 about the reputation information of E_1, E_3, E_4 and again E_5. This goes on until E_5 has all reputation information of all group members. Entities uses the equations 4.1, 4.2, and 4.3 to be able to compute the final reputation of each other in every context.

Then, the leader E_5 uses this information to calculate the initial trust value of the group. It may sound like E_5 may manipulate the reputation value that it receives, which is true. To avoid that particular case, this proposal assumes that the leader role is an important function in the group so the leader must not cheat, otherwise the whole system may fail if it is trust based. It is important to consider that this is not a particular problem of the proposed model alone. If a group leader in any group cheats, the group itself is not trustworthy as it will be shown in section 4.6.

It is important to mention that there are few propositions for trust protocol to exchange trust and reputation information. In other words, there is no common trust communication protocol agreed in the literature for exchanging information in distributed systems. It is then assumed that the protocol to exchange information with the leader can be proactive, reactive and hybrid depending on the scenario in which the protocol is deployed. Also entities are able to exchange trust and reputation information securely, otherwise there is no point in distributing trust information and relying on them for soft security.

Note that in some scenarios the members of groups already know who the leader is and then can proactively send such information. Thus, a new member can always ask for the leader of the group, so we assume that the entity is able to find and communicate with the leader.

Once the process of trust and reputation exchange is defined, the leader computes the final reputation of each entity as the average of the reputation values provided by the rest of entities within the group. This is performed using equation 4.4.

$$\overline{\omega}_g^n = \frac{\sum_{i=1}^{j} \overline{K}_{n \to m}}{j} \text{ , for } j > 1 \tag{4.4}$$

where $\overline{\omega}_g^n$ is the average reputation of entity n as seen by the rest of entities of group g; j represents the quantity of members in group g; and $\overline{K}_{n \to m}$ is the final reputation value that entity n has about m for all contexts, considering the perspective of equation 4.3.

4.3.5 Final Value for Group Trust

After performing the final average reputation of every entity in the group, the leader can generate the final trust value of the group using all reputation values collected and computed before. This is done using equation 4.5.

$$\overline{\lambda}_g = \frac{\sum_{i=1}^{x} {}_i\overline{\omega}_g^n}{x} \text{ , for } x > 1 \tag{4.5}$$

where $\overline{\lambda}_g$ represents the final trust value of the group g and x represents the quantity of members in the group.

Once the leader has the trust value of the group, the leader can share this information to all members within the group or when asked by an outsider. In the case where there are many groups, every group leader can perform its own trust value calculation and inform its trust value to other group leaders.

Every group leader has the responsibility to store group trust information, send it when asked, and distribute this value to new members, new leaders, and outside group requests as well. As seen, the group role is very important, so the leader must be chosen carefully in order to enhance soft security.

In the following sections, there is the implementation of the proposed model and the results acquired by the experiments developed.

4.4 Proposed Algorithm

In order to create a common process to perform group trust calculation the algorithm represented in Figure 4.2 can be used. It is divided into 3 steps and each step is explained below.

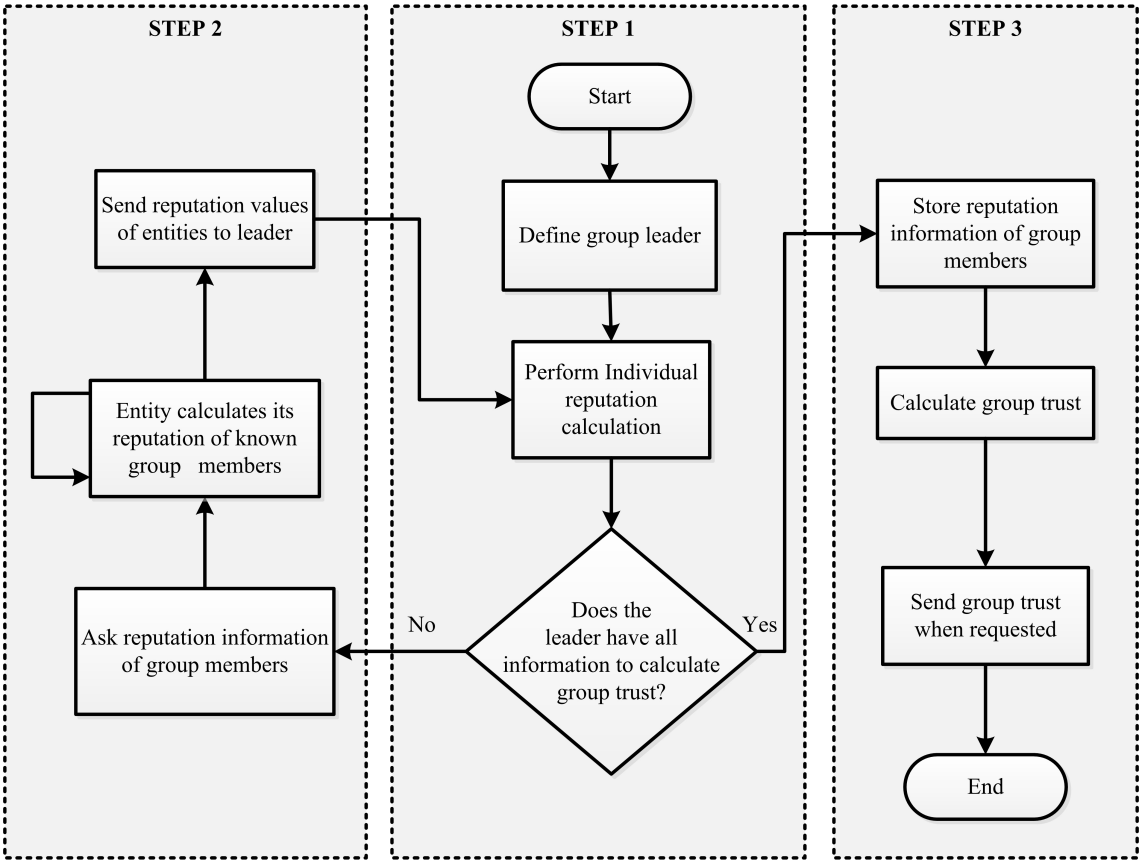


Figure 4.2: Proposed algorithm for group trust calculation

4.4.1 Step 1

The whole group trust process starts with the definition of the group leader. As explained in section 4.1, it is expected that the group chooses its leader according to its interests. Thus, this leader calculates the individual reputation of the members of the group he might already have information on.

After that, the leader checks if he has all the information on the group members that is necessary to accomplish the group trust calculation. Then the leader gets in a decision tree where if he does not have all the information, he goes to step 2. If he has the necessary information, he goes to step 3.

4.4.2 Step 2

Well, if the leader's information is incomplete, he starts a process of discovering and requesting reputation information from the group members. Following the equations 4.1,

4.2, and 4.3, defined previously in section 4.3, entities that are members of the group start their own calculation of the others entities' reputation.

After generating reputation values as seen by each particular entity of the group, each group member then sends this information to the leader using the communication protocol defined to do this task.

The leader than goes back to step 1, performs the individual reputation calculation, and gets to the decision tree until he has all the information required, so he goes to step 3.

4.4.3 Step 3

The leader stores the reputation information of group members in a repository and using this information, the leader applies the initial group trust using equation 4.4 and then the final group trust using equation 4.5.

Once this calculations are finished the leader stores the group trust information and sends it when requested to members within the group or to outsiders, depending on the case and the algorithm of group trust calculation ends.

4.5 Testbed Environment and Characteristics

As means to verify the viability of the proposed model, a set of implementation tests was developed constituted by a basic infrastructure for network communication and simulation where the group trust model could be checked. In essence, a testbed has been set up by means of a P2P simulation tool to create an environment and to develop all calculation processes. The use of P2P is due to the necessity of representing a CS with distributed characteristics. The following subsections explains the test bed environment and its particularities assumed for this thesis.

4.5.1 Initial Assumptions

The simulation tool uses asynchronous interactions between machines and 5 different scenarios where simulated according to specific policies in the network, resumed in Table 4.1.

Table 4.1: Scenarios developed

Scenario	Description
1	All peers behave accordingly
2	All peers behave randomly
3	Random behavior of 20% peers in group 1
4	Random behavior of 40% peers in group 1
5	Random behavior of 60% peers in group 1

After the definition of the basic characteristics of the scenarios, some assumptions were defined in the test environment in order to organized the tests. For simplicity, the testbed

is accomplished assuming that peers do not lie about trust and reputation values in the network, however this behavior can also be detected using the underline trust model used in the proposal.

All participants are doing the same number of interactions in the testbed. The intention is to find standards and verify certain behaviors about reputation values in the system. When interactions between entities correspond (or not) to the expected behavior, then it can be determined if the peer is trustworthy or if it is not.

Some behavior patterns have been delimited as desirable in the system, as follows. Firstly, there are no errors in the communication transmission; secondly, the time for transmitting a file is determined by the quality level of the transmission. These parameters have been chosen in order to simplify the P2P environment, thus permitting the focus of the analysis in trust and reputation values considered in the interactions of peers and to perform the calculation of the group trust value.

4.5.2 JXTA Shell Basics

The JXTA Project [JXT13a] is constituted of open protocols responsible for the execution of all necessary functions in a P2P network, as nodes research, resource discovery, query publishing, etc, all defined in the JXTA framework. Peers were set up in machines used in the test environment, using standard commands of JXTA Shell [Jxt13b]. In this shell the commands for trust, reputation and group trust were developed.

Basically, JXTA functions are executed through publishing and exchanging notifications (advertisements) and using Extensible Markup Language (XML) messages among the peers. It is based in a hybrid P2P architecture that uses a Distributed Hash Table (DHT) to store all information related to the peers.

The JXTA [Jxt13b] platform keeps an ensemble of Java open codes and enables specific codes to be easily added. JXTA Shell [Jxt13b] is an application build based in JXTA [JXT13a] that works through commands that simulates an environment similar to a UNIX Shell. This environment allows java classes to be created and added to the basic structure of the code, facilitating the platform P2P to be used as a base to distinct studies in distributed environments; therefore providing a P2P platform so trust and reputation could be implemented in accordance with the objectives of the proposed model.

In short, during the development process, new commands were created in the JXTA shell in order to develop a P2P simulated environment system to establish connections based in trust and reputation between the nodes. Such commands allowed the analysis of the group trust view.

4.5.3 Network Infrastructure

The network topology created for the simulation is simple and uses common layer 2 switches. The purpose of this topology is to represent a P2P network connected directly to a LAN. The peers are configured in the same network segment with no additional hops as seen in Figure 4.3(a) and each peer uses a different TCP port. The physical topology

uses a normal switch with 5 machines directly connected and each machine simulates a group id as represented in Figure 4.3(b).

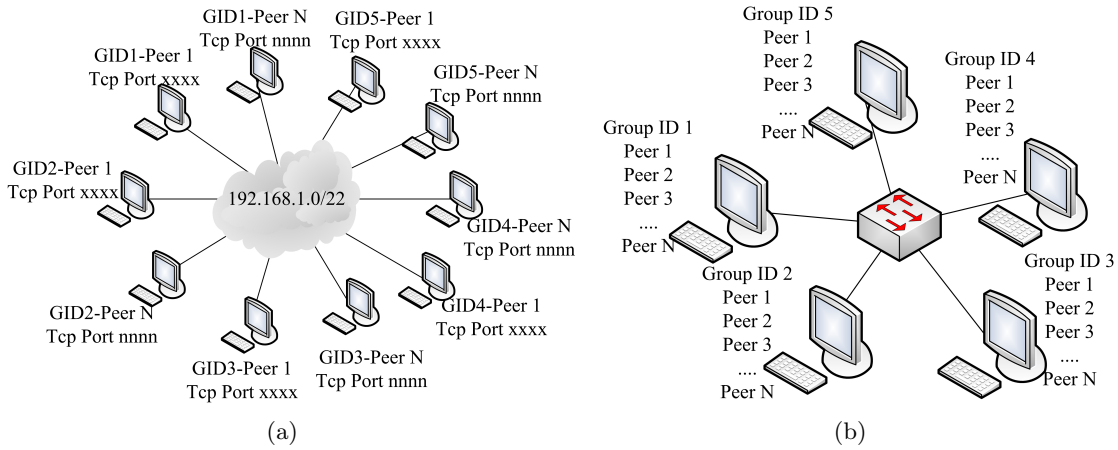


Figure 4.3: Topology of the testbed environment

4.5.4 Group Formation

The same consideration settings applied during the tests were used regarding the group formation and group amount of nodes. The testbed is executed in machines with JXTA Shell [Jxt13b] installed and configured. The simulated environment was composed of 500 nodes, where they constituted 5 different groups with a hundred nodes in each group. Each node only performs interactions within its group. Nodes use one particular context in our simulation.

Also, nodes are able to perform trust and reputation calculation. For this task, the model proposed by Patel [Pat07], TRAVOS, is used for the initial and individual trust and reputation generation for each node. In every simulation we performed 20 rounds of nodes interaction, so nodes could create a reputation image of each other.

It is considered that if a threshold of ($\overline{\lambda}_g \geq 0.7$) is reached then the group is said to be trustworthy. Each group has a leader that is trustworthy, which means that the leader does not behave unexpectedly, it is honest and it is able to perform individual trust calculation as defined by the group trust model proposed.

Information regarding the environment itself, such as network, bandwidth, capacity, delay, etc. is not considered important for the proposed simulation environment.

4.5.5 File Sharing Considerations

Once the simulations uses a P2P as the computational system, a service of file sharing is the context applied to this simulation. It is considered that the transmission delay and the integrity are the parameters used of a file shared in the network to decide whether a peer is malicious or not. This means that a peer can send a corrupted file, delay its sending to another peer or perform both.

The interval of time values for the transmission is defined after some file transfer tests were executed used for this estimative. Several interactions have been fulfilled for a file with fixed size of 100 Kbytes, and a standard time could be defined for a successful interaction.Table 4.2 has some summarized definitions of the test interactions to limit the values expected.

Table 4.2: Resumed interactions used to adjust the test parameters

Source Peer	Destination Peer	Time (s)	Speed (Kb/s)
GID1-Peer 1	GID1-Peer 2	0.121	3363
GID1-Peer 3	GID1-Peer 4	0.280	723
GID1-Peer 1	GID1-Peer 6	0.441	459
GID1-Peer 2	GID1-Peer 7	0.510	797
GID1-Peer 6	GID1-Peer 3	0.480	1221
GID1-Peer 9	GID1-Peer 4	0.701	1161
GID1-Peer 10	GID1-Peer 5	0.881	923
GID1-Peer 6	GID1-Peer 7	0.160	635
GID1-Peer 7	GID1-Peer 8	0.210	1937

Based on this information it has been determined that:

1. The expected transference time of a file is up to 1s;
2. A little delay would be between 1s and 2s;
3. It is completely delayed above 2s;

Another parameter defined is the integrity of the received file and for this a hash integrity check is used to verify this condition. Once this parameter is defined, the file load times are parametrically determined by the variable **Var A**, the file integrity check times are determined by the variable **Var B** and the variable **Var C** determines the reputation feedback for the trust model associated with the given interaction. Table 4.3 shows the parameters used in this testbed to define how to infer some reputation for a peer.

Table 4.3: Possible situations considered as parameters

Description	Var A	Var B	Var C
1. File corrupted and on time	0.00	1.00	0.250
2. File corrupted and a little delayed	0.00	0.50	0.125
3. File corrupted and completely delayed	0.00	0.00	0.000
4. File not corrupted and on time	1.00	1.00	1.000
5. File not corrupted and a little delayed	1.00	0.50	0.875
6. File not corrupted and completely delayed	1.00	0.00	0.750

The reputation value **Var C** is calculated by the equation 4.6.

$$VarC = ((P * VarA) + ((1P) * VarB)) \tag{4.6}$$

where the parameter P represents the importance that the network administrator allocates to the integrity of the file. Related to peers behavior, peers only accomplished interactions with appropriate parameters to verify the convergence of trust and reputation values.

After the definition of the explained parameters and configuring the network properly, the following sections show the results and analysis for the scenarios, previously defined in Table 4.1.

4.6 Results and Analysis

The underlying trust model used in the testbed is TRAVOS [Pat07], which allows peers to realize that some members of the network changed their behavior. Direct trust of the peers and reputation values based on context of the groups are calculated in order to generate group trust.

For the basic understanding, in all the result graphics the **X axis** is the *number of interactions* and the **Y axis** is the correspondent *group trust value* in each round.

4.6.1 Group Trust for Scenario 1

In this test all the peers in all groups behave as expected. This means that they fulfill their requirements and performs their defined context correctly. Note that the entire peer acts under the same behavior, without changing any aspect of its functional context. In this case, the trust value of the group is considered extremely trustworthy and it tends to stabilize in a value near 1, thus avoiding blind trust. When there is no malicious peer in the network, the trust value of the group reflects the individual behavior of the peers in the group. This is considered the ideal world and Figure 4.4 shows this result.

4.6.2 Group Trust for Scenario 2

In this test all peers in all groups behave randomly after round 4. This means that it is not known for certain by the other group members whether a particular peer behaved accordingly or not. This test was set up in order to verify the results when nodes sometimes behave in a proper manner and then change its behavior with no particular reason. This can be considered the worst environment imaginable because it cannot be possible to predict if a node will or will not behave accordingly. This scenario is represented in Figure 4.5.

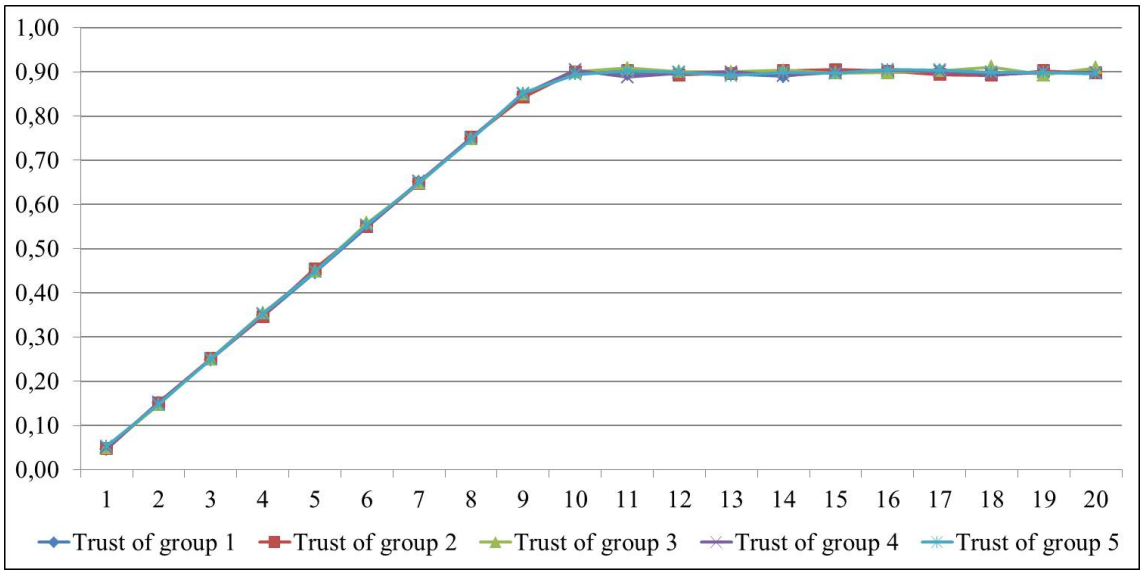


Figure 4.4: Group Trust for Scenario 1

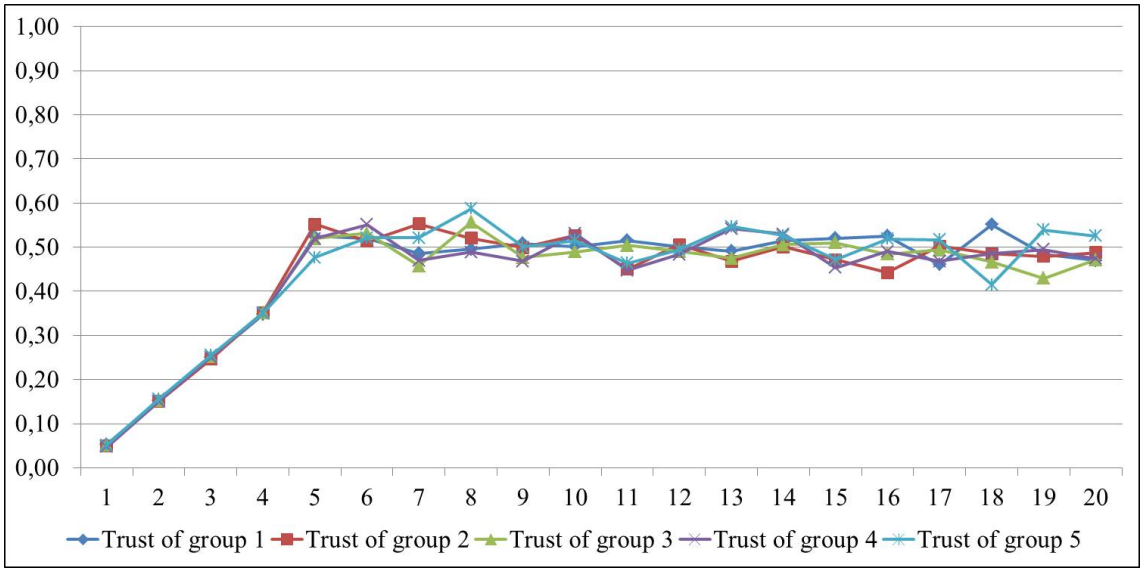


Figure 4.5: Group Trust for Scenario 2

4.6.3 Group Trust for Scenario 3

In this test 20% of group 1 behaves randomly. This test simulates a coalition of peers in order to modify the group trust. Such behavior is considered as if the nodes suffer some kind of attack or there are peers acting as black holes in the P2P environment. When 20% of the peers start behaving in a malicious manner, the trust value of the group decreases and tends to stabilize in a value near 0.8. The analysis shows that the increase of the trust coefficients provided by the good peers overcomes the decrease of the coefficient of the malicious peers. In this case the group is still considered trustworthy ($\overline{\lambda}_g > 0.7$) despite having malicious members, as represented in Figure 4.6.

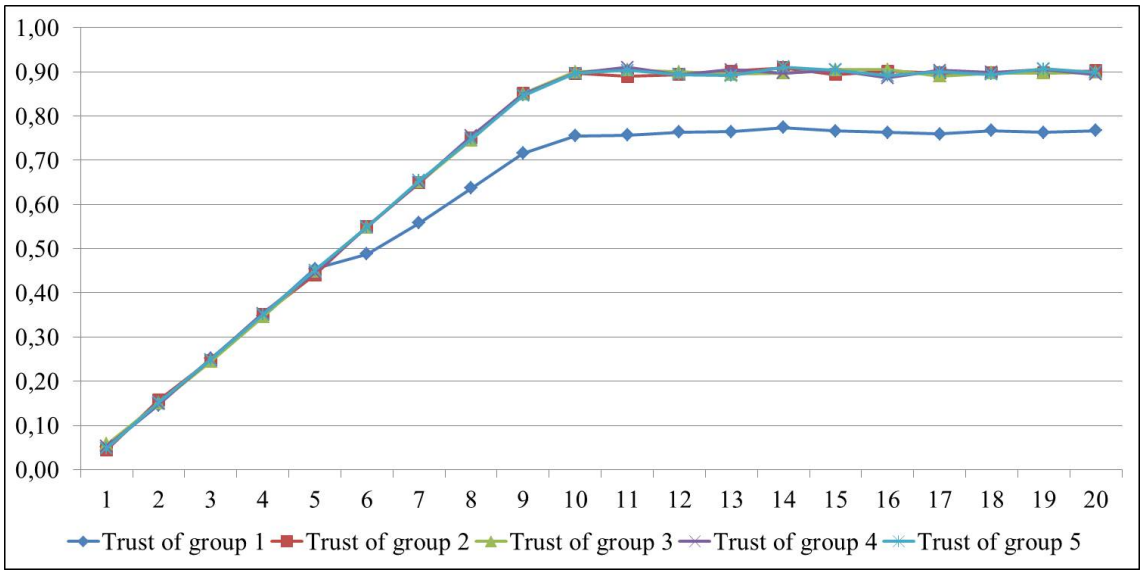


Figure 4.6: Group Trust for Scenario 3

4.6.4 Group Trust for Scenario 4

In this test 40% of group 1 behaves randomly. This test simulates a situation where the P2P network is compromised and there is no guarantee that the peers in this group are trustworthy or not. When 40% of the group members are malicious, the trust coefficient of the group tends to stabilize in a value near 0.6 what represents that the peers in the group are not trustworthy, thus the group is not considered trustworthy because the defined threshold. Figure 4.7 shows this result.

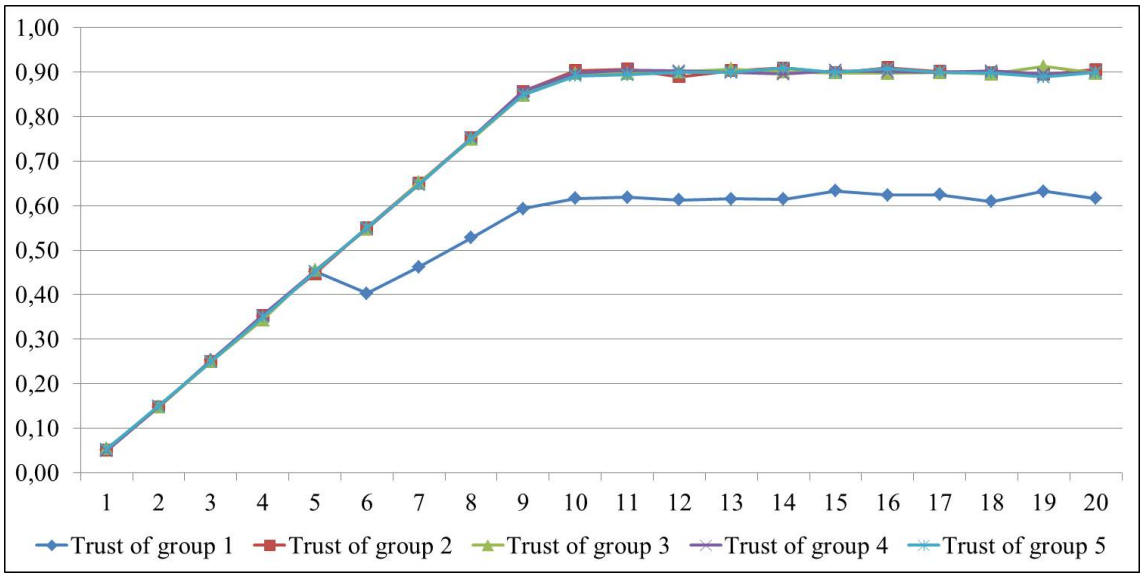


Figure 4.7: Group Trust for Scenario 4

4.6.5 Group Trust for Scenario 5

In this test 60% of group 1 behaves randomly, then the trust coefficient tends to stabilize in a value near 0.5, also making the group untrustworthy because of the defined threshold, as seen in Figure 4.8.

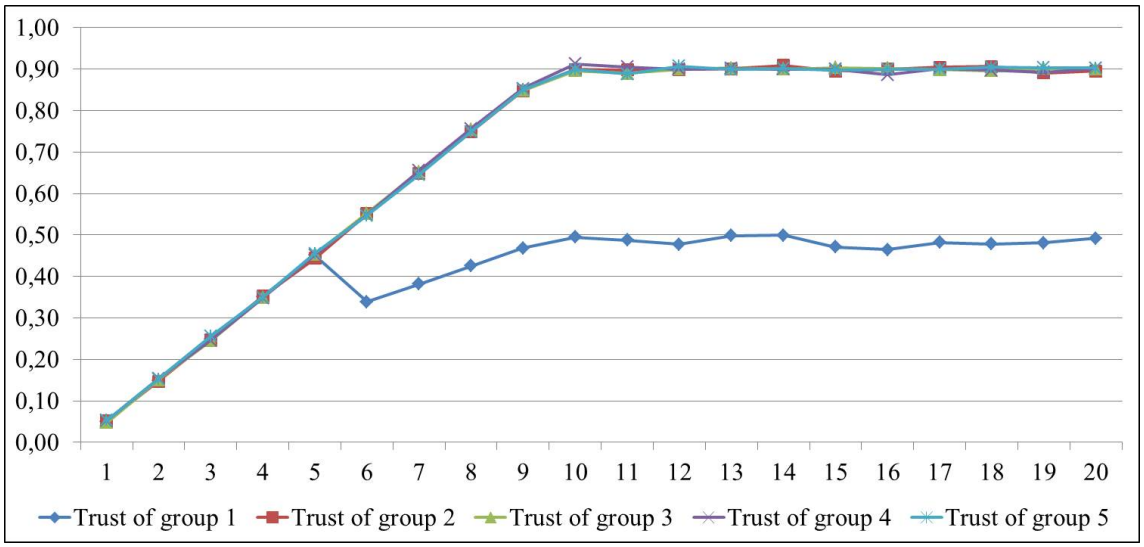


Figure 4.8: Group Trust for Scenario 5

4.6.6 Analysis of Group 1

This test was performed to analyze group 1, compiled in one graph, as seen in Figure 4.9. When peers change their behavior, the trust value of the group decreases, thus making a group with constant changes in its behavior to be untrustworthy considering a defined threshold of $\overline{\lambda}_g > 0.7$.

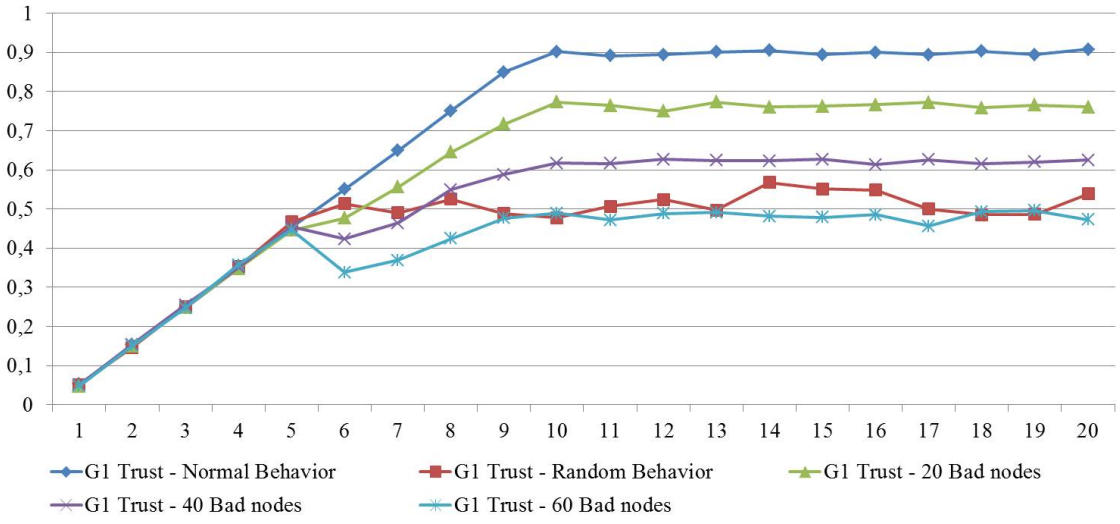


Figure 4.9: Group 1 synthesis

When nodes behave randomly the value of the group trust tends to be 0.5. The reader

may realize that the individual behavior of each member in the group influences the trust value of the group as a whole. The results can also be considered satisfactory because all the peers initiate at the same time in the network and interact with each other the same number of times.

These results also show that the trust value of group 1 in the first scenario is originally high (moment in which all the peers have good behavior). After that, it starts to decrease in the moment the peers in the group change their behavior or acts forming a coalition. As a result, the group trust model can be used as a parameter to interact or not with a specific group, thus it is understandable that it can used as parameter of enhancing soft security.

4.6.7 Extending the Number of Nodes of the Simulation

After it was possible to measure trust regarding groups in computational system and in distributed environments, it was decided to extend the measure capabilities with more nodes in a simulation. These new simulations considered the same aspects of the previous one, but instead of a 100 nodes in each group, it was doubled. Thus the total amount of nodes simulations became 1000 nodes divided into 5 groups of 200 nodes each.

Then three different scenarios were simulated. In the first scenario it was defined as the ideal environment, where all nodes behave accordingly and perform their job as expected so it fulfills other nodes outlooks. This scenario represents an environment where there are no malicious nodes inside the network and all groups are considered trustworthy. Figure 4.10 illustrates this perspective with 200 nodes in the P2P system.

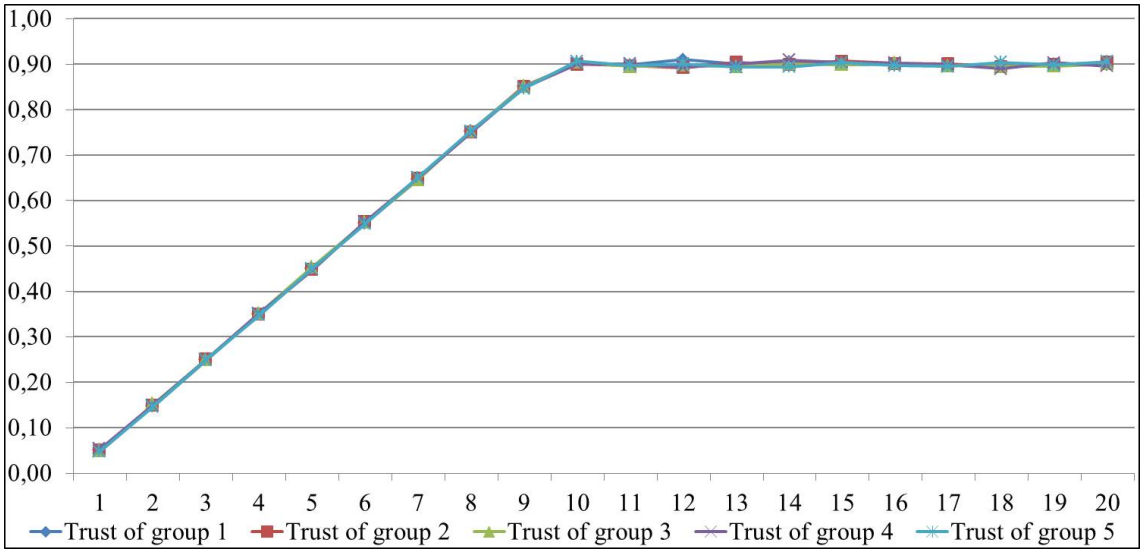


Figure 4.10: Group Trust with 200 nodes

In the second scenario, we have all nodes moving in a random way. This scenario, illustrated in Figure 4.11, represents the worst environment possible where it is not possible to say if the node is bad or good because it changes its behavior unexpectedly. In such cases, one particular node may be trustworthy to one entity because it may fulfill ones expectation in a particular condition, but next it may completely change its state.

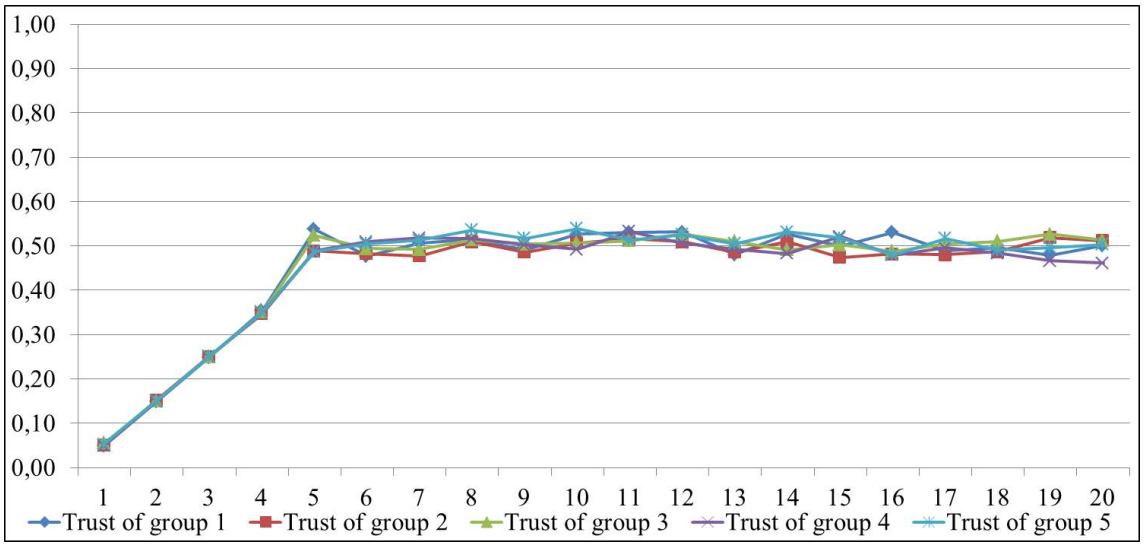


Figure 4.11: Group Trust with 200 nodes with random behavior

The last scenario, illustrated in Figure 4.12, summarizes the view of group 1 in 4 different approaches. It compares group 1 trust with all nodes behaving accordingly and all nodes behaving randomly. Than it shows a particular case where 40 nodes (20% of the members of group 1) changes their behavior after round 5. After it shows 80 nodes (40%) of the members with change in their behavior. In both cases, with 40 and 80 bad nodes, it is possible to see that when a coalition of nodes are made, the group trust model is able to address the change in trust of the group.

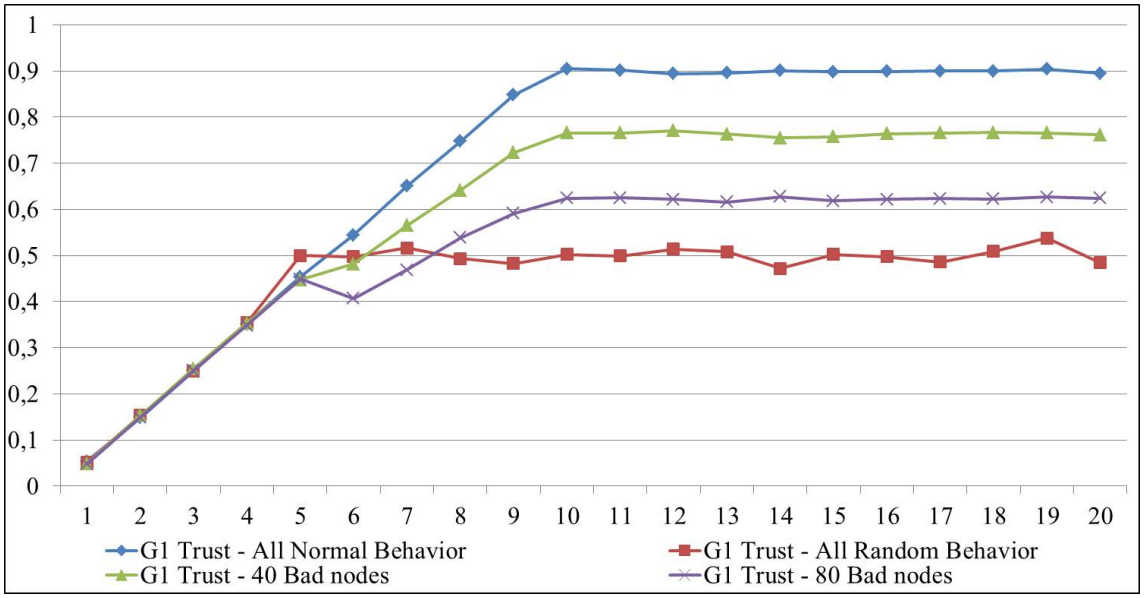


Figure 4.12: Group Trust resume for group 1 with 200 nodes

4.7 Synthesis of This Chapter

This chapter presented the proposed group trust model developed in this research. The model can be seen as an extension to support the calculation of trust values of groups of entities in distributed systems.

A simulation was defined with a set of requirements and parameters in the testbed environment. After that, the proposed model has been validated in a P2P simulation tool. The results showed that it is possible to generate and to calculate group trust behavior in a computational system. As a means of creating distributed environment for the simulations a P2P network was used.

Using the concept of group trust, the proposed model in this chapter can be used in bigger and more complex distributed systems architectures as well.

Basically, this chapter presented the initial discussions to the understanding of the group trust problem. After that, some considerations were presented and problems about group leadership and trust consensus. Following this, the proposed group trust model was presented and the related calculations mechanisms. Then an algorithm was designed in order to verify the viability of the model. As a last step of the schema, a testbed environment was proposed and the results acquired by the experiments developed were shown.

The next chapter discusses deeper relations of trust and information security.

Chapter 5

Trust and Information Security

What normally is seen when dealing with security is that information can be considered the most important asset of any modern organization. It is clear that securing this information involves preserving confidentiality, integrity and availability, the well-known CIA triad.

International normative such as ISO/IEC 27000 family of standards, BS7799-x and reports provided by NIST, characterize information security as a risk management job where the main task is to manage the inherent risks regarding information. Normally these risks deal with disclosure, destruction, misuse, etc. of information. Such normative and guidance are put together to form the perspective of ISMS.

The following sections are organized as follows. In section 5.1 the initial perspective considered to the developed architecture is presented. Section 5.2 describes the proposed layered architecture and section 5.3 discusses information representation and its relationship to information security. Finally, section 5.4 compares the layered architecture with other known information security models and section 5.5 summarises this chapter.

5.1 Initial Perspective

Managing information security is critical. Even so, very few formal models are able to manage the tasks of keeping the information secure. After what happened with the information disclosed by Mr. Snowden ([GMP13], [Ric13]) what already seemed very difficult in terms of security, controls, policies and so on, became much more problematic. Research on information security already acknowledged many possibilities about the options of the disclosure and technologies that could be used, but its level was beyond expectations.

Basically, the fact is that it is no longer possible to deal with information security naively or by obscurity.

The power behind security agencies and information security systems has proven to the world to be something that only security experts were able to understand before. There is a clear shift in the world of security that takes place after Snowden's information disclosure, i.e, new encryption tools, the increase of using https protocol in more websites, new Virtual Private Network (VPN) providers, etc. Of course, it is important to mention

that information security compliance has been proven not effective.

Indeed, the discussion about information security seems endless. Consequently, the information security research field faces new challenges in terms of confidentiality, privacy, anonymity, plausible deniability and so on. If the perspective of information security as an economic strength is observed, it is a billion dollar industry [Gar13], only with respect to cloud solutions.

Nevertheless, the problem is that no matter how much money you pay for security, it may never be enough to avoid data theft, information disclosure, system exploitation and other risks that are part of the information technology security context. News about hacks are evidence of these aspects.

There is also a common understanding that, usually, information security is not properly understood by organizations, because most projects start with no security approach to the problem itself nor information security related to the whole project.

One way to deal with the information security problem is to manage those risks from different perspectives. Risks that can compromise enterprises or governments are associated with natural phenomena, technological risks and human-related risks [BMG02], but it is not enough to only consider these characteristics. It is necessary to further improve communication systems and the way they are secured.

5.2 The Proposed Trust Information Security Architecture

The way this research assesses security is based on a layered architecture with components connected in a way everything is part of a puzzle that must be well-connected and understood, so information security can be seen as a whole. Figure 5.1 illustrates the proposed trust information security architecture (TISA) and its layers.

In the following topics each layer is described in a top-down description and its corresponding subtopics, in addition to a brief description of its importance to the puzzle.

5.2.1 Layer 1

It covers the basics to start thinking about information security inside an organization. If one does not understand what data, information, information assets, etc. are to an organizational environment, there is no point in discussing information security, simply because one does not know what should be protected.

It is important to point out that the "protect everything" approach is not effective and it is very expensive. After that, there is the CIA triad and what this work decided to call "information security extensions". The following items summarize the proposed approach.

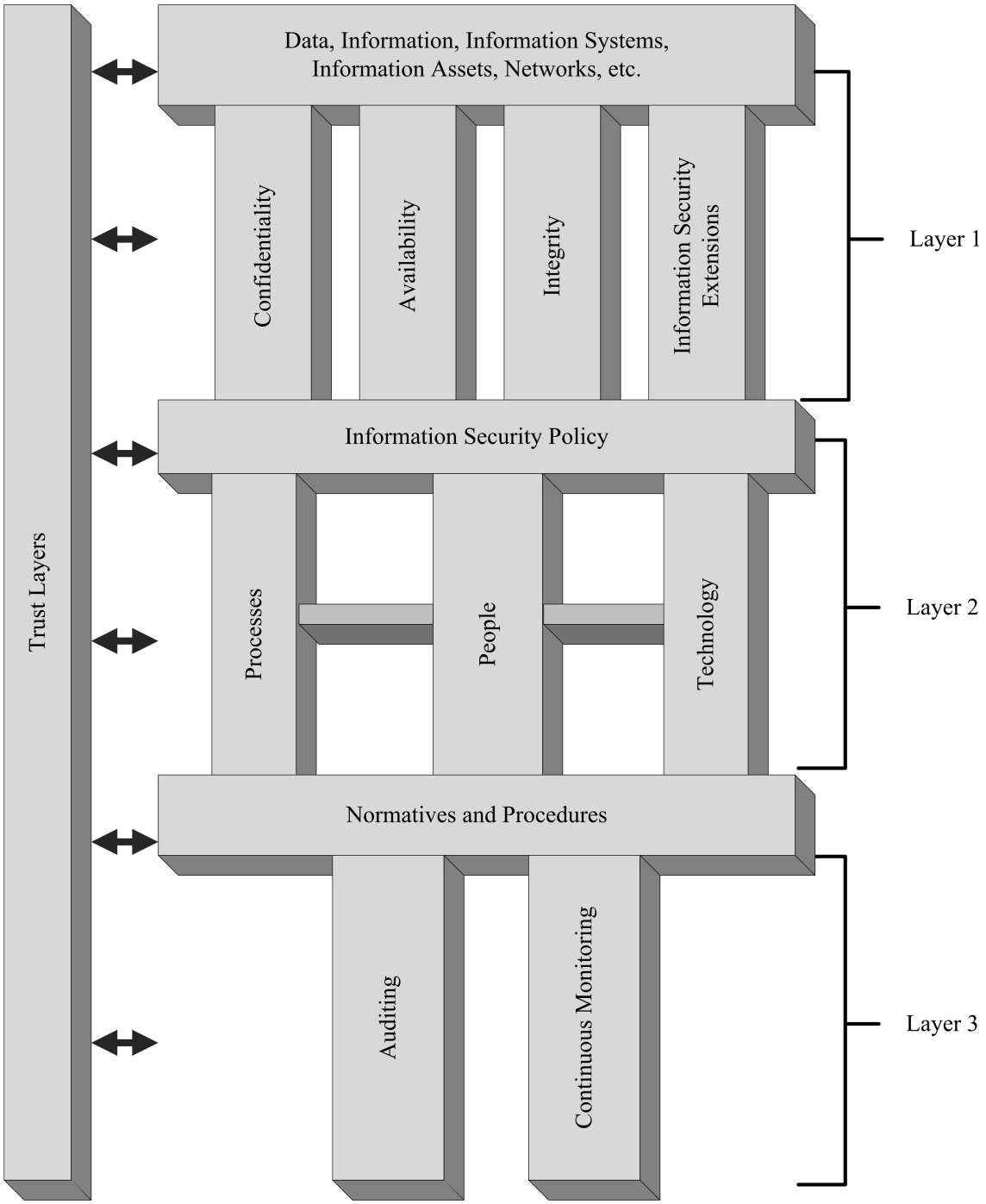


Figure 5.1: Trust Information Security Architecture

5.2.1.1 Data, Information, Information Systems, Information Assets, Networks, etc.

Generally, this is the layer in which data is important to organizations or in which individuals are mapped. Considering the importance of data for organizations nowadays, information can be retrieved from raw data and transformed throughout the use of information systems. Using the correct tools and techniques, it is possible to create new

knowledge from data, which at first, made no sense at all.

In the vast data world, crawling, retrieving, analyzing, discovering relations, finding hidden patters are the areas in which efforts have been made in order to create knowledge from raw data.

When it comes to information assets, it is also very important that they are identified and labeled, and its relationship to information should be clearly understood by the organization and the ones responsible for its protection. Methodology such as Octave [AD02] are good starting points.

Networks connect every piece of data, information and information assets so anyone with granted access is able to browse them. Considering information or data in transit, it is important to remember that, in networks, bits are not treacherous by themselves: people who control it or how they do so (process, hardware or software) are who/what makes them dangerous or not. Therefore, marking network perimeters, having network policies, providing defense mechanisms etc., are still ways of managing what comes in and out of our network. This helps the strength of the security measures.

The proper use of such mechanisms is key for understanding what happens in the web and information systems. These components of the puzzle create inputs for supporting decisions related to information security.

5.2.1.2 Confidentiality

Confidentiality is the information security property responsible for preventing unauthorized disclosure of information. In other words, it is a mechanism to give access for authorized individuals or systems in the organization only. It applies principles such as the "need-to-know" and, in order to be effective, confidentiality must ensure that access to vital information is limited only to those who specifically need to have access to or use that particular information. This piece of the puzzle deals with the organization's ability to keep its data, information and knowledge protected from unauthorized disclosure.

5.2.1.3 Availability

Every piece of information has a specific value or use depending on a specific end. In order for information systems to serve their purpose, information must be available when necessary. To acquire such a property, all information systems, networks, databases, information assets, storage mechanisms and so on must be accessible by the ones who have granted access to manage them when required. Information is considered unavailable not only when it is lost or destroyed, but also when its access is denied or delayed for those who are authorized to use it.

Therefore, to guarantee availability, communication channels used to access information, wherever it is, and must be operating correctly and accordingly to information security policies. This piece of the puzzle deals with the required organizational skill to guarantee that information can be accessed by the ones who have the necessary permission regarding its use when necessary.

5.2.1.4 Integrity

Integrity is the ability to guarantee accuracy and consistency of data and information during its entire life cycle. Considering this quality, data or information should not be destroyed or modified without authorisation, which would hinder its detection. Integrity has to guarantee that the information is accurate, reliable and, most importantly, that it has not been changed or unexpectedly tampered with by an unauthorized entity.

Integrity may include the ability to verify whether information content has been unauthorized altered, or to determine the origin of any action in the system and to associate it with a specific entity. This piece of the puzzle deals with the way information is securely managed in organizations with no loss of its basic properties.

5.2.1.5 Information Security Extensions

There is discussion among the information security experts on whether the information security triad is enough to stand as the basis of information security by itself; this research believes it is not. However, it is also very difficult to extend all types of information security properties or attributes, thus, ensuring that your information is secure and for you to have access to the so called extensions.

This manuscript pictures information security extensions as a group of new attributes or properties to protect information and systems, but they are not limited to it. Table 5.1 summarizes some of them from the perspective of this work.

Accordingly, these properties are able to support themselves as elements that depend on many other complex characteristics, such as context, technologies that support them, usage objectives, and so on. In short, information security extensions are a bit of the puzzle that completes the information security basis, and, in addition to that, some of the terms are sufficient by themselves.

Table 5.1: Summary of information security extensions

Information security extension	Description
Authentication	It is the measure that verifies identities. In a communication process, the party must provide evidence that it is the person or entity to whom the credentials belong. At least one identification element is included in the authentication process. Usually, this process requires personal/individual knowledge, ownership information or personal information, despite the way these are linked. They are related to something you know, something you have or possess, and some information about yourself.
Access control	It is what restricts access to a certain data, information or resource. According to this principle, once access control is guaranteed, the entity should be able to extract, enter or use a particular piece of information or an information system. Generally, access control mechanisms begin with the identification and authentication processes.
Non-repudiation	In short, a party within a communication process cannot deny having received specific information nor can the other party deny having provided specific information. Generally, cryptographic systems can support non-repudiation efforts using digital signatures, but the discussion goes beyond the technological field because one cannot guarantee that such signature proves authenticity and integrity, thus preventing repudiation; for instance: cases of data theft. The opposite is so called plausible deniability, in which one's culpability might be denied or mitigated, or it is even possible to deny that one was responsible for it in the first place.
Authenticity	It is a way to ensure that data, systems, communication processes or information are genuine. Authenticity is also responsible for validating that both parties involved in a communication process are who they claim to be.
Privacy	It regards the right to control information about an individual and the right to limit access to that information. It is also related to domains in which individuals have the right to keep confidential information and data, and to share them in private conversation [DL07]. It is also the right to protect your personal information and to prevent invasions of privacy. Privacy is preserved for one's good [HS04].
Anonymity	It is simply a result of not having identifying characteristics disclosed or made available to the public which would allow the identification of an entity. In a communication system, anonymity is the party's state of being anonymous and yet being able to respond to or interact with another party (without revealing its identity). It is also related to terms such as untraceability and unlinkability [Inf09].
Authorization	It is the process of providing access to particular information or system to a party based on their identity. After going through the authorization process, one is allowed to have access to some or all data in a specific environment or system. Authorization allows an entity to access and perform determined actions regarding data. In order to be effective, authorization should be based on the roles that an entity may have.
Resilience	It is the ability of an information system to keep its minimal service levels guaranteed, even under challenges to its normal operation, attacks or failure of some of its components. It is also a perspective of dependability when facing changes in its operations.

5.2.2 Layer 2

According to the top-down description, this is the layer where the architecture will help to understand how, why and which technology may be used and who may use it in order to provide information security for higher levels. The following items summarize them.

5.2.2.1 Information Security Policy

The information security policy is a high-level document that outlines specific requirements or rules that must be met regarding information security. Generally,

this policy is very specific and covers only one organization. The information security policy also links security management to security issues and security controls. Once the information security policy is simply attached to the organization, it is important that the policy is formulated taking the organization's features into consideration.

The information security policy should explain the fact that all users need information security within the organization and should complement business objectives. Thus, the policy should be aligned with the organization's strategic information plan [Inf09].

The information security policy is the piece of the puzzle that guides people during the creation of the processes and definition of technological components in order to help protect information. Daily practices show that without information security policies, things are done based on individual efforts that, usually, are non-effective.

5.2.2.2 Processes

Inside information security, processes are formal mechanisms to identify, measure, manage and control risks related to information or its value to the organization. Processes are very important when information security is in question, thus it should not be seen as a black box or something that is meaningless to the organization. To be more specific, processes include formal and informal mechanisms (large and small, simple and complex) and provide a fundamental link to all dynamic interconnections.

Processes derive from strategies and implement the operational side of what should be done within the organization. To be useful and provide advantages to the enterprise, processes must be directed linked to business requirements and be aligned with information security policy in the puzzle. They should also consider emergence situations and be adaptable to changes in requirements. It is important that information security processes are well documented and communicated to human resources that should know about them. It is fundamental that processes should be reviewed periodically to ensure efficiency and effectiveness [Inf09].

5.2.2.3 People

"People" is the number one subtopic of the puzzle and represents human resources. In general, people develop and implement each part of an information security policy, create and maintain processes, information assets, define which technology should be used, design networks, etc. Security issues affect people and their relationships, values and behavior.

When working with information security it is important to address points such as strategies related to hiring, access, responsibilities, training, dismissal, damages, and whatever is considered important to be addressed to help to maintain the organization's information security strategy about the human resources.

When dealing with people, it is very important to understand that what sounds obvious to security experts, definitely does not sound obvious to someone without the same experience. People are the ones whose actions within the organization influence the information security triad and security extensions related to data, information, information systems, information assets, network usage, resources and whatever is valuable to the

organization. Including the information itself.

In short, individuals' actions and motivations have a directly positive or negative influence on information security. All this is related to behavioral information security [CJL+13] [AP12].

5.2.2.4 Technology

This is the piece of the puzzle that is the set of all informatics systems, applications, tools, infrastructures and defense mechanisms that the organization applies to achieve its goals and to help information security. Technological elements may frequently change and update, become obsolete very fast, or be the core of an organization's infrastructure.

Technology may also solve security threats and risks. Users and the organization's environment also have a strong impact, once technology can be perceived as a way to avoid security controls bypass [Inf09].

It is very important to remember that technology, by itself, does nothing. It should be seen as part of a complex system that has specific needs to protect what is valuable within the organization, and should work along with people and processes in a cyclic pattern, all those guided by information security policies.

5.2.3 Layer 3

It represents the piece of the puzzle that deals with daily activities and how they should be done. In addition to that, it is related to what actions should be taken in case a specific problem arises. Bear in mind that complex systems and their security practices are guides to keep information secure; but norms, procedures, monitoring and auditing will give systems administrators the tools to help them keep information, information assets, networks, systems, etc. more protected.

5.2.3.1 Normative and Procedures

Basically, normative is related to how things should be and how they should be rated. It is also related to how to classify things as good or bad or which actions are right or wrong. Normative are fundamental for prioritizing goals, organizing and planning actions in order to define how things should be done. Considering the user's perspective, it is much more common than it may seem, but users generally try to follow social expectations rather than following security procedures [PCK11]. This is not desirable regarding security perspectives because such conflicts lead to security policy inconsistency.

In general, normative can be implicit or explicit while security policies are explicitly mandatory normative. Procedures are step-by-step defining how things must be done considering specific contexts. It is also a guide to what one should do if a particular condition is met.

A procedure is a way to provide minimum safeguards (administrative, technical, physical or all of those) that are employed to protect sensitive information. It is expected from such procedures to be technical, and to be intended for information custodians, systems administrators and information technology personnel within the organization.

5.2.3.2 Auditing

Consider that auditing information security is a process that takes measures in terms of qualitative and quantitative assessing the current state of what is being audited regarding specific criteria of information security.

Auditing is key to discover risks, technical flaws, policies, procedures and normative problems. Bear in mind that auditing is an everlasting process and when auditing information security, one should be prepared to cover topics from physical security of data centers to logical security, including network perimeters, system configuration and information systems. Each one has its peculiarity, thus different methods for auditing should be taken into consideration and should also be applied to guarantee the auditing results.

Always assume that information security auditing is for information security professionals and the entire auditing process should be part of an overall plan. Auditing will provide at least an independent review of the adequacy and effectiveness of the internal controls regarding business process, people behavior, infrastructure current state, policies, normative and procedure compliance.

Auditing is also an operating measure to verify the state of what is being audited in a specific period of time so it can verify previous behavior, but will not guide the organization to all future possibilities.

5.2.3.3 Continuous Monitoring

Continuous monitoring keeps track of ongoing knowledge of information security, vulnerabilities, and threats [D+11], thus their associated risks. It is key to support decisions regarding risk management. It also guides decisions regarding protective and proactive measures.

This piece of the puzzle is important because compliance with international normative and standards does not guarantee that the organization's security objectives will be achieved. In practice, using normative and standards will help to achieve higher levels of maturity regarding information security.

Monitoring systems begins with the definition of what, how, when and why monitoring information assets or any part of the architecture. It relies on technology, processes, procedures, operating environments and people. It is deeply connected to the understanding of organizational risk tolerance. It also helps to set priorities and to consistently manage risk inside the organization. It should be standardized so one is able to give visibility and identify security status at all monitored information assets.

The task of monitoring can be perceived as an active security component [HTK13]. It can sense the infrastructure's current state. It also considers a wide variety of data and information in order to react according to specific defined polices as a way to help to protect the information within the organization. Continuous monitoring should be based on measures such as protection, sense, adjustment, collection, alarm and others related to information security.

5.2.4 Trust Layer

When it comes to security, trust is zero or one. You trust your information systems, network, etc. or you do not; "Maybe" should be avoided. Usually, trust can be acquired by empiric observation, by formal proof of the systems and its mechanism involved and other techniques [Lam01]. Once all expectations are fulfilled one may establish trust.

The problem is that trust, by its own means, is an expectation. It is a probability that things will work and keep working as they are supposed to.

Considering a security perspective, for something to be trusted it must be clearly identified and operate exactly as planned and expected. It must also not do anything it was not supposed to do and must be able to operate nonstop. If a trust approach is taken and it is acknowledged that a system may be or has been compromised, it is enough to make it suspicious or untrustworthy. Vulnerabilities, exploits, APTs are ways of mining trust in information systems. Details of exploits and APTs are in chapter 6.

Trust and security are closely related [Lam01]. If security objectives are considered, it is clear that trusts are connected to security because information security depends on people and security extensions such as authentication, authorization, access control, non-repudiation, etc. However, in practice, information security approach should be - "trust, but verify." [The13]. And always verify, apparently even when there is nothing suspicious. This leads back to the idea of continuous monitoring.

In general, trust is the piece of the puzzle in which there is enough knowledge about information, systems, technology and other components to help one to make assertions such as "fully secured" or "information is secure because a particular condition was met". This layered approach allows one to deal with specific components and isolate issues related to each one of them.

5.3 Information Representation and Treatment and its Relation to Information Security

When it comes to security, type of information, operations that can be performed regarding it and the supporting components must be considered if you want to protect them. In fact, you cannot protect information if you do not understand where it is stored, how it is represented or who or what manages it. In other words, how information is digitally treated is the key to trying to protect it.

Figure 5.2 depicts how information is treated regarding its basic representation, the basic operations to process it and where these activities may digitally happen from a digital perspective. If the whole complexity of Figure 5.2 is considered, it sounds very difficult to guarantee or to achieve information security.

From the perspective of this research, a trust layer is used as a means of explaining when technology or regular approach by themselves are not enough to guarantee security during the entire process. For instance, from users' perspective things should work as long they are supported by an information security architecture as described in previous sections.

Considering the digital representation of information, types of information can be images, text, voice or sounds, videos (combination of images and sounds). When it has no particular characteristic, it can be perceived as data (which may be a set of digital representation, such as database files or proprietary data types).

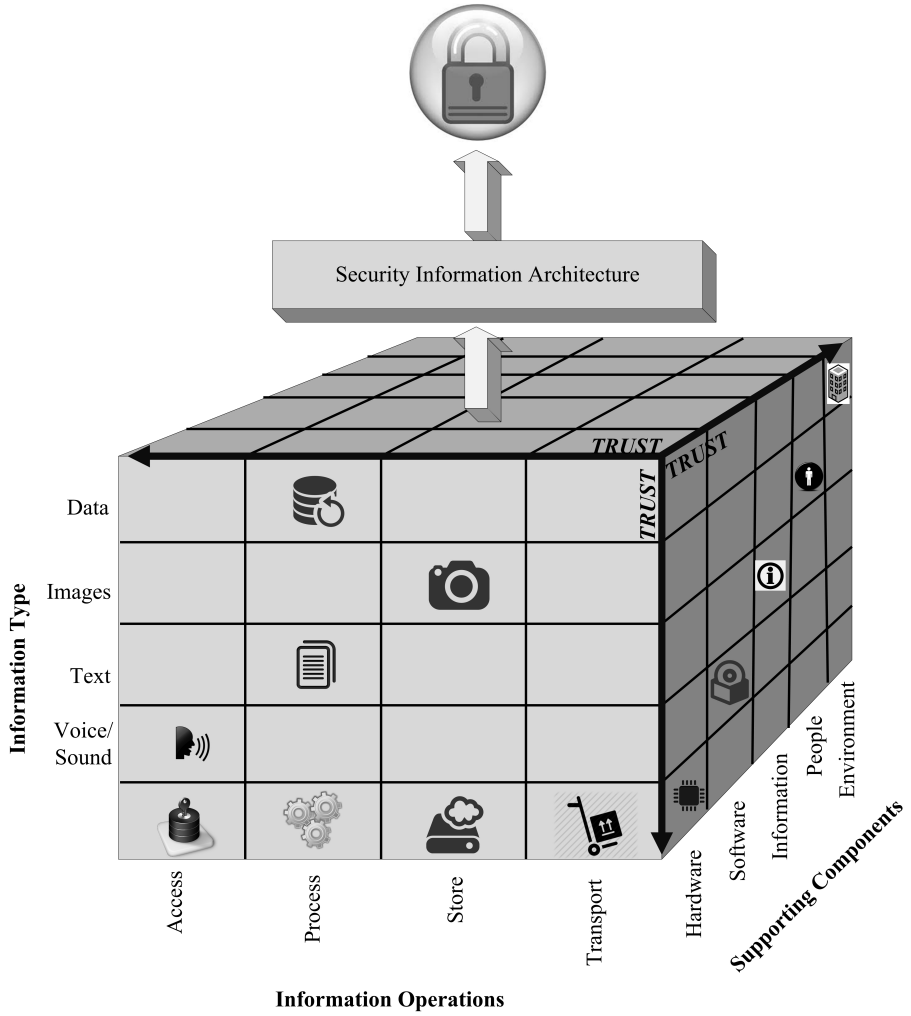


Figure 5.2: Types of information, operations and components

Once the information has its representation regarding a particular information type or a combination of them, it may undergo specific operations in the digital environment. Operations such as access data, process text, store voice records or transport any of those from one place to another using any kind of media (disks, cell phones, tablets, networks, etc.) are very common in modern organizations.

Any combination of such information operations is also possible, for example: one particular user may wish to transform a text file into a voice file or vice-versa, or a particular user may wish to describe an image giving it more sense than the previous one. All these processes are possible and very common in digital environment, and there are a lot of possibilities when it comes to the combination of information type and information operations. Thus, protecting it in all existing scenarios is rather difficult. It can be noticed by every piece of the cube towards information security (Figure 5.2).

The supporting components are where or how the information is used. For example: a video is stored in hardware and uses proprietary software to play it. It is also where people directly act regarding information. For example: if a user copies a file from the database to its thumb drive disk through the network, different supporting mechanisms are applied so the user is able to achieve its goal.

The entire cube (Figure 5.2) depicts how information is digitally represented and treated. Therefore, it should be considered that information security should be guaranteed inside environments where the follow activities take place: copying files, printing images, writing data to database systems, using distributed network environments, network connections of all types are supported, users making common mistakes such following unknown hyperlinks, and so on.

According to a very simple approach, it is a complex matrix of at least three dimensions to keep one piece of information secure inside such environments. That happens because one should consider the type of information, which operations it may undergo and what information support is used

In the view of this work, it is believed to be very difficult to achieve information security regarding the cube perspective depicted in (Figure 5.2) without using an information security architecture as proposed in section 5.3 and its understanding, besides the correct application of techniques to secure information.

According to [GMP13], [Ric13] and [The13], using technology or being compliant with standards is not enough to keep information protected. It needs more to be accomplished and, in such cases, trust should be used even without knowing it in first place.

5.4 Comparing Information Security Architectures

It is very hard and also complex to compare information security architecture in general. There is no common approach or benchmark able to measure every peculiarity of each known architecture. Doing so may still be incomplete to address every aspect related to information security.

The work [VO14] addresses the problem of how hard it is to compare information security models. In general, it recommends first performing the high-level comparison based on criteria definition. Then address that it is also important to create an appropriate selection of candidates, and then perform a more time-consuming extensive evaluation.

The work in [HCCT03] also reviews some information security standards and architectures in terms of security policy theory, risk management theory, control and auditing theory, management system theory and contingency theory.

In [SAT11], authors reviewed various information security standards and compared major information security standards (ISO27001, BS 7799, PCIDSS, ITIL and COBIT) in terms of information security policy, communications and operations management, access control, information system acquisition, development and maintenance, organization of information security, asset management, information security incident management, business continuity management, human resources security, physical environment security, compliance. The authors [SAT11] conclude that each standard plays its own role and

position regarding information security management systems.

Having the assumptions of previous works in mind and taking a trust approach, Table 5.2 compares the proposed architecture with two major widely known information security guides

Table 5.2: Comparison criteria of architectures

Criteria	ISO 27001	BS 7799	TISA
Trust as an important role	No	No	Yes
Address security extensions	Yes	Yes	Yes
People as an important role in security	Yes	Yes	Yes
Continuous monitoring activities	Yes	Yes	Yes
Auditing activities	Yes	Yes	Yes
Address Security Compliance	Yes	Yes	No

5.5 Synthesis of the Chapter

This chapter provided the reader with the developed architecture (TISA) to provide a closer view of the use trust to information security. This part of this thesis tries to focus on the problem that information security is a risk management activity, but it also discusses that current information security architectures lack proprieties to deal with the different facets of information technology correctly. In order to contribute to the discussion in information security and trust, a layered information security architecture was presented connecting together elements of information security with trust directly. The presented architecture aims to help deal with information security and trust from different points of view in order to help protect information.

It is expected that TISA is useful to help manage information security at different levels regarding information, so information security can be analyzed as a whole and not just as a part of a small component, as it is commonly seen in organizations.

Chapter 6

Cyber Security and Trust

This chapter provides further discussion on three connected fields: trust, cyber security and information security. In cyber security there are hazards such as exploits and APTs, which are discussed in sections of this chapter. As seen in chapter 2 and 4, when emphasis is given to trust, it has different lines of understanding because of its subjective evaluation. In general, this chapter details discussions about trust and cyber security and its relations to information security (provided in chapter 5). This chapter also details aspects regarding APTs and exploits and how they are being used in cyber space.

The following sections are organized as follows. In section 6.1 there is the initial considerations and some reviews regarding cyber security. Section 6.2 presents a discussion about cyber security and information assurance. Section 6.3 considers what is important when dealing with cyber security strategy and section 6.4 presents the areas of interest in cyber security strategy. Section 6.5 discusses exploits and section 6.6 refers to APTs discussions and references. Section 6.7 discusses the relationship between information security, trust and cyber security. Finally, Section 6.8 presents a synthesis of this chapter.

6.1 Initial Considerations

Protecting cyberspace is vital to keep trusting Internet and most of the information systems the way it is known. Most of it is because cyberspace is based on the uninterrupted availability of the Internet and because information systems are expected to work efficiently. Cyberspace is considered as critical infrastructure to many countries these days. Thus, dealing with threats in this environment is important and affects areas in society such as economics, health, energy, among others. It is clear that exploring these environments in ways that are considered invasive to privacy also mines trust applied to them.

When things such as exploits, APTs, cyber campaigns, and so on, that are part of day-to-day activities in Internet, one may start thinking if cyber security is really being taken care of as it should be or if Internet and Information Systems are as reliable as people think they are. Cyber security is inserted in an environment where daily threats are real and they do have destructive potential. Seen from this perspective cyber security is tied to management, policies, physical infrastructures, virtual environments and collaboration

among different parties, which needs trust to be developed.

6.2 Cyber Security and Information Assurance

It is a common understanding that information assurance is supported by multidisciplinary and multidimensional subjects besides information security. From the perspective that information assurance is constituted of operations and defensive measures, we can infer that it also deals with proactive and sometimes offensive activities. This section reviews and presents some definitions about cyber security and information assurance in order to bring more value to the discussion.

6.2.1 Cyber Security

From the perspective of this work, cyber security is considered the set of technologies, processes and practices designed to protect computer networks, computers systems, hardware and software, information and data from attacks, damage or unauthorized access. In information technology, the term security implies cyber security.

Cyber security is also a field where information security and computer network security are closed together. It deals with application security, end-user systems and information. Due to its characteristics, cyber security constantly changes and requires coordinated efforts throughout an information system environment. In such scenarios, threats advance quicker than a particular organization can keep up with.

Generally in cyber security, threats change faster than the main idea of the risk itself to a particular system. Thus, it is very difficult to address risks in cyber security because once it is addressed the threat may already be something else. From such a perspective, according to [BAG15], ensuring cyber security is not a simple task. It requires domain knowledge and cognitive abilities to determine possible threats from large amounts of network data.

6.2.2 Information Assurance

Information assurance has strong relations with information security and business continuity. It is an interdisciplinary field that requires expertise in accounting, user experience, fraud examination, business, computer and network forensic science, management, systems engineering, security engineering, and depending on the case, even in criminology, in addition to computer science. It may be seen and understood as a superset of information security.

This work considers that information assurance is an area that seeks to protect and defend information and information systems supported by computer systems and networks. It uses means such as tools, processes and frameworks to try to ensure confidentiality, integrity, authentication, availability, and non-repudiation. The practice of assuring information and managing risks are related to the use, processing, storage, and transmission of information or data. The systems and processes used for these purposes

in order to guarantee that authorized users have access to authorized information at the authorized time are also subject to information assurance.

Also this research considers that information assurance includes the use of physical, technical and administrative controls to accomplish the task of data protection. And this encompasses not only digital but also analogue or physical forms of data. It can be applied to data in transit, as well as data at rest. Information assurance is also a method of adding benefit to business using information risk management activities. It is expected to increase the utility of information to authorized users and, of course, to reduce the utility for those unauthorized.

Considering a wider picture, using critical infrastructure as example, the cyber security context is embraced by information security itself as part of efforts towards information assurance. As a means of creating a representation of the steps and areas it implicates, Figure 6.1 summarises this view in a top-down approach.

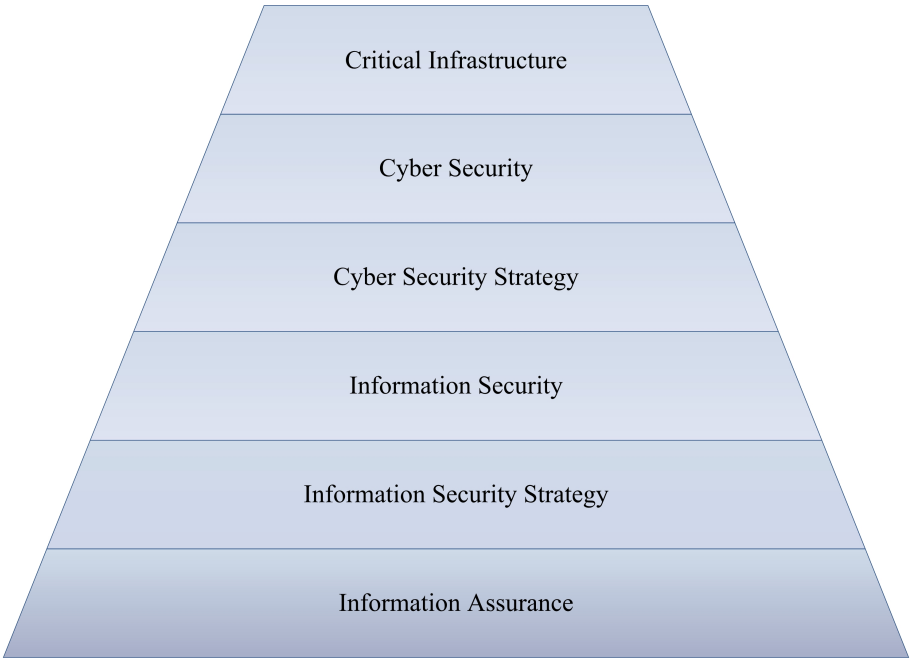


Figure 6.1: Implications of cyber security towards information assurance

6.3 Cyber Security Strategy

Cyber security strategy is a high level plan to achieve conditions of security in cyber environment, for example, where critical infrastructure systems are placed. Cyber security is seen as a strategic area by many countries.

For example, Table 6.1 details the time when cyber security strategy became part of overall international planning by some countries around the world, according to the ENISA and North Atlantic Treaty Organization (NATO) Cooperative Cyber Defence Centre of Excellence.

Table 6.1: Summary of since when some countries have overall planning of cyber security strategy

Country	Year
Qatar, Nigeria, Costa Rica, Jamaica	2015
Czech Republic, Estonia, Denmark, Latvia, Kenya, Namibia, Rwanda, Azerbaijan, Bangladesh, Pakistan, United Arab Emirates, Ghana, Mauritius, Grenada	2014
Austria, Finland, Italy, Hungary, Poland, India, Japan, Montenegro, Singapore, Turkey, Zimbabwe, Romania, Spain, Cyprus, Saudi Arabia, Egypt, Marrocco, Panama	2013
Belgium, Norway, Switzerland, Netherlands, Georgia, Jordan, Mauritania, South Africa, Trindad and Tobago	2012
France, Germany, Lithuania, Luxembourg, United Kingdom, New Zealand, South Korea, Uganda	2011
Canada, Mongolia	2010
Australia	2009
Slovak Republic	2008
Malaysia	2006
United States of America	2003
Russia	2000

6.4 Cyber Strategy Areas of Interest

In order to better discuss and understand what cyber security strategy includes, this section is divided into multiple subsections in areas that are important to consider when preparing a strategy to safeguard cyberspace. Things such as threats in cyber space, analysis of huge volumes of data, etc. are subjects of interest when thinking about cyber strategies development.

6.4.1 Cyber Threats

Cyber threats go beyond aspects that deal with national security or military defense. First have in mind that there is no silver bullet. Cyber threats are related to critical infrastructure systems, computer networks, financial systems, research facilities, industry, etc. Thus, to address cyber threats it is necessary to have integrated efforts, multidisciplinary and specialized knowledge and skilled people. The knowledge required to deal with cyber threats requires contextualization, means and indicators, direct and indirect implications, warnings, monitoring new threats to assets and critical infrastructure systems. All this is used together to support strategic decisions and measures that may be applied in each situation.

Such analyses are impossible without humans, because they require deep knowledge in information security and communications, information assets, auditing, engineering, and what else might be necessary, such as politics and attribution of attacks. Thus, to be able to monitor and follow cyber threats, some processes may be automated but it is no guarantee of efficiency because, to the best of our knowledge, there is no such automated system that may be able to protect, adapt and learn efficiently from everything

that is required to address such problems. On the other hand, the amount of data is such that it is impossible for humans to process it without information systems, security systems, correlational systems, big data analytics, etc., but all these systems cannot be fully automated and integrated with current technology.

To deal with cyber threats, the focus has to be with processed information, ordered information, information treatment, cyber intelligence, reliable sources, information correlation, decisions without complete knowledge, tempested answers, global players, nation-state efforts, organized crime, hacktivism activities, and other subjects, which depends of each case.

6.4.2 Cyber Attacks

This work believes that cyber attacks are offensive maneuvers conducted by individuals or organizations that use technological resources with the intention to deactivate, disrupt, data theft, vulnerability exploitation, implant malicious code or cause damage to critical infrastructure systems, computer networks, information systems, hardware, software or any technical resource of the attackers' interest. A cyber attack is a planned exploitation of a computational system related to particular technology. It is important to remember that a cyber attack may be just an ordinary attack or a step that is part of a particular cyber campaign, cyber espionage, cyber terrorism or even non-declared cyber war.

Considering observations over time, such attacks will become more sophisticated and disruptive in the future (most of them already are). Normally a cyber attack uses malicious codes, bad assembled packets in networks, disguised packets, hidden codes in computer systems, furtive techniques in order to accomplish its objectives. Cyber attacks also deal largely with exploits (section 6.5) and advanced persistent threats (section 6.6).

When cyber attacks take place, the attention is related to denial of service attacks, distributed denial of service attacks, suspicious code in computational systems, botnets, network misbehavior, reverse engineering, penetration tests, breaking computational codes, among other activities.

6.4.3 Network Flow Analysis

There is no point in arguing how computer networks are widely used nowadays. They have its bases in communication infrastructure and protocols to send and receive data. Cyber attacks make use of computer networks to accomplish data theft, systems exploitation, implant malware and so on. It is important in a cyber security strategy to understand and be able to perform network flow analysis. By using such capability one may be able to understand standard network behavior, detect anomalous traffic, the network flow in order to detect indicators of compromise of the computer network.

The focus of such activities relies on behavior network analysis, protocol analysis and understanding communications protocols, collecting and executing network flow analysis, processing network flow reconstruction, temporal analysis, and other activities.

6.4.4 Malware Analysis

From the perspective of this work, malware analysis is the study of suspicious or malicious code using reverse engineering and behavior analysis in order to understand and discover the code functionalities and the techniques that were used to build such codes. Malware analysis allows one to understand what the main purpose of malicious code is so it can be used do better deploy security measures and deal with incident response. It is also used to learn how offensive techniques are applied in computational environments.

The effort towards malware analysis basically is divided into static malware analysis and dynamic malware analysis. However it is required deep knowledge and skills in reverse engineering, behavior analysis, coding and decoding techniques, among other areas.

6.4.5 Big Data Analytics

The amount of data generated, for example, by the traffic inside an ordinary communication network, is such that it is impossible to be processed or analyzed by simple tools. Much of the data is not structured, or semi structured to be intelligible, but it is not restricted to it. In most cases, even if the data is structured, the amount of raw information is such that one cannot analyze it without algorithms and expert systems.

Big data analytics have the capability to exam large data sets with a variety of data types. Once data is processed it can uncover hidden patterns, unknown correlations, find trends, preferences and other useful information that is important for security analysis. With such tools, security and analysis experts can analyze large volumes of data that may be unused by conventional business intelligence solutions.

This kind of data includes server logs and Internet data, social media content and social network activity, open source content, mobile data, records and data captured by sensors disposed in network points of interest. Big data analytics uses software tools as part of advanced analytics disciplines such as predictive analytics, data mining, text analytics and statistical analysis. Software and data visualization tools are all part of the analysis process.

Big data focus considering a cyber security perspective embraces massive data visualization, open source and distributed system analysis, data processing infrastructure, heterogeneous data source, analytics algorithms, parallel processing, anomaly detection, and other research areas of interest.

6.4.6 Underground Cyberspace

Much of the Internet is not indexed in search engines like Google or Yahoo. Projects as Memex [Dar14] aims a break through paradigm in Internet search capabilities. Web environments such as closed forums, web pages not indexed, unavailable public web servers and anonymous networks are sources of information that may help mitigate or even anticipate and prevent cyber attacks.

P2P, BitTorrent, The Onion Router (TOR), Invisible Internet Project (I2P), closed forums, anonymity networks, deep web and darknet are subjects of interest in cyber security studies.

6.4.7 Cyber Alliances

To be able to go further in cyber security, it is important to keep in mind that creating cyber alliances with like-minded actors based on common sets of practices and principles is key to advance in security. With cyber alliances one may be able to share information about threats and responses, train both civil and military defense authorities and conduct joint cyber exercises.

These alliances also permit discussions in cyber terrorism, and best practices to protect critical infrastructure systems. Intelligence and law enforcement agencies may discuss and share practices in searching and blocking threat actors, for example people that are part of organized crime and use black markets to buy and sell illegal goods. Cyber alliances also include private sector, international law enforcement and nongovernmental organizations. In short, cyber alliances should be expansive and go further than state-to-state diplomacy, if a wider view is considered.

The focus of a cyber alliance is related to establishing partnerships with centers of recognized capacity in information security, research and development institutions, founding of solutions of cyber security, joint information security projects, transfer of technology among participants, and others interests.

6.5 Exploits

Basically, cyberspace is a place where threats like exploits and APTs are used to gain benefits, profit or strategic advantage. It all depends on the actor that is behind the deployment of such techniques. This section explains exploits and the following section (6.6) explains APTs.

Exploits basically are tools used in cyberspace. They can be considered cyber weapons as well, depending on its capabilities and objectives. An exploit can be seen as a threat in a cyber security perspective. The sections below review and explain details about exploits and a study of the market associated with it.

6.5.1 Definition

An exploit is a piece of code developed and compiled with a particular hardware architecture, a particular software or application in mind. This code is created with the intention of exploiting a particular vulnerability in some technological resource. In such cases vulnerability is a kind of failure that permits the attacker to have success in exploiting a particular resource, be it hardware or software.

The associated vulnerability may be Zero-day (0day) or maybe a known vulnerability. 0day vulnerability is a kind of failure for which there is no previous defense against it because there is no patch or update for the technology being exploited. Such failure allows attackers, considering its objectives, a high level of success in exploring such resource. Once this failure is publically known, it is no longer a 0day, but it maybe still be exploitable because there may be no patch or update for it. In other words, the vulnerability may be

widely known but it is not fixed, thus it represents an opportunity to the attacker and a risk to the target of such tools.

Anyway, an exploit makes use of low-level language, may use assembly instructions that are able to manipulate and change the normal flow of execution of a program. Such changes allow an attacker to execute arbitrary code in the system exploited thus permitting control over the system functions.

From a security perspective, an exploit is a threat that may be used in combination with another piece of code to gain control over the technological resource of interest to the attacker. If an exploit is well deployed and well controlled it allows unauthorized access to computational resources, permitting privilege escalation or denial-of-service to authorized users. If it is badly used, an exploit is a lost resource, may allow identification of the attacker and may be no longer efficient.

An exploit can be classified as local or remote. A local exploit is when the attacker already has access to a particular computational resource and may execute the code locally with intention of privilege escalation, install some other code or make the computational resource remotely controlled. A remote exploit is when the attacker may execute the code remotely by use of a communication channel or network, thus exploring vulnerability to gain control over a particular system.

In short, an exploit is a technical resource that can hit any user in any cyber environment, either connected or isolated. In isolated environments removable media can deliver an exploit for example. An exploit alters hardware or software normal functions to accomplish what the attacker desires. Having said that, there is a market where such goods are traded.

6.5.2 Exploit Market

From the perspective of this work, if one considers that an exploit is based on a market approach, it is simply based on the fact that there are buyers and sellers. There is one side interested in buying such goods and there is the seller of such goods. But the use of exploits is very restricted. Normally it is linked to a particular technology. Exploits are supported and developed by organized crime, hacking activities and nation-state actions. There are also specialized actors that research vulnerabilities and create exploits for them.

It is important to understand that an exploit is not an end in itself. It is a way to perform much bigger activities. Developing exploits is a complex activity, it requires high expertise in information security, it deals with very skilled people, it is constantly developing, and may be high-priced in specialized markets. The participation in such markets as buyer requires planning, focus and concrete objectives. It demands a good strategy and needs proper infrastructure to deal with such activities. An exploit may be the guarantee of success in exploring some particular resource. Or it may be the cause of failure if misused.

What basically defines an exploit market is the law of supply and demand. An exploit can be acquired from specialized sources, people may exchange it in closed negotiations, or it may be even sold to Government agencies by private companies. It all depends on the needs of each player. Sometimes it is a legal market (conducted according to the law)

where the prices are very high. Sometimes it is conducted in illegal markets, where there is no control or the law is always forgotten and most of the time this trade is conducted in closed forums, using virtual currency and encryption techniques.

This market is basically conducted by the discovery of vulnerability that the manufacturer normally is not aware of. The value of an exploit for unknown vulnerabilities may reach up to U$ 500.000,00 according to the study conducted by Forbes Magazine [Gre12]. The values are also variable because it all depends of its utility, reach of the exploit or exploitation difficulty. This market is also connected to cyber crime with impact in finance and revenue, which reaches orders of 40% in losses according to the study conducted in [HP 12]. Basically, what is seen in such environments is that when an organization is victim of a cyber attack, where exploits or an APT are used, it is highly probable that the organization will not be able to mitigate the attack before the damage is already done.

It is very hard to define the real cost of an exploit. Things such as complexity of the exploit, its effectiveness, the target, the opportunity window (see APT in section 6.6), target localization, penetration difficulty (network perimeter, objective of the attack, etc.) are factors that influence the price of an exploit. Table 6.2 summarises some findings conducted by [Gre12].

Table 6.2: Mean Price of an exploit according to Forbes Magazine Study in 2012

Product	Estimated Price
Adobe Reader	U$ 5.000 - 30.000
Mac OSX	U$ 20.000 - 50.000
Android	U$ 30.000 - 60.000
Flash or Java Plugins for web browsers	U$ 40.000 - 100.000
Microsoft Word	U$ 50.000 - 100.000
Windows	U$ 60.000 - 120.000
Firefox or Safari	U$ 60.000 - 150.000
Chrome or Internet Explorer	U$ 80.000 - 200.000
IOS	U$ 100.000 - 250.000

Other interesting point of the exploits market goes beyond the price of exploits or its kind. It is based on who is paying for such threats. According to a broker known as Grugq, most of the clients are occidental government (USA and Europe) simply because they pay more than China or Russia [Gre12]. Another important thing to remember is that when there are many offers, the price tends to decrease because there is too much competition. Besides that, the quality of such goods may be doubtful if one has the ability or not to check the effectiveness of what one is buying.

Symantec Labs studied cyber crime and cyber attacks [Sym14]. It showed that, along with other information related to security, in the year 2013 there was an increase of 91% in targeted attacks, an increase of 62% of vulnerabilities, more than 552 million identities were exposed, at least 23 0day vulnerabilities of high impact were publically discovered, and one in 392 emails had stealing passwords attacks.

A study conducted by Goncharov [Gon14] shows that many parts of the Russian underground are highly specialized. Hackers with good social network contacts do not have to create all his tools anymore. He may rent or buy it from others. There are experts for almost everything one might need. There are service offerings for deny-of-service or distributed deny-of-service, traffic redirection, pay-per-install, malware development, etc. The study [Gon14] also points that, for example, the price paid by valid credit cards are on the fall (Table 6.3) as are the price of stolen credentials (Table 6.4).

Table 6.3: Price of stolen credit card data in Russia market per year [Gon14]

Country of origin of the stolen data	2011	2012	2013
Australia	U$ 7	U$ 5	U$ 4
Canada	U$ 5	U$ 5	U$ 4
Germany	U$ 9	U$ 5	U$ 6
United Kingdom	U$ 7	U$ 6 - 8	U$ 5
United States of America	U$ 3	U$ 1	U$ 1

Table 6.4: Price of stolen credentials of web services per year [Gon14]

Service	2011	2012	2013
Facebook	U$ 200	U$ 160	U$ 100
Gmail	U$ 117	U$ 120	U$ 100
Hotmail	U$ 107	U$ 100	U$ 100
Mail.ru	U$ 74	U$ 70	U$ 50
Twitter	U$ 167	U$ 40	–

There is another market that is part of exploits and vulnerabilities. In 2013, the NSS Labs conducted a study that analyzed data from two major bug bounty programs [Fre13]. The results show that in the last 3 years (before 2013), at any given day, privileged groups had access to at least 58 vulnerabilities reaching systems such as Microsoft, Apple, Oracle or Adobe and those vulnerabilities remained private for at least 151 days. Table 6.5 resumes some data reported by the study [Fre13].

Table 6.5: Mean price paid by bug bounty programs [Fre13]

Company	Price Paid	Description
Google	U$ 580.000	Mean price paid for at least 3 years for 501 vulnerabilities disclosure in Chrome web browser. Corresponds to 28% of updates on the same period.
Mozilla	U$ 570.000	Mean price paid during 3 years for 190 vulnerabilities disclosure in Firefox web browser. Corresponds to 24% of updates on the same period.
Facebook	U$ 1.000.000	Mean price since 2011 for Facebook bug bounty programs.
Microsoft	U$ 130.000	Mean price paid since 2013 by Microsoft for new exploitation techniques in its products.

Also, regarding bug bounty programs competition, the CanSecWest Pwn2Own 2015 had payouts for vulnerabilities in four major browsers (Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari). The payouts for all competitors reached U$ 442.500 in two days of competitions [Bro15].

The exploit market keeps running so far. There are many sites specialized in selling exploits. Such sites are available in networks using anonymous access such as TOR or I2P. As another example, U$ 17.000 in bitcoin was the price of a new method of hacking Apple iCloud accounts [Gre15]. Other prices of exploits to be used against Android's browser, to attack Internet Explorer browser in Windows XP, Windows Vista and Windows 7, are available for around U$ 8.000 in bitcoin [Gre15].

An online article from help net security [Hel15] shows that the price of stolen information sold on underground market from users and corporations also vary depending on the good itself. Table 6.6 shows some of them [Hel15].

Table 6.6: Price List of Stolen Information [Hel15]

Type of information sold	Price
1.000 stolen email address	From U$ 0.50 to U$ 10
Stolen Gaming Accounts	From U$ 10 to U$ 15
Stolen Cloud Accounts	From U$ 7 to U$ 8
Scans of Real Passports	From U$ 1 to U$ 2
Custom Malware	From U$ 12 to U$ 3.500
Credit card details	From U$ 0.50 to U$ 20
1.000 Social Network Followers From	U$ 2 to U$ 12
1.000.0000 Verified Email Spam Mail-outs From	U$ 70 to U$ 150
Registered and Activated Russian Mobile Phone SIM Card	U$ 100

Also, with recent values (dated from 2015), some tools used to specific systems and offered privately by one hacker were found [1]. Table 6.7 summarises some values offered by the hacker responsible for the page. It is important to mention that the products in this case are not 0day, but they target specific systems versions and platforms, which are dependable of the exploit itself and its efficiency.

6.5.3 Basic Process of Exploit Usage and Development

An exploit can be used as executable code for a particular system architecture (hardware and operational system), can be deployed throughout web pages, it may be an attached file to an email message, it can be as a text message in a cell phone or it can be embedded in digital files with a lot of different extensions.

Once it is executed, an exploit allows an attacker to carry out different desired actions. It can control the system locally or remotely; interrupt its functioning, exfiltrate data or do anything that the computational resource permits.

The development of an exploit goes through specific activities that vary according

[1] Found at http://apt0.no-ip.biz and visited on March 2015

Table 6.7: Exploits price offered by a single hacker

Tool	Target System	Price
Exploits, Penetration Testing Kits, Silent Infection	Firefox 22-27	U$ 200 (Reduced)
Mozilla Firefox Bootstrapped Code Execution	Firefox addon (Windows 7, Windows XP, Window 8.1)	U$ 400 (Reduced)
OLE automation array remote code execution	Internet Explorer $\leq$ 11	U$ 800
WolfPack	Java 1.6.0 Java 1.7.0_06 Java 1.7.0_10 Java 1.7.0_17 Java Applet (Windows 7, Windows XP, Window 8.1)	U$ 1000
Exploits, Penetration Testing Kits, Silent Infection	Firefox 31-34 (Windows 7)	U$ 800
Polymorphism IE11 Exploit Source Code	Internet Explorer 11 (Windows 7)	U$ 1200
Polymorphism Firefox 31-34 Exploit Source Code	Firefox 31-34 (Windows 7)	U$ 1000
Insanity - Infection Kit	Pentest infrastructure location	U$ 2000

to the system of interest of the attacker. In order for a particular vulnerability to be exploited, time, skills and domain of the target technology are all necessary. Sometimes it requires even being able to fully duplicate the target system. Some of this information should be gathered by other activities, depending on the case.

If it is considered an action supported by a particular demand, an exploit may be locally developed or bought from sellers in specialized market. But it all depends on the opportunity window of the vulnerable in the target system. The simplified process of exploit deployment can be seen in Figure 6.2 (created from the initial perspective of [Tie14]) and Table 6.8 resumes each process phase.

The opportunity window (Figure 6.2), where the resource is exploitable, is variable in time, depends on various aspects such as update available time, discovery time, failure complexity, and other details.

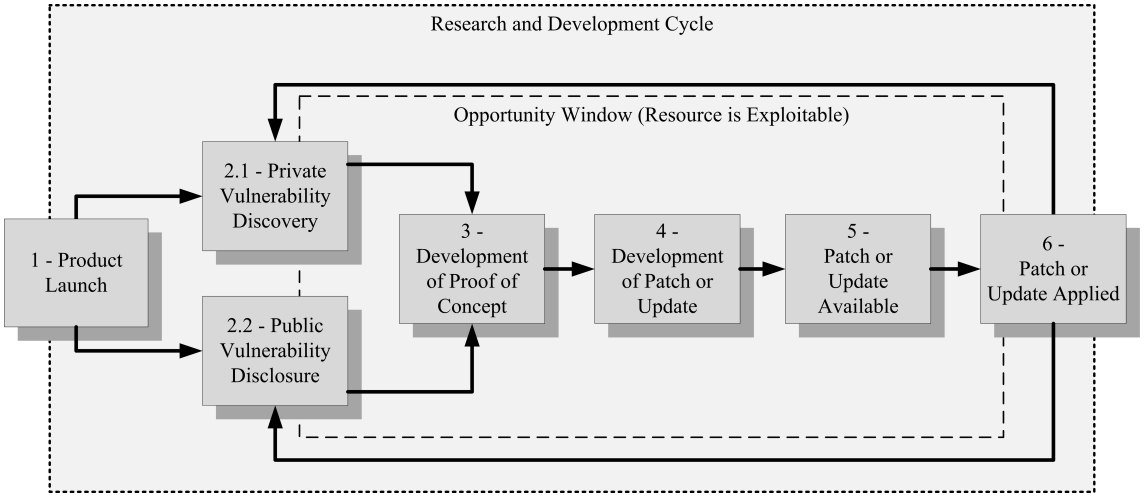


Figure 6.2: Exploit process lifecycle

Table 6.8: Exploit process lifecycle resume

Phase	Description
1 - Product Launch	A manufacturer makes a product available in information technology business market. This product may be hardware, software, application, operational system, client-server platform or any technological resource.
2.1 - Private Vulnerability Discovery	Through research and development, a private organization or researcher, discovers a particular vulnerability. If it is not made publicly available it is a Zero-day (0day) vulnerability.
2.2 - Public Vulnerability Disclosure	It is when vulnerability is discovered and made publically available in the Internet or specialized sources. At this phase, normally the manufacturer is previously aware of it and develops a patch or update for its product before vulnerability announcement in the Internet. Notice that before the patch or update the product is vulnerable to exploitation, which is an indication of the opportunity window.
3 - Development of Proof of Concept	It is when a Proof of Concept (PoC) code is made available to demonstrate that the vulnerability in question is exploitable. Normally the PoC is a step towards the creation of an exploit.
4 - Development of Patch or Update	The manufacturer develops a patch or update so the product cannot be efficiently exploitable by the known vulnerability.
5 - Patch or UpdateAvailable	The product manufacturer makes the correction public available so users can deploy the patch or update in its environment.
6 - Patch or UpdateApplied	Users do apply the patch or update in their system so it cannot be exploited by the known vulnerability.

According to a study by Bilge and Dumitras [BD12], the mean time of Zero-Day detection is about 300 days before the resource is already in phase of exploitation, which gives an idea of the opportunity window time. The duration of attacks that explores Zero-Day vulnerability may last from 19 days to 30 months and normally a Zero-day attack is directed to a particular target, so its discovery is restricted, complex, and largely depends on the target itself.

6.6 Advanced Persistent Threats

It seems that the American Air Force General Greg Rattray created the term APT [Gol14]. It was used to designate any adversary engaged in technological conflicts of long duration with planned and defined strategic objectives. In the cyber security area, an APT is part of planned actions and strategy, which are related to specific goals using the cyber environment to achieve its objectives.

An APT has a high level of analytic and technical knowledge and uses a large amount of resources. With such characteristics an APT creates means of achieving strategic goals using multiple vectors, be it physical, virtual, social, stealthy, or even others. Normally an APT is used to establish and keep continuous covert access, as long as the attacker desires, inside a target's infrastructure and performing the attacker's intentions. Through an APT, an attacker may cause damage, delay actions, destroy information, exfiltrate data and information, provoke wrong signals in systems, change information and system parameters. It all considers a present state or future where the APT is inserted and used.

An APT seeks its objectives during a long period of time, is able to adapt itself to new

circumstances, avoids technological defenses, and keeps in constant communication with its command and control system, even using air gap techniques, where the system is not connected to the Internet. An APT is capable of advanced exploitation techniques, uses attacks with no proper defensive measures, and may use 0day exploits as a starting point of infection, so its level of success is even higher.

An APT uses stealthy capabilities so it is not easily detected by common defense perimeter technology. It can change its behavior to bypassing detection, it may attach itself to another program, and it can even alter firmware and device drivers, so it cannot be removed from the infected resource without enormous amount of efforts. Such characteristics make an APT hard to detect thus creating counter measures efficiently is also difficult.

The use of rootkits is also very common when dealing with APTs. Rootkits allow an APT to perform privilege escalation, hide or spread itself, monitor the infected resource, capture network information, and anything an attacker may desire. It basically depends on the attacker's resources, intentions and capabilities.

Most known APTs are believed to be work sponsored by nation-state actions because of its capabilities, resources and development process. Analysis of APT shows that just one person will not be able to create such tools in a small amount of time.

Also the fact that normally most APTs have specific targets and goals, and normally, they are not related to the gain of profit in terms of money. This belief has support in facts such as high technology used, high investments, high evasion techniques and very specific targets and goals. Such characteristics are very unusual in common hacking groups or even in organized crime activities. Studies regarding APTs show that to create such tools, it requires large amount of investment.

6.6.1 APT Basic Flow Process

Considering a minimal flow an APT may use Figure 6.3 which illustrates the basic flow steps, but it is not restricted to the following logics because it basically depends on each malware. It is divided into 8 steps, and each step is described in Table 6.9.

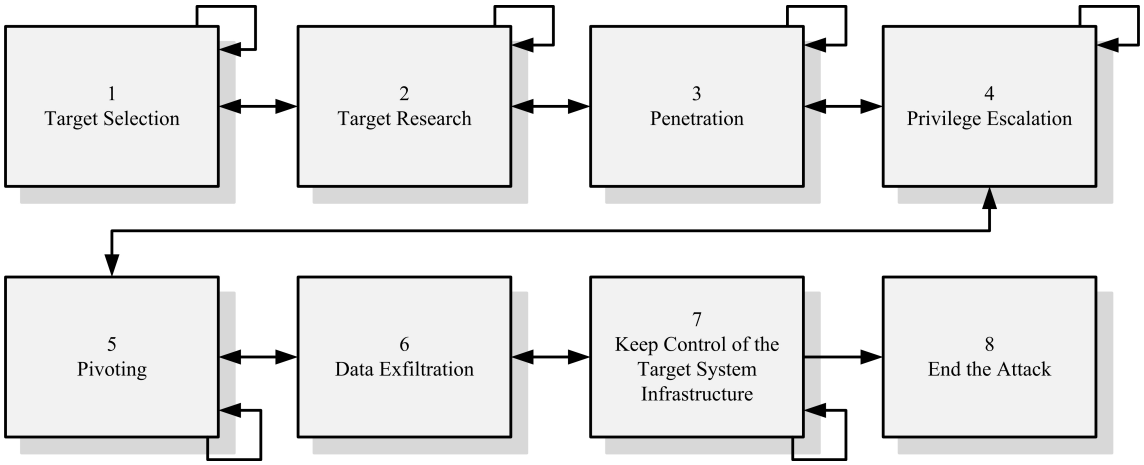


Figure 6.3: APTs basic flow process

Table 6.9: Explanation of APT basic flow process

Phase	Description
01 - Target Selection	Normally, selected targets are high valuable users, infrastructure, industry with high research and development technology, diplomatic sources, etc.
02 - Target research	Once the target is chosen, the process starts with information gathering, discovering target capabilities, collect open source information on the target, etc. In this step the objective is to detail the target information, used systems, software, hardware, defense resources, defense perimeters, etc.
03 - Penetration	In this step the target may suffer social engineering attacks, may have some of its system attacked and accessed, may receive e-mail with malware, may receive some threat from social networks relationships, may have passwords captured, etc. The objective is to access some of the target's resources. The attacker may use any means it has depending on the target and its objectives.
04 - Privilege Escalation	In this step the attacker may have penetrated some system, but it lacks administrative control on the system being explored. It may use exploits, key loggers, network sniffers, etc. to try to gain privileges so it can deploy new resources of its control on the target system or part of the infrastructure associated with it. Rootkits may be used in this step.
05 - Pivoting	The attacker spreads itself in the infrastructure of the target so he can control multiple target resources. To be able to perform such actions, the attacker may use sniffers, exploits, information gathering, exploit new systems, etc. The idea is to have multiple sources of access to the target's infrastructure, guaranteeing multiple access paths.
06 - Data Exfiltration	Once the attacker controls the system, he performs data exfiltration from the target infrastructure using communication channels, disguised packets, email, command and control systems, etc. In this step the attacker collects network information or any piece of information he considers useful to keep its actions.
07 - Keep control on the target infrastructure	After being successful in taking full control of the target system, the attacker may use rootkits to keep its activities stealthy on the victim system. The attacker may erase previous steps, advance in command and control techniques, search for new data or information, etc.
08 - End the attack	If the objective is fulfilled, the attacker may end the campaign in the particular system and fully erase all of its steps, so an investigation will not be able to fully understand the attack or even find any indication of compromise on the target system.

6.6.2 Examples of APTs, Advanced Malware and Cyber Campaign

This topic describes some known malware, cyber campaigns and APTs. These examples are merely informational with no intention of deep analysis. All information listed below is widely available online, even with multiple sources.

Indeed, some of the examples below may not fulfill all of what is required to consider such malware as an APT, but the impact it caused was also considered, alongside the complexity and targets etc.

6.6.2.1 Resume of APTs, Advanced Malware and Cyber Campaign examples

- **Titan Rain**: It was a cyber campaign that targeted American defense companies back in 2003. The investigations traced back the source as being in China. This attack called attention at the time because it already used furtive techniques and trojan horses, it also used multiple attack vectors and social engineering attacks directed to specific users. The nature of this particular attack remained restricted to government sources, but it had success hitting industries, research facilities,

aerospace agencies, finance institutions, etc.

- **GhostNet**: This was the name of a cyber espionage campaign that is believed to have started in 2009. It used Trojan horses, command and control systems, and malware that took control over the affected system. It could record audio and video of the victim and it is estimated that more than a 100 countries where attacked.

- **Cuckoo's Egg**: It is one of the first attacks established in military computer networks. Back in the eighty's, a German hacker invaded computer networks in California to steal secrets related to the American Star Wars research program. The researcher that revealed the malicious activities tracked back the origin using digital traps and discovered a satellite connection from a German University and identified a student who was likely selling the stolen secrets to the old Russia KGB.

- **Moonlight Maze**: This was an attack directed to the US government web sites back in 2000. This attack went undiscovered for approximately 2 years and compromised systems in Pentagon, NASA and the American Department of Energy. Also Universities and other military facilities where targeted. This attack was able to steal military installation maps, military hardware projects and other information. The US authorities back at the time estimated the impact of this attack in millions of dollars in lost secrets.

- **Skpyot**: This campaign was hidden for many years and it is believed to have started in 2006. It was directed at intellectual property theft including projects, manufacturing, finances, and others interests. It used Zero Day exploits back that time and targeted companies in UK and US. The areas of interest include defense, telecommunications, energy, government and chemical industries.

- **Hydraq**: It was a cyber campaign that used zero-days exploits to infect target systems. This campaign was also known as Aurora. This attack was under detection for at least a year at the time it was reported and targeted technological companies, oil and gas industries and defense sectors.

- **Stuxnet**: It is probably the most famous APT up to now because it is considered by most as the first developed cyber weapon with nation-station origin and goals. Its detection backs to 2010 but it is public appearance backs in 2008 or 2007 depending on the source. This APT was able to control low level hardware and destroyed hundreds of centrifuges in Natanz Nuclear Facility.

- **Gozi**: This was a malware that was discovered in 2007 by security experts, and it had already infected more than one million computers. It targeted countries as US, UK, France, Italy, Germany and others. It was reported that Nikita Kuzmin developed this malware and sold it in the Internet with profit intentions. This malware had a lot of variants with different purposes.

- **Zeus**: This malware used social networks as way of spreading itself. It was used to data exfiltration in targets in US and other countries. It is considered a modular

platform that uses multiple attack vectors. It is estimated that this malware could have generated more than U$ 70 million in profit for specialized groups.

- **Flame**: This was firstly public reported by Iran's Cert in 2012. It was a sophisticated cyber espionage campaign that affected Government, Universities and specific targets in Iran, Israel, Sudan, Egypt and others. It used USB cards to spread itself and could take screenshots, network traffic capture, key logger capabilities, audio recording, etc. Its command and control system had self-destruction capabilities and the attacks stopped shortly after it was public reported. It is likely that this campaign remained undercover for at least 5 years.

- **SpyEye**: It has its origins in the Russia Internet underground. Analysis reported that a command and control system of this malware had data of more than 256 different finance institutions. It is believed that a hacker from Algeria is the responsible for the first development of this malware.

- **Doqu**: This malware was revealed in 2011 in a limited number of companies that worked with industrial control systems. It had similarities with Stuxnet and a platform known as Tilded. Its command and control systems were spread over countries as Germany, Belgium, India and China as source of hiding its origin. This malware could gather information in particular industrial control system that could be latter used to attack such systems efficiently.

- **Uroburos**: Also named snake, it is a rootkit based APT malware with two main components, one device driver and one encrypted file system. It was public reported in 2014 and had data exfiltration functions, network sniffing, modular design so it can be extended as desired by its creators. This APT has evasive techniques and is hard to be detected by normal methods. It is believed that this malware kept its undercover activities for at least 3 years. Public analysis reported similarities with Agent.BTZ and the cyber campaign codenamed Turla which remained active in 2014 and target systems in United States and Europe, and East European countries as Estonia, Lithuania and Ukraine.

- **Gauss**: It was first reported in 2012 with its target in Middle East. It was considered a nation-state initiative with rootkit projected to steal sensitive information on the target system. Some analysis reported similarities with Flame as structural modules, code bases, and command and control communication capabilities. It used the same vulnerabilities used by Stuxnet and Flame to infect USB devices. But it had more functions, as for example, being able to keep hidden stolen files in USB sticks.

- **Agent.BTZ**: It is considered one of the worst security failures in US Military Computer Networks. This malware was used to attack the US Defense State Department in 2008. It was left in a lost USB Stick in the parking lot of the department at the Middle East. According to sources, this malware took more than 14 months to be completed removed from the computer network it attacked in Pentagon. Due to its impact it was considered as a military incident and was

classified as secret, so very little can be verified using open source information, but it is reported that this malware could scan networks, open backdoors and communicate with command and control systems.

- **Careto**: Also named the mask, it was reported in 2014. The analysis report showed high level of sophistication and expertise. Targets in South America, Morocco and Gibraltar were discovered. Targets of this APT were diplomatic offices, embassies, universities and government agencies. This APT can capture keyboard interruption, sniff network, record skype conversations, copy SSH keys, VPN configuration files, and other capabilities.

- **MtGox Bitcoin**: This attack stole more than 850.000 bitcoins from Japanese MtGox Company. Some security experts consider that this attack remained undercover for years without being detected. Later investigations found 200.000 bitcoins in a wallet that was assumed to be inactive. The impact of this attack at the time was in magnitude of U$ 300 million.

- **BlackPOS**: This was the malware used to attack the American discount retailer store Target. This attack expected to reach more than 100 million credit card customers. It was able to read credit card data from memory, send the data to a gateway and then to a command and control server at Russia. The investigation showed that the attack started at a third party company. From there the point-of-sale systems were infected.

- **RSA SecureID attack**: This attack was able to steal credentials used by the token SecurID. Companies listed as the 500 biggest companies by Forbes magazine used this token. Sources reported that the actions taken to mitigate the damage had costs bigger than 100 million of dollars. This malware used part of another known malware named Poisonvly, which had already attacked chemical companies and human right institutions.

- **Red October**: It was a cyber campaign designed to steal secrets from Government and research laboratories. It was reported that the campaign was active for at least 5 years before being uncovered. Its victims were in more than 35 countries with a focus on Energy, Military, Diplomacy, Space. Systems at NATO, Europe Parliament and Europe commissions were infected.

- **Volatile Cedar APT**: It is part of a group that deploys remote access tools and USB propagation components and performs targeted and managed cyber campaigns. Their targets are chosen with care and the deployment takes only the necessary to achieve the attackers goals supported by previous intelligence gathering processes. This APT can connect to command and control servers or other domains accordingly to the infection process.

- **Eurograbber**: It was a malware that was used to steal more than 36 million of euros from costumers of at least 36 different finance institutions in Italy, Spain

and Netherlands. It used Trojan horses and used a previously known malware variant. This malware could bypass SMS systems codes asking users to install security software at their mobile devices.

- **Shamoon**: This was a cyber campaign using malware targeting the energy sector. It could spread itself using network shares and perform alterations at the infected system. The oil company Saudi Aramco was the biggest target having approximately 75% of its infrastructure compromised.

- **Operation DeputyDog** : This APT used Zero-Day exploits in Internet Explorer browser where its main targets were users in Japan. This cyber campaign was reported in 2013 and had strong relations with attacks directed to BIT9 company.

- **NR4**: It was a cyber campaign reported in 2011 and attacked mainly government institutions and diplomatic offices. It used fake email accounts and its messages with political body to call attention of its victims. It had a command and control system that was used to data exfiltration. Messages used in this campaign target both English and Chinese language users.

- **The Sony case**: It was made public at November 24, 2014. But it is believed that this attack started more than a year before. This attack exposed data from employees, executives, films and much more information from Sony Corporation. The U.S. intelligence service accused North Korea as being responsible for the attack, which arouse many suspicious about the veracity of the accusation. This attack was discovered because a malware previously installed, rendered many computers inoperable at the time.

- **Equation Group**: This group was exposed to public in February 2015. It was reported as the most advanced cyber espionage platform known to the time, being operated by a highly sophisticated threat actor involved in multiple computer network exploitation operations, dating back to 2001 and maybe earlier. Its modules use high technological capabilities, strong encryption and before it is fully deployed in the computer victim, it first checks with its command and control system if the target is correct. Versions in multiple systems were found.

- **Trojan. Laziok**: This was recently discovered in 2015 and it is used as a reconnaissance tool that allows attackers to gather information and tailor their attack methods for each compromised computer. This malware was part of a targeted attack campaign against energy companies around the world, with a focus on the Middle East.

- **Barbar**: This malware first appeared in 2009. Reports states that babar is part of cyber espionage campaign named Animal Farm Group or Snowglobe. This malware is able to steal keystrokes, clipboards and listen to Skype conversations among other functions. It is reported by some online sources as being controlled by the France Intelligence Agency. It appears to have been used in actions against Syrian targets

using zero-day exploits hosted on a Syrian Government website. Analysis showed that the authors of this malware had deep knowledge of how some antivirus products worked.

- **Regin**: This malware is a sophisticated toolkit widely reported in 2014. This malware was used in a cyber espionage campaign that target Belgian Belgacom telecommunication provider. The first samples dates back 2003. Sources from the news points its origin to UK and US as controllers of this malware. Infections were found in Russia, India, Mexico, Ireland, Austria, Afghanistan and other countries. This malware was compared to Stuxnet and developed as being a multi-purpose data collection tool.

- **Dragonfly**: Also named Havex, successfully managed to spy strategic organizations. If the affected systems were explored the way they could have been, it could have caused damage to energy supplies in those who were affected. Their targets include the energy sector, major electricity generation firms, petroleum pipeline operators, and energy industry industrial equipment providers. Victims were located in the United States, Spain, France, Italy, Germany and others countries. This cyber operation caused companies to install the malware when downloading software updates for computers running Industrial Control Systems (ICS) equipment.

Germany recently suffered from a target attack in a steel mill facility but the malware or APT did not have its common name reported. The attack manipulated and disrupted control systems to such a point that a blast furnace could not be properly shut down, causing damage to the system. The attackers infiltrated the corporate network using a spear-phishing attack. After that the attackers gained access to the network they explored, they moved to other networks and compromised multiple different systems, including industrial components on the production network of the target.

6.7 Relations of Information Security and Trust and Cyber Security

The way this works sees the connections between cyber security, trust, information security and information assurance is shown in Figure 6.4. Information assurance is considered the bigger part of the schema, but it has strong relations with trust (chapters 2 and 4), information security (5) and cyber security.

In general, if one wants to understand and apply information assurance in a particular organization, it is important to remember that the connections among information security, trust and cyber security should also be understood and detailed. It is not the objective of this work to fully detail all kinds of connections between information security, cyber security and trust because it depends on every organization itself. So, what may be applied to one place may not be used in another. Anyway, this schema (Figure 6.4) shall help in understanding the difficulties that exist in protecting information considering cyber security.

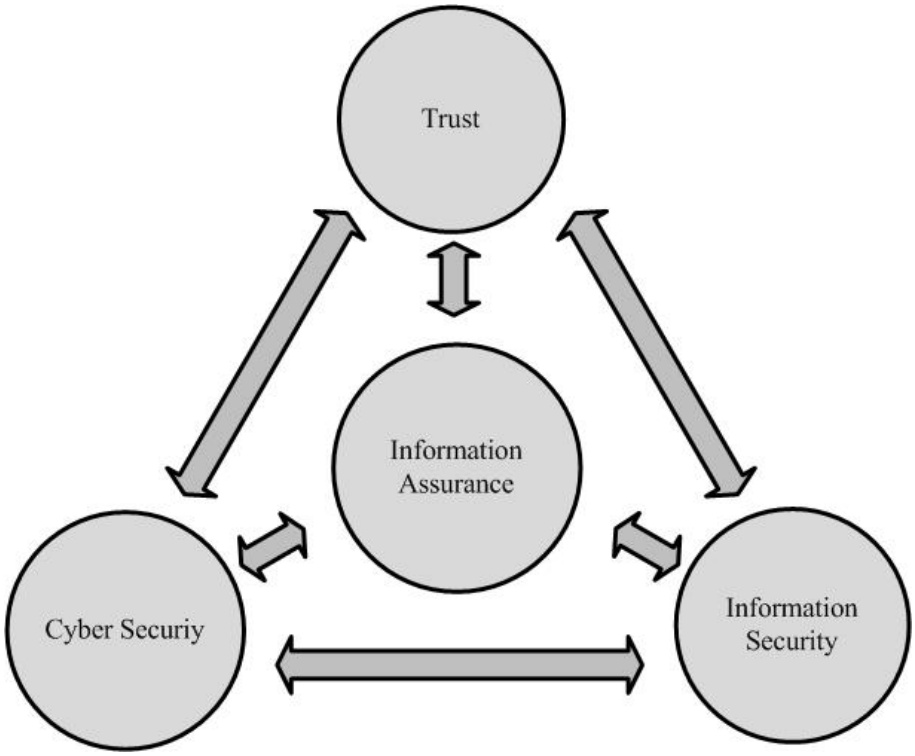


Figure 6.4: Relations in information assurance, information security, trust and cyber security

As seen before in the trust review (chapter 2) and the developed trust model for groups in computational systems (chapter 4), trust in information system is subject to many discussions. Trust is important to ensure secure and reliable communications. The accuracy of trust evaluation depends whether a trust system can function properly under all situations and be capable of handling unfair rating attacks in a satisfactory manner ([WMI+15]).

It is clear that trust management systems are necessary in order to accomplish security related tasks in distributed systems [dOAGVK14]. The management of trust relationships between different entities can be done using different approaches. Considering this view, trust relationship can be manually established by each node in a network but it does not scale when the number of nodes increases. In such cases trust models are used to automatically calculate how much a node can trust other nodes.

Trust has a complex relationship in information security and cyber security. Up to the end of writing this manuscript, one of the main problems of trust yet is how to build trustworthy systems from untrustworthy components. From the perspective of this work, the absence of protocols to exchange reliable trust and reputation information is a problem that needs attention to help build trust in network systems, which would lead to the improvement of computational trust and consequently to the increase of computer security.

6.8 Synthesis of the Chapter

Cyber security gained a lot of attention lately. Protection of cyberspace is vital to many sectors of economics and modern society. Many actors are exploring cyber space gaining profit but also gaining strategic advantages. They perform such actions by mining trust of computation systems and exploring vulnerabilities that most people are not aware of.

In order to promote further discussion about cyber security, information security and trust, this chapter presented a schema where these areas are connected to information assurance. It also discussed current cyber security and trust aspects dealing with cyber security strategies, where Government and enterprises worldwide concentrate efforts on dealing with threats in cyber space.

In particular sections, it discussed and presented some information about exploits with consideration given to its basic development life cycle and an idea of time of the opportunity window it gives attackers in computational resources. Also, this chapter detailed information about APTs and some of its characteristics. As for examples, it collected information of various malware available in Internet.

Chapter 7

Conclusions and Future Works

Trust in computational systems is still an open subject with many research opportunities, just like the information security field. The trust model for distributed systems based on groups of peers for computational system could be developed and some results were reached. In order to try to put trust in a closer relation to information security, this work elaborated an architecture putting information security and trust in a perspective that uses the concept of information security with the possible applications of trust in computational environments.

As stated at the beginning of this research document, the main objective consisted of using trust, reputation and information security in the same perspective. Once this focus was achieved, the research then was divided in two main areas - trust and information security.

In the area of trust, it was possible to develop a model of trust that may be applied to groups in computational systems such as P2P. With further research, it can be adapted to grids, software agents and even cloud computing.

This research considered a group was defined as a collection of entities with particular affinities and capabilities, where all entities used a trust and a reputation value in the system. One of the problem using trust in large scenarios is that in many cases it is necessary to trust the whole system instead of one particular entity. This is the case where most trust and reputation systems are used, considering a 1:1 relationship and not a 1:N or M:N approach.

This work then developed the concept that group trust represents the trust of their particular members and to reach a result in this perspective, a group trust calculation model was proposed, developed and implemented. The developed model is also understood as an extension to support the calculation of trust values of groups of entities. The results show that it is possible to generate and to calculate group trust behavior in distributed system.

As a means of validating the proposed model, this research used a P2P simulation tool. The use of P2P is because it is a complex distributed system, but entities share resources and common goals in the system. The results showed that in groups with a 100 to 200 entities, when 40% of its members, the group is not trustworthy depending on the threshold established. The model was able to identify the change in behavior of the group,

where decisions such as the variation of the changes of the group reaches some value, the whole group can be considered untrustworthy. In such cases the leader may decide to eliminate the peers that mislead the group trust information.

After reaching the first part of the problem, this research developed a trust information security architecture (TISA). It is widely known that information is an important asset to any organization. However securing information is very difficult in terms of preserving confidentially, integrity and availability. The problem arises when privacy, anonymity, resilience, etc. are also considered as important as the CIA triad.

The presented layered trust information security architecture considers that information and security should be seen and understood from different points of view in order to protect it. In order to extend the discussion regarding information security, information representation and treatment elements, operations and support components were integrated to show the various sources of problems and risks that should be considered when dealing with both information and security.

Regarding information security architecture, trust, cyber security, advanced persistent threats and exploits, this research presented a wide review that shows that cyber security management needs a systematic approach to consistently address security in every level, reducing unmanaged risks and improving operational security efficiency.

It is also important to consider that in the perspective of this work, there is no known proven technology or framework in information technology without security issues, such as exploitable flaws, configuration errors or system misuse. All that leads such systems to an untrusted perspective. Inside an environment where more people, computer networks, technology, wearable devices are connected together, due care with information security should be much more than network perimeters, antivirus, intrusion detection system, intrusion prevention systems, etc. Information security is a never ending task. It is a continuous task with cyclic approach that changes constantly.

It is important that trust keeps advancing in cyber environments. It is no longer possible to trust software or hardware without taking careful steps towards information security. Cyber attacks to infrastructure where human lives are at risk and people using APTs and exploits may cause harm to innocents. Such concerns must be treated properly. Considering this perspective, the correct understanding of cyber security is fundamental to increase the security level of critical infrastructure systems to avoid harm to humans and assets.

Basically, to fully trust something that you did not build may not be the correct approach, even though, users believe their systems, networks, databases, and so on. Thus, to trust information system is basically to assert that the system does and operates as required, aside environment problems, human or operators errors and even attacks. But the most important in trust approach is that the system will not do other things it should not, from a security point of view. So to trust information system or a network information system requires more than just assembling parts or components. Rather, due to its characteristics, trust depends on subjective evaluation and requires both context and analysis, which leads to soft security.

Also, the correct approach to dealing with information in a connected world is a wide concern to multiple sectors of the society because the cyber space is also a place where technology is used and developed by highly specialized players, with high capability that explores vulnerabilities to gain strategic benefits in many fields of knowledge. Many countries face daily threats to their information assets and most of them even do not know they are already compromised, with their knowledge being exfiltrated.

Technology will keep changing the way we live and will keep allowing people and organizations to get even more connected. In this highly technologically linked world real threats are being forgotten in many situations. Vulnerabilities will keep coming up and hackers will keep exploring them. Nation-states will keep their activities to gain knowledge over what interests them. That has a lifecycle that will require even more knowledge as technology advances. Information security will keep increasing its importance even more as the cyber space keeps growing.

APTs and exploits will keep appearing. The more research in areas such as deep packet inspection, operational systems, hardware, software, wearable devices, gadgets, and as industries create new dependable technological products, the more APTs and exploits will be used to bypass protections, controls and security measures to perform data exfiltration and provide strategic advantages to those who use them. The way the internet is nowadays, to certainly identify the origin of an attack is highly improbable for those who are the victims, thus new methods of trying to indicate the attribution of such activities are very important.

Considering an overview of how information is represented and treated nowadays in technological environment shows the reason why it is so difficult to guarantee security in all aspects of the information pathway. Trust in computational system is an option to help increase the security in such systems.

The cyberspace is essential to support multiple economy sectors, as it constitutes a source of resources and wealth, as well as representing the projection of power from multiple nations. Due to its characteristics, cyberspace has become a place where the exploitation of information vulnerabilities occurs continuously. Tools for vulnerability exploitation are in the middle of a production and trade process, which, with a variety of actors, form a specialized market, where large amounts of money takes place.

For law and enforcement agencies that fights cybercrime it is very important to have knowledge of hazards such as exploits and APTs, not only to protect information systems, but to perform cybercrime investigation and forensics activities. This points to a situation where strategic planning is a critical requirement to deal with information security and cyber security.

Trust is still a difficult subject in terms of information security. It has lines of understanding related to subjective evaluation. Finally, considering the whole research, the proposed objective and its consequent specific objectives of the first part of the problem, it was possible to review both the state of the art of trust and reputation models and trust applied to distributed systems, basically seen in chapter 2 and 3.

Chapter 4 shows what was developed as the proposed group trust model applied to computational systems and the results reached by the developed model.

Regarding the second part of the problem, chapter 3 provided a review of information security and cyber security, both used as fundamentals in the discussion provided in chapter 5, where the development of information security architecture and its relation to trust was presented and discussed. Chapter 6 described cyber security and its relations to trust regarding threats in information security field.

7.1 Future Works

Considering that both trust and information security are not an end by itself; this research considers that both are a means to achieve an end. It is important to touch on some future works that can be conducted from where this thesis reached its results.

Considering the trust perspective, it is important that the problem of trust leadership and trust consensus should be better studied in order to create a better way of dealing with leaders in groups in a distributed manner.

It is also important to define a trust protocol as a platform to support trust based communications. There are not many known trust protocols developed or even implemented in computational systems to exchange trust and reputation information over the Internet. Trust protocols for exchanging information in networks is seen as a research area that can be studied in depth.

Using the concept of group trust, the proposed model in this paper can be evolved to be applied in bigger and more complex distributed systems architectures. The proposed model can be adapted to be used in software agents, grid platforms or cloud environments.

There is also the need and the intention to extend the trust functions and further explain the relations between the sections and all layers of the proposed architecture (TISA), so more details regarding trust in the architecture could be better explained. It is also another goal to detail how the pieces of the architecture, such as people, technology and processes should be connected and guided using information security polices, continuous monitoring and so on, with the main objective of creating an information security and trust methodology using TISA as a starting point of thinking.

At last, further research is important to develop new manners of dealing with computational trust in cyber space, which could be useful to provide advances in information security strategies and more efficient solutions to keep information protected.

Bibliography

[ABB15] Joseph Kwame Adjei, Colin Blackman, and Colin Blackman. Explaining the Role of Trust in Cloud Computing Services. *info*, 17(1):54–67, 2015.

[AD02] Christopher J. Alberts and Audrey Dorofee. *Managing Information Security Risks: The OCTAVE Approach*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[ALR05] Ali Shaikh Ali, Simone Ludwig, and Omer F. Ran. A Cognitive Trust-Based Approach for Web Service Discovery and Selection. In *Proceedings of the Third IEEE European Conference on Web Services (ECOWS)*, Växjö, Sweden, November 2005.

[AP12] Salvatore Aurigemma and Raymond Panko. A Composite Framework for Behavioral Compliance with Information Security Policies. In *Proceedings of the 45th Hawaii International Conference on System Science (HICSS)*, pages 3248–3257, Wailea, Maui, Hawaii, January 2012.

[ASS07] Rose-Mharie Ahlfeldt, Paolo Spagnoletti, and Guttorm Sindre. Improving the Information Security Model by using TFI. In *Proceedings of the IFIP TC-11 22nd International Information Security Conference (SEC)*, pages 73–84, Sandton, South Africa, May 2007.

[BAG15] Noam Ben-Asher and Cleotilde Gonzalez. Effects of Cyber Security Knowledge on Attack Detection. *Computers in Human Behavior*, 48:51–61, July 2015.

[BATG15] Ghusoon Salim Basheer, Mohd Sharifuddin Ahmad, Alicia Y. C. Tang, and Sabine Graf. Certainty, Trust and Evidence: Towards an Integrative Model of Confidence in Multi-Agent Systems. *Computers in Human Behavior*, 45:307–315, April 2015.

[BBK94] Thomas Beth, Malte Borcherding, and Birgit Klein. *Valuation of Trust in Open Networks*. Springer, 1994.

[BD12] Leyla Bilge and Tudor Dumitras. Before we Knew it: An Empirical Study of Zero-Day Attacks in the Real World. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 833–844, Raleigh, North Carolina, USA, October 2012.

[Bib77] Kenneth J. Biba. Integrity Considerations for Secure Computer Systems. Technical Report, DTIC Document, 1977.

[BJ13] HMN Dilum Bandara and Anura P. Jayasumana. Collaborative Applications over Peer-to-Peer Systems–Challenges and Solutions. *Peer-to-Peer Networking and Applications*, 6(3):257–276, September 2013.

[BL73] D. Elliott Bell and Leonard J. LaPadula. Secure Computer Systems: Mathematical Foundations. Technical Report, DTIC Document, 1973.

[BL04] Eric Byres and Justin Lowe. The Myths and Facts Behind Cyber Security Risks for Industrial Control Systems. In *Proceedings of the VDE Kongress*, volume 116, pages 213–218, Berlin, Germany, October 2004.

[BMG02] Bob Blakley, Ellen McDermott, and Dan Geer. Information Security is Information Risk Management. In *Proceedings of the Workshop on New Security Paradigms (NSPW)*, pages 97–104, Cioudcroll, New Mexico, USA, September 2002.

[BNO+05] Jim Basney, Wolfgang Nejdl, Daniel Olmedilla, Von Welch, and Marianne Winslett. Negotiating Trust on the Grid. In *Proceedings of the Seminar on Semantic Grid: The Convergence of Technologies*, Hannover, Germany, November 2005.

[Bro15] Chris Brook. All Major Browsers Fall at Pwn2Own Day 2. https://threatpost.com/all-major-browsers-fall-at-pwn2own-day-2/111731, 2015.

[Bur83] James H. Burrows. Guideline for Computer Security Certification and Accreditation, 1983.

[Chu03] Heting Chu. *Information Representation and Retrieval in the Digital Age*. Information Today, Inc., 2003.

[CJL+13] Robert E. Crossler, Allen C. Johnston, Paul Benjamin Lowry, Qing Hu, Merrill Warkentin, and Richard Baskerville. Future Directions for Behavioral Information Security Research. *Computers & Security*, 32:90–101, February 2013.

[Con13] Jorge L. Contreras. Developing a Framework to Improve Critical Infrastructure Cybersecurity . Technical Report SSRN 2248658, Response to NIST Request for Information Docket No. 130208119-3119-01, April 2013.

[D+11] K. Dempsey et al. Information Security Continuous Monitoring (ISCM) for Federal Systems and Organisations. *NIST Special Publication*, pages 800–137, September 2011.

[Dar14] Darpa Information Innovation Office. Memex (Domain-Specific Search). http://www.darpa.mil/newsevents/releases/2014/02/09.aspx, 2014.

[Das00] Partha Dasgupta. Trust as a Commodity. *Trust: Making and Breaking Cooperative Relations*, 4:49–72, March 2000.

[DCdSJdOAdM12] Edna Dias Canedo, Rafael Timóteo de Sousa Junior, Robson de Oliveira Albuquerque, and Fabio Lucio Lopes de Mendonca. File Exchange in a Private Cloud supported by a Trust Model. In *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 89–96, Sanya, China, October 2012.

[Dep07] Department of Communications and Information Technology and the Arts and the Trusted Information Sharing Network. Secure Your Information: Information Security Principles for Enterprise Architecture. Technical Report, TISN Australia, July 2007.

[Die08] Tim Dierks. The Transport Layer Security (TLS) Protocol version 1.2. RFC 5246, The Internet Engineering Task Force (IETF), August 2008.

[Dis13] Georg Disterer. ISO/IEC 27000, 27001 and 27002 for Information Security Management. *Journal of Information Security*, 4(2), April 2013.

[DJGKS+15] H. Devare Jayvant, D. Gavade Kiran, B. E. Student, M. Domale Madhuri, D. Ghanwat Rohini, and R. Hivrale Sunil. Multi-Cloud Security For Hadoop Data. *International Journal of Informative & Futuristic Research*, 2(7):2210–2216, March 2015.

[DL07] Whitfield Diffie and Susan Eva Landau. *Privacy on the Line: The Politics of Wiretapping and Encryption*. MIT Press, 2007.

[DLG15] Haiyang Ding, Xiguang Li, and Changqing Gong. Trust Model Research in Cloud Computing Environment. In *Proceedings of the International Symposium on Computers & Informatics (ICACCI)*, Beijing, China, January 2015.

[dOABGV14] Robson de Oliveira Albuquerque, Fábio Buiati, and Luis Javier García Villalba. Arquitectura de Seguridad Multinivel: Una Guía para las Organizaciones Modernas. In *Actas del XIII Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, pages 233–236, Alicante, España, Septiembre 2014.

[dOAdSJBAGV05] Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Jr., Cláudia Jacy Barenco Abbas, and Luis Javier García Villalba. MANET Auto Configuration with Distributed Certification Authority Models Considering Routing Protocols Use. In *Proceedings of the 3rd International Workshop on Security in Information Systems (WOSIS)*, pages 57–66, Miami, USA, May 2005.

[dOAdSJdS+04] Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Jr., Tamer Américo da Silva, Ricardo S. Puttini, Cláudia Jacy Barenco Abbas, and Luis Javier García Villalba. Load Balancing and Survivability for Network Services Based on Intelligent Agents. In *Proceedings of the International Conference on Computational Science and Its Applications (ICCSA)*, pages 868–881, Assisi, Italy, May 2004.

[dOAGV12] Robson de Oliveira Albuquerque and Luis Javier García Villalba. Group Trust Model. In *Booklet of the 10th International Conference on High Performance Computing & Simulation (HPCS)*, Madrid, Spain, July 2012.

[dOAGVdSJ08] Robson de Oliveira Albuquerque, Luis Javier García Villalba, and Rafael Timóteo de Sousa Jr. Enhancing an Integer Challenge-Response Protocol. In *Proceedings of the International Conference on Computational Science and its Applications (ICCSA)*, pages 526–540, Perugia, Italy, July 2008.

[dOAGVK14] Robson de Oliveira Albuquerque, Luis Javier García Villalba, and Tai-Hoon Kim. GTrust: Group Extension for Trust Models in Distributed Systems. *International Journal of Distributed Sensor Networks*, Article ID 872842, February 2014.

[dOAGVSO+14] Robson de Oliveira Albuquerque, Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Fábio Mesquita Buiati, and Tai-Hoon Kim. A Layered Trust

Information Security Architecture. *Sensors*, 14(12):22754–22772, December 2014.

[dOAMBGV+06] Robson de Oliveira Albuquerque, Fabio Mesquita Buiati, Luis Javier García Villalba, Daniel Silva Almendra, Leonardo Lobo Pulcineli, Rafael Timóteo de Sousa Jr., and Cláudia Jacy Barenco Abbas. Proposal of a System for Searching and Indexing Heterogeneous Vulnerabilities Databases. In *Proceedings of the International Conference on Computational Science and its Applications (ICCSA)*, pages 819–828, Glasgow, UK, May 2006.

[dOASALP+05] Robson de Oliveira Albuquerque, Daniel Silva Almendra, Leonardo Lobo Pulcineli, Rafael Timóteo de Sousa Jr., Cláudia Jacy Barenco Abbas, and Luis Javier García Villalba. SisBrAV - Brazilian Vulnerability Alert System. In *Proceedings of the 3rd International Workshop on Security in Information Systems (WOSIS)*, pages 67–76, Miami, USA, May 2005.

[dOAVRTGdD11] Robson de Oliveira Albuquerque, Luis Javier García Villalba, Osmar Ribeiro Torres, and Flavio Elias Gomes de Deus. Virtualization with Automated Services Catalog for Providing Integrated Information Technology Infrastructure. In *Proceedings of the 8th International Conference Autonomic and Trusted Computing (ATC)*, pages 75–91, Banff, Canada, September 2011.

[Dou02] John R. Douceur. The Sybil Attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 1–21, Cambridge, MA, USA, March 2002.

[dSJdOAMBGV06] Rafael Timóteo de Sousa Jr., Robson de Oliveira Albuquerque, Fabio Mesquita Buiati, and Luis Javier García Villalba. An Overview of Open Issues and Preliminary Solutions Regarding Security in Ad-Hoc Networks. In *Booklet of the Workshop on Security and Privacy in Mobile and Wireless Networking (in conjunction with IFIP Networking)*, Coimbra, Portugal, May 2006.

[Dun00] Paul Dunn. The Importance of Consistency in Establishing Cognitive-Based Trust: A Laboratory Experiment. *Teaching Business Ethics*, 4(3):285–306, August 2000.

[DWJZ04] Wen Dou, Huai-Min Wang, Yan Jia, and Peng Zou. A Recommendation-Based Peer-to-Peer Trust Model. *Journal of Software*, 15(4):571–583, April 2004.

[Elh11] Nelson Elhage. Virtunoid: A KVM Guest − > Host Privilege Escalation Exploit, 2011.

[Eur15] European Union Agency for Network and Information Security. National Cyber Security Strategies in the World. https://www.enisa.europa.eu/activities/Resilience-and-CIIP/national-cyber-security-strategies-ncsss/national-cyber-security-strategies-in-the-world, 2015.

[FC01] Rino Falcone and Cristiano Castelfranchi. *Trust and Deception in Virtual Societies*, chapter Social Trust: A Cognitive Approach, pages 55–90. Springer Netherlands, 2001.

[FHAS14] J. E. Fadul, K. M. Hopkinson, T. R. Andel, and C. A. Sheffield. A Trust-Management Toolkit for Smart-Grid Protection Systems. *IEEE Transactions on Power Delivery*, 29(4):1768–1779, August 2014.

[FKK11] Alan Freier, Philip Karlton, and Paul Kocher. The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101, The Internet Engineering Task Force (IETF), August 2011.

[Fre13] Stefan Frei. The Known Unknowns: Empirical Analysis of Publicly Unknown Vulnerabilities. Technical Report, NSS Labs, 2013.

[FSSF15] Ivo Friedberg, Florian Skopik, Giuseppe Settanni, and Roman Fiedler. Combating Advanced Persistent Threats: From Network Event Correlation to Incident Detection. *Computers & Security*, 48:35–57, February 2015.

[Gam00] Diego Gambetta. *Trust: Making and Breaking Cooperative Relations*, chapter Can we Trust Trust?, pages 213–237. Basil Blackwell, 2000.

[Gar13] Gartner Press Release. Gartner Says Cloud-Based Security Services Market to Reach 2.1 Billion in 2013. Technical Report 2616115, Gartner, October 2013.

[GBS14] Ekta Gandotra, Divya Bansal, and Sanjeev Sofat. Computational Techniques for Predicting Cyber Threats. In *Proceedings of the International Conference on Intelligent Computing, Communication and Devices (ICCD)*, pages 247–253, August 2014.

[GC14] Avijit Gayen and Joydeep Chandra. Role of Trust in Evolution of Scientific Collaboration Networks. In *Proceedings of the Seventh International Conference on Social Computing (SocialCom)*, Beijing, China, August 2014.

[Gee14] Dan Geer. Cybersecurity as Realpolitik. https://www.blackhat.com/us-14/video/cybersecurity-as-realpolitik.html, 2014.

[GMP13] Glenn Greenwald, Ewen MacAskill, and Laura Poitras. Edward Snowden: The Whistleblower Behind the NSA Surveillance Revelations. *The Guardian*, 9, 2013.

[Gol14] Steve Gold. APTs: Not as Advanced as you Might Think. http://www.scmagazineuk.com/apts-not-as-advanced-as-you-might-think/article/345953/, May 2014.

[Gon14] Max Goncharov. Russian Underground Revisited. Technical Report, Trend Micro, 2014.

[Gre12] Andy Greenberg. Shopping for Zero-Days: A Price list for Hackers' Secret Software Exploits. *Forbes Magazine*, March 2012.

[Gre15] Andy Greenberg. New Dark-Web Market is Selling Zero-Day Exploits to Hackers. http://www.wired.com/2015/04/therealdeal-zero-day-exploits/, 2015.

[Har14] Sean L. Harrington. Cyber Security Active Defense: Playing with Fire or Sound Risk Management? *Richmond Journal of Law & Technology*, 20:12–13, September 2014.

[HBC15] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. Reputation Systems: A Survey and Taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, January 2015.

[HCCT03] Kwo-Shing Hong, Yen-Ping Chi, Louis R Chao, and Jih-Hsing Tang. An Integrated System Theory of Information Security Management. *Information Management & Computer Security*, 11(5):243–248, May 2003.

[Hel15] Help Net Security. Attackers use Deceptive Tactics to Dominate Corporate Networks. http://www.net-security.org/secworld.php?id=18208, 2015.

[HJS06] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. An Integrated Trust and Reputation Model for Open Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, September 2006.

[HP 12] HP Research. Cybercrime Costs Rise Nearly 40 Percent, Attack Frequency Doubles. Technical Report, Hewlett-Packard, 2012.

[HRMV14] Sheikh Mahbub Habib, Sebastian Ries, Max Mühlhäuser, and Prabhu Varikkattu. Towards a Trust Management System for Cloud Computing Marketplaces: Using Caiq as a Trust Information Source. *Security and Communication Networks*, 7(11):2185–2200, April 2014.

[HS04] Dominic Hughes and Vitaly Shmatikov. Information Hiding, Anonymity and Privacy: A Modular Approach. *Journal of Computer Security*, 12(1):3–36, January 2004.

[HTK13] Ryan Hand, Michael Ton, and Eric Keller. Active Security. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks (HotNets-IV)*, page 17, College Park, Maryland, USA, November 2013.

[IBM04] IBM Grid Computing. What is Grid computing. http://www-1.ibm.com/grid/about_grid/what_is.shtml, 2004.

[Inf09] Information Systems Audit and Control Association. An Introduction to the Business Model for Information Security. Technical Report, ISACA, August 2009.

[JG05] Devon Johnson and Kent Grayson. Cognitive and Affective Trust in Service Relationships. *Journal of Business research*, 58(4):500–507, April 2005.

[JIB07] Audun Jøsang, Roslan Ismail, and Colin Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2):618–644, March 2007.

[JKJA14] Masoumeh Rezaei Jam, Leili Mohammad Khanli, Morteza Sargolzaei Javan, and Mohammad Kazem Akbari. A Survey on Security of Hadoop. In *Proceedings of the 4th International eConference on Computer and Knowledge Engineering (ICCKE)*, pages 716–721, Mashhad, Iran, October 2014.

[JSJM05] Zhu Junmao, Yang Shoubao, Fan Jianping, and Chen Mingyu. A grid & P2P Trust Model based on Recommendation Evidence Reasoning. *Journal of Computer Research and Development*, 42(5):797–803, 2005.

[JXT13a] JXTA Board. The Language and Platform Independent Protocol for P2P Networking. https://jxta.kenai.com/, 2013.

[Jxt13b] Jxta Implementation. The Java Implementation of the JXTA Protocols. https://jxse.kenai.com/, 2013.

[JYJ10] Gu-Heon Jeong, Dong-Wook Yi, and Seung-Ryul Jeong. The Effect of Composition and Security Activities for Information Security Architecture on Information Asset Protection and Organizational Performance. *The KIPS Transactions: PartD*, 17(3):223–232, 2010.

[Kha13] Safwan Mahmud Khan. Decentralizing Trust: New Security Paradigms for Cloud Computing. PhD. Thesis, University of Texas, 2013.

[Kie14] Rolf Kiefhaber. Calculating and Aggregating Direct Trust and Reputation in Organic Computing Systems. PhD. Thesis, University of Augsburg, 2014.

[Kiz15] Cagri Kizak. Reputation Management: How to Deal with Online Reputation Threats? Technical Report, University of Twente, 2015.

[Lam01] Pradip Lamsal. Understanding Trust and Security. Technical Report, Department of Computer Science, University of Helsinki, Finland, 2001.

[LC12] Chi-Chun Lo and Wan-Jia Chen. A Hybrid Information Security Risk Assessment Procedure Considering Interdependences between Controls. *Expert Systems with Applications*, 39(1):247–257, January 2012.

[LKYL14] Lixiang Li, Jürgen Kurths, Yixian Yang, and Guole Liu. Prevention and Trust Evaluation Scheme Based on Interpersonal Relationships for Large-Scale Peer-To-Peer Networks. *Mathematical Problems in Engineering*, 2014, September 2014.

[Men15] Joseph Menn. Politics Intrude as Cybersecurity Firms Hunt Foreign Spies. http://mobile.reuters.com/article/idUSKBN0M809N20150312?irpc=932, 2015.

[MFGBH14] Francisco Moyano, Carmen Fernández-Gago, Kristian Beckers, and Maritta Heisel. Enhancing Problem Frames with Trust and Reputation for Analyzing Smart Grid Security Requirements. In *Proceedings of the Second International Workshop on Smart Grid Security (SmartGridSec)*, pages 166–180, Atlanta, GA, USA, May 2014.

[MHAH08] Janne Merete Hagen, Eirik Albrechtsen, and Jan Hovden. Implementation and Effectiveness of Organizational Information Security Measures. *Information Management & Computer Security*, 16(4):377–397, 2008.

[Mil15] Greg Miller. CIA Plans Major Reorganization and a Focus on Digital Espionage. http://www.washingtonpost.com/world/national-security/cia-plans-major-reorganization-and-a-focus-on-digital-espionage/2015/03/06/87e94a1e-c2aa-11e4-9ec2-b418f57a4a99_story.html, 2015.

[Mor98] Patricia Morreale. Agents on the Move [Mobile Software Agents]. *Spectrum, IEEE*, 35(4):34–41, April 1998.

[MPR+14] Fabrizio Messina, Giuseppe Pappalardo, D. Rosaci, C. Santoro, and Giuseppe M. L. Sarné. A Trust Model for Competitive Cloud Federations. In *Proceedings of the Eighth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pages 469–474, Birmingham, UK, July 2014.

[MSGGGS+12] Iñaki Martínez-Sarriegui, Fernando García-García, Gema García-Sáez, M. Elena Hernando, and Michael Luck. TRHIOS: Trust and Reputation in Hierarchical and Quality-Oriented Societies. In *Proceedings of the 7th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–4, Madrid, Spain, June 2012.

[NB14] I. S. Nithya and Kannan Balasubramanian. Trust Metrics Evaluation for Peer-to-Peer Systems. *International Journal of Advanced Research in Computer Science & Technology*, 2(1), January 2014.

[NJ15] Sadegh Dorri Nogoorani and Rasool Jalili. TIRIAC: A Trust-Driven Risk-Aware Access Control Framework for Grid Environments. *Future Generation Computer Systems*, March 2015.

[NSSV13] Richi Nayak, Pierre Senellart, Fabian M Suchanek, and Aparna S Varde. Discovering Interesting Information with Advances in Web Technology. *ACM SIGKDD Explorations Newsletter*, 14(2):63–81, December 2013.

[Nwa96] Hyacinth S. Nwana. Software Agents: An Overview. *The knowledge engineering review*, 11(03):205–244, October 1996.

[Pat07] Jigar Patel. A Trust and Reputation Model for Agent-Based Virtual Organisations. PhD. Thesis, University of Southampton, January 2007.

[Pau14] Paul English. Basic Trust Models, 2014.

[PCK11] Wolter Pieters and Lizzie Coles-Kemp. Reducing Normative Conflicts in Information Security. In *Proceedings of the Workshop on New Security Paradigms (NSPW)*, pages 11–24, Marin County, CA, USA, September 2011.

[PD05] Ana Marilza Pernas and Mario Dantas. Grid Computing Environment using Ontology based Service. In *Proceedings of the 5th International Conference on Computational Science (ICCS)*, pages 858–861, Atlanta, GA, USA, May 2005.

[Pel13] Thomas R. Peltier. *Information Security Fundamentals*. CRC Press, 2013.

[PF04] Elvis Papalilo and Bernd Freisleben. Towards a Flexible Trust Model for Grid Environments. In *Proceedings of the First International Conference on Grid Services Engineering and Management(GSEM)*, pages 94–106, Erfurt, Germany, September 2004.

[POK13] Reza Parvizi, Fereshteh Oghbaei, and Seyyed Raouf Khayami. Using COBIT and ITIL Frameworks to Establish the Alignment of Business and IT Organizations as One of the Critical Success Factors in ERP Implementation. In *Proceedings of the 5th Conference on Information and Knowledge Technology (IKT)*, pages 274–278, Shiraz, Iran, May 2013.

[PSM13] Isaac Pinyol and Jordi Sabater-Mir. Computational Trust and Reputation Models for Open Multi-Agent Systems: A Review. *Artificial Intelligence Review*, 40(1):1–25, June 2013.

[PVS04] Shaun Posthumus and Rossouw Von Solms. A Framework for the Governance of Information Security. *Computers & Security*, 23(8):638–646, December 2004.

[QB14] Chenhao Qu and Rajkumar Buyya. A Cloud Trust Evaluation System using Hierarchical Fuzzy Inference System for Service Selection. In *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 850–857, Victoria, BC, May 2014.

[RGP06] Martin Rehak, Miloš Gregor, and Michal Pěchouček. Multidimensional Context Representations for Situational Trust. In *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pages 315–320, Prague, Czech Republic, June 2006.

[Ric13] Jeffrey T. Richelson. The Snowden Affair. *Washington Post*, 2013.

[RLO15] Nelson Rodrigues, Paulo Leitão, and Eugénio Oliveira. *Service Orientation in Holonic and Multi-agent Manufacturing*, volume 594, chapter Self-Interested Service-Oriented Agents Based on Trust and QoS for Dynamic Reconfiguration, pages 209–218. Springer International Publishing, 2015.

[RRM09] Negin Razavi, Amir Masoud Rahmani, and Mehran Mohsenzadeh. A Context-Based Trust Management Model for Pervasive Computing Systems. *International Journal of Computer Science and Information Security (IJCSIS)*, 6(1):137–142, November 2009.

[RSG12] Domenico Rosaci, Giuseppe M.L. Sarné, and Salvatore Garruzzo. Integrating Trust Measures in Multiagent Systems. *International Journal of Intelligent Systems*, 27(1):1–15, November 2012.

[SAT11] Heru Susanto, Mohammad Nabil Almunawar, and Yong Chee Tuan. Information Security Management System Standards: A Comparative Study of the Big Five. *International Journal of Electrical & Computer Sciences IJECS-IJENS*, 11(05), October 2011.

[Sch06] Carsten D. Schultz. A Trust Framework Model for Situational Contexts. In *Proceedings of the International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, page 50, Markham, Ontario, Canada, November 2006.

[SCSW11] Dawei Sun, Guiran Chang, Lina Sun, and Xingwei Wang. Surveying and Analyzing Security, Privacy and Trust Issues in Cloud Computing Environments. *Procedia Engineering*, 15:2852–2856, 2011.

[SD15] Mark Seaborn and Thomas Dullien. Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges. http://googleprojectzero.blogspot.com.br/2015/03/exploiting-dram-rowhammer-bug-to-gain.html, 2015.

[SHL15] C. M. Shipman, K. M. Hopkinson, and J. Lopez. Con-Resistant Trust for Improved Reliability in a Smart-Grid Special Protection System. *IEEE Transactions on Power Delivery*, 30(1):455–462, Febryary 2015.

[SK15] Manpreet Kaur Shergill and Harjot Kaur. Survey of Computational Trust and Reputation Models in Virtual Societies. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(4), April 2015.

[SM13] Sugandh Shah and B. M. Mehtre. A Modern Approach to Cyber Security Analysis using Vulnerability Assessment and Penetration Testing. *International Journal of Electronics Communication and Computer Engineering*, 4(6), November 2013.

[SOZL13] Yosra Ben Saied, Alexis Olivereau, Djamal Zeghlache, and Maryline Laurent. Trust Management System Design for the Internet of Things: A Context-Aware and Multi-Service Approach. *Computers & Security*, 39:351–365, November 2013.

[SR13] Saurabh Srivastava and Himanshi Raperia. Various Grid Productions and Comparison of ORACLE and IBM Grid. *International Journal of Advance Innovations, Thoughts & Ideas*, 2(3):1, 2013.

[SS02] Jordi Sabater and Carles Sierra. Reputation and Social Network Analysis in Multi-Agent Systems. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 475–482, Bologna, Italy, July 2002.

[SS05] Jordi Sabater and Carles Sierra. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24(1):33–60, September 2005.

[SS14] Jagpreet Sidhu and Sarbjeet Singh. Compliance based Trustworthiness Calculation Mechanism in Cloud Environment. *Procedia Computer Science*, 37:439–446, September 2014.

[SS15] Rizwana Shaikh and M. Sasikumar. Trust Model for Measuring Security Strength of Cloud Computing Service. *Procedia Computer Science*, 45:380–389, March 2015.

[SSY14] Nitin Kumar Saini, Vikas Kumar Sihag, and Ramesh Chand Yadav. A Reactive Approach for Detection of Collusion Attacks in P2P Trust and Reputation Systems. In *Proceedings of the IEEE International Advance Computing Conference (IACC)*, pages 312–317, Gurgaon, Haryana, India, February 2014.

[ST04] Girish Suryanarayana and Richard N. Taylor. A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications. Technical Report UCI-ISR-04-6, University of California, Irvine, 2004.

[Sta11] Mark Stamp. *Information Security: Principles and Practice*. John Wiley & Sons, 2011.

[Ste94] Marsh Stephen. Formalising Trust as a Computational Concept. PhD. Thesis, University of Stirling, Scotland, UK, April 1994.

[Sym14] Symantec Labs. 2014 Internet Security Threat Report. Technical Report, Symantec, April 2014.

[Sza15] Gabor Szappanos. Exploit This: Evaluating the Exploit Skills of Malware Groups. Technical Report, SophosLabs, January 2015.

[Tad03] Steven Tadelis. Firm Reputation with Hidden Information. *Economic Theory*, 21(2-3):635–651, March 2003.

[Tan14] Chang Long Tang. Establish a Dynamic Business Driven Integrative Information Security Architecture. *Applied Mechanics and Materials*, 513:1309–1315, February 2014.

[TAS+10] André Teixeira, Saurabh Amin, Henrik Sandberg, Karl H Johansson, and Shankar S Sastry. Cyber Security Analysis of State Estimators in Electric Power Systems. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pages 5991–5998, December 2010.

[The13] The National Institute of Science and Technology. Developing a Framework to Improve Critical Infrastructure Cybersecurity. Technical Report, NIST, August 2013.

[Tie14] Tiedata. What are Web Based Exploits. http://www.tiedata.com/webexploits.asp, 2014.

[TLRJ12] W. T. Luke Teacy, Michael Luck, Alex Rogers, and Nicholas R Jennings. An Efficient and Versatile Approach to Trust and Reputation using Hierarchical Bayesian Modelling. *Artificial Intelligence*, 193:149–185, December 2012.

[TMP15] Ginés Dólera Tormo, Félix Gómez Mármol, and Gregorio Martínez Pérez. Dynamic and Flexible Selection of a Reputation Mechanism for Heterogeneous Environments. *Future Generation Computer Systems*, 49:113–124, August 2015.

[TWHV05] Huu Tran, Paul Watters, Michael Hitchens, and null Vijay Varadharajan. Trust and Authorization in the Grid: A Recommendation Model. In *Proceedings of the International Conference on Pervasive Services (PERSER)*, pages 433–436, Santorini, Greece, July 2005.

[TYZL14] Chunqi Tian, Baijian Yang, Jidong Zhong, and Xiaojian Liu. Trust-Based Incentive Mechanism to Motivate Cooperation in Hybrid P2P Networks. *Computer Networks*, 73:244–255, November 2014.

[TZWC08] Chunqi Tian, Shi-Hong Zou, Wen-Dong Wang, and Shi-Duan Cheng. A New Trust Model based on Recommendation Evidence for P2P Networks. *Chinese Journal of Computers -Chinese Edition-*, 31(2):270, 2008.

[USB14] S. Udhaya Shree and Saleem Basha. An Exhaustive Survey of Trust Models in P2P Network. *arXiv*, arXiv:1411.3294, October 2014.

[Ver14] Verizon Enterprise. 2014 Data Breach Investigations Report. Technical Report, Verizon, 2014.

[VO14] Rob Van Os. Comparing Security Architectures: Defining and Testing a Model for Evaluating and Categorizing Security Architecture Frameworks. Master's Thesis, Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering, Sweden, 2014.

[VPZ14] Sokratis Vavilis, Milan Petković, and Nicola Zannone. A Reference Model for Reputation Systems. *Decision Support Systems*, 61:147–154, May 2014.

[Wis08] Thomas Wiseman. Reputation and Impermanent Types. *Games and Economic Behavior*, 62(1):190–210, January 2008.

[WLM+02] Xinsheng Wang, Peng Liang, Huidong Ma, Dan Xing, and Baozong Wang. A P2P Trust Model Based on Multi-Dimensional Trust Evaluation. In *Proceedings of the International Conference on Life System Modeling and Simulation (LSMS)*, pages 347–356, Shanghai, China, September 2002.

[WM13] Michael Whitman and Herbert Mattord. *Management of Information Security*. Cengage Learning, 2013.

[WMI+15] Dongxia Wang, Tim Muller, Athirai A. Irissappane, Jie Zhang, and Yang Liu. Using Information Theory to Improve the Robustness of Trust Systems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 791–799, Istanbul, Turkey, May 2015.

[Woj14] Rafal Wojtczuk. Poacher Turned Gamekeeper: Lessons Learned from Eight Years of Breaking Hypervisors. http://www.bromium.com/sites/default/files/wp-bromium-breaking-hypervisors-wojtczuk.pdf, 2014.

[WWLY06] Li Wang, Wenli Wu, YingJie Li, and XueLi Yu. Content-Aware Trust Statement for Semantic Grid. In *Proceedings of the Second International Conference on Semantics, Knowledge and Grid (SKG)*, pages 95–95, Guilin, China, November 2006.

[Yan14] Hao Yan. Research of Trust Model based on Interest Group and Similarity Recommendation. *Open Automation and Control Systems Journal*, 6:754–759, December 2014.

[YLDZ14] Yun Ye, Wei-min Lin, Song Deng, and Tao Zhang. A Practical Solution to the Information Security Risk Evaluation Problems in Power Systems. In *Proceedings of the International Conference on Future Computer and Communication Engineering (ICFCCE)*, Tianjin, China, March 2014.

[Yor15] Amit Yoran. Escaping Security's Dark Ages. In *Proceedings of the RSA Conference*, Moscone Center, San Francisco, CA, USA, April 2015.

[ZM00] Giorgos Zacharia and Pattie Maes. Trust Management through Reputation Mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.

[ZT15] Eva Zupancic and Denis Trcek. QADE: A Novel Trust and Reputation Model for Handling False Trust Values in E-Commerce Environments with Subjectivity Consideration. *Technological and Economic Development of Economy*, pages 1–30, June 2015.

Part II

Papers Related to This Thesis

Appendix A

List of Papers

1. Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Jr., Tamer Américo da Silva, Ricardo S. Puttini, Cláudia Jacy Barenco Abbas, Luis Javier García Villalba: Load Balancing and Survivability for Network Services based on Intelligent Agents. Proceedings of the International Conference on Computational Science and Its Applications (ICCSA 2004), Assisi, Perugia, Italy, May 14 – 17, 2004. Lecture Notes in Computer Science, Volume 3043, pages 868–881, 2004.

2. Robson de Oliveira Albuquerque, Maíra Hanashiro, Rafael Timóteo de Sousa Jr., Cláudia Jacy Barenco Abbas, Luis Javier García Villalba: MANET Auto Configuration with Distributed Certification Authority Models Considering Routing Protocols Use. Proceedings of the 3rd International Workshop on Security in Information Systems (WOSIS 2005), pages 57–66, Miami, USA, May 24 – 25, 2005.

3. Robson de Oliveira Albuquerque, Daniel Silva Almendra, Leonardo Lobo Pulcineli, Rafael Timóteo de Sousa Jr., Cláudia Jacy Barenco Abbas, Luis Javier García Villalba: SisBrAV - Brazilian Vulnerability Alert System. Proceedings of the 3rd International Workshop on Security in Information Systems (WOSIS 2005), pages 67–76, Miami, USA, May 24 – 25, 2005.

4. Robson de Oliveira Albuquerque, Fábio Mesquita Buiati, Luis Javier García Villalba, Daniel Silva Almendra, Leonardo Lobo Pulcineli, Rafael Timóteo de Sousa Jr., Cláudia Jacy Barenco Abbas: Proposal of a System for Searching and Indexing Heterogeneous Vulnerabilities Databases. Proceedings of the International Conference on Computational Science and its Applications (ICCSA 2006), Glasgow, UK, May 8 – 11, 2006. Lecture Notes in Computer Science, Volume 3984 pages 819–828, 2006.

5. Robson de Oliveira Albuquerque, Luis Javier García Villalba, Rafael Timóteo de Sousa Júnior: Enhancing an Integer Challenge-Response Protocol. Proceedings of the International Conference on Computational Science and its Applications, Perugia, Italy, June 30 – July 3, 2008. Lecture Notes in Computer Science, Volume 5073, pages 526–540, 2008.

6. Robson de Oliveira Albuquerque, Luis Javier García Villalba, Osmar Ribeiro Torres, Flavio Elias Gomes de Deus: Virtualization with Automated Services Catalog for Providing Integrated Information Technology Infrastructure. Proceedings of the 8th International Conference Autonomic and Trusted Computing (ATC 2011), Banff, Canada, September 2 – 4, 2011. Lecture Notes in Computer Science, Volume 6906, pages 75–91, 2011.

7. Robson de Oliveira Albuquerque, Luis Javier García Villalba: Group Trust Model. Proceedings of the 10th International Conference on High Performance Computing & Simulation (HPCS 2012), Madrid, Spain, July 2 – 6, 2012.

8. Robson Oliveira Albuquerque, Luis Javier García Villalba, Tai-Hoon Kim: GTrust: Group Extension for Trust Models in Distributed Systems. *International Journal of Distributed Sensor Networks*, pages 1–10, February 2014.

9. Robson Oliveira Albuquerque, Fábio Mesquita Buiati, Luis Javier García Villalba: Arquitectura de Seguridad Multinivel: Una Guia para las Organizaciones Modernas. Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2014), Alicante, España, Septiembre 2 – 5, pages 233–236, 2014.

10. Robson de Oliveira Albuquerque, Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Fábio Buiati, Tai-Hoon Kim: A Layered Trust Information Security Architecture. *Sensors*. 14(12):22754-22772, December 2014.

11. Robson de Oliveira Albuquerque, Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Rafael Timoteo de Sousa Jr., Tai-Hoon Kim: Relations in Cyber Security and Information Security and Trust. *Journal of Supercomputing* (accepted), September 2015.

Load Balancing and Survivability for Network Services Based on Intelligent Agents

Robson de Oliveira Albuquerque[1], Rafael T. de Sousa Jr.[1], Tamer Américo da Silva[1], Ricardo S. Puttini[1], Cláudia Jacy Barenco Abbas[1], and Luis Javier García Villalba[2]

[1] Universidade de Brasília
Campus Universitário Darcy Ribeiro
Faculdade de Tecnologia
Depto de Engenharia Elétrica e Redes de Comunicação
Laboratório de Redes - sala B1
CEP: 70910-900 - Brasília - DF – Brazil
{robson, tamer, barenco}@redes.unb.br, {desousa, puttini}@unb.br

[2] Universidad Complutense de Madrid (UCM)
Facultad de Informática, Despacho 431
Departamento de Sistemas Informáticos y Programación
Juan del Rosal, 8
28040 Madrid – Spain
javiergv@sip.ucm.es

Abstract. This paper describes the functionalities and the implementation of a load balancing scheme with fault tolerance characteristics to guarantee the survivability of a web service using intelligent agents technology. In this proposed model, agents are responsible for the automatic reconfiguration a traffic load balancer for web servers by exchanging information related to availability and workload and using an agent communication language. Besides this feature, with the objective of the network service survival, the intelligent agents monitor the load balancer and may assume its functions in case of failure and thus keeping the network service available.

1 Introduction

Web based systems have become a necessity in different areas. Also the system continuity in case of failures has raised as one of the most important system characteristics. This characteristic is related to the system quality in answering users demands.

In order to allow more to be done in a given amount of time and to have all users get served faster, it may be necessary to add more web servers and also to add controls according to the system architecture. This creates specific management problems depending on how many web servers are added and how they are controlled.

In this paper, a control methodology using a dynamic load balancing structure based on intelligent agents is proposed and implemented. The developed agents

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3043, pp. 868–881, 2004.

collaborate with each other, providing information about their environment and deciding, according to the information provided, specific issues related to the load balancing distribution and the service survival of a web based system. The agents are also capable of intelligent decisions in case of system failures as well as in case of agent own failures.

2 Load Balancing with Intelligent Agents Software

Using intelligent agents software in a load balancing structure can be considered as an application of survival and fault tolerance techniques with the main purpose of obtaining the quality of a network service. This was the main focus during the development of the solution presented hereafter. The following topics also explain some important issues related to this paper research area.

2.1 Fault Tolerance and Dependability

Dependability[1] indicates the quality of a service supplied by a specific system and the trust that is given in the service supplied by this system. To reach dependability some specific points should be attended, such as safety, reliability, and availability. These properties are correlated to measurable parameters of the system. For instance, related to reliability there are parameters such as mean time between failure (MTBF) and mean time to repair (MTTR), mean time to failure (MTTF).

Considering that most fault tolerance mechanisms in computer environments are related to the duplicity of items [2], the proposed model uses several web servers and more than one traffic load balancer. This environment is controlled by intelligent agents, configured with hardware and software elements connected together, and using a dynamic and adaptive control strategy that is transparent the final user.

2.2 Definitions for Load Balancing

Load balancing could be understood as the division of a common job among entities with the same final objective [3]. Considering this description, this division of work also needs acceptable quality and acceptable time response. It also needs some measurable parameters such as quantity of simultaneous connections.

Load balancing techniques are divided into two main categories: static and dynamic. In the static category, the quantity of work is divided among the entities in a pre-determined form and the entity that is responsible for the job division does not know the workload situation of the destination of a specific requisition. In the dynamic category the entity responsible for the job division knows and considers the workload of the destination of a requisition so it can choose the best destination to process a specific job and creates a dynamic load balancing distribution.

In this paper, the load balancing structure is based on the cooperation among the intelligent agents located in the web servers and also in the traffic load balancer. The implemented model can also work together with the load balancing structure based on Domain Name System (DNS). To exchange information, agents use agent

communication language (ACL). This characteristic is very important due to the necessity of a common language in agents environment that can permit agents to talk to each other using the Internet.

2.3 Network Service Survival

Network service survival[4] should be understood as "the capacity of a system perform its mission in a determined amount of time even though in presence of attacks, failures or accidents", in such manner that the system does what it is supposed to do during abnormal situations. In the environment described in this paper most of the survival characteristic relies based in two main factors. The first factor consists in a failure realized by the intelligent agents in a web server, the intelligent agents exchange information among them according to the environment actual situation and removes automatically the scant web server from the load balancing structure. It provides transparency to the final user in the system availability. The second factor is based on a redundant traffic load balancer that observes the behavior of the principal load balancer. Upon a failure in communication, interfaces or information exchanged, the redundant traffic load balancer confirms the failure performing specific tests in the environment and if it the case, assumes the load balancing distribution. An important characteristic of the traffic load balancers is that they keep a database of the load balancing structure because in case of a failure the system can reconfigure itself with the minimal configuration loss.

3 Developed Model

The model was implemented developing a particular methodology of valuation the CPU availability of the web servers. The developed model used Linux as main operational system for the traffic load balancers and for the web servers. The language used for development of the agents is based in Java Agent Development Framework (JADE)[5]. The JADE framework is based on the specifications developed by the FIPA research group [6]. The web servers used were Apache in version 1.27 without any specific improvements.

The model relies on that every web server there is a developed AgentWeb, and in every traffic load balancer there is a developed AgentRouter.

3.1 Model Environment

The principal traffic load balancer has the following functions: a. Performs network address translation (NAT). b. Traffic control and redirection of http requests to the web servers according to dynamic load balancing configuration. c. Protect the internal network.

In this case the principal traffic load balancer becomes a failure point for the hole system. If it fails the hole system fails. To solve out this specific problem the stand-by traffic load balancer is responsible for assuming the load balancing environment upon a failure of the principal traffic load balancer. Every Firewall/Router has a virtual

interface that is seen by the clients and by the web servers. Fig. 1 illustrates the model architecture.

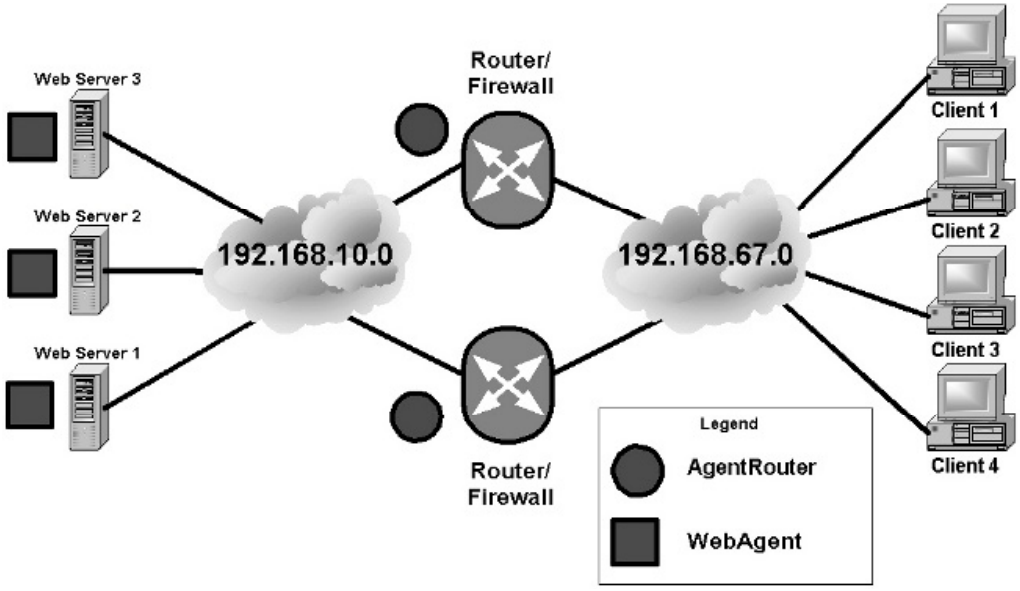


Fig. 1. Implemented Model Environment.

3.2 AgentRouter Architecture

The AgentRouter is divided into two specific agents. AgentMaster and AgentSlave. The AgentMaster is responsible for requesting and receiving information from the web servers. Once it receives the information, it processes it using an inference model and stores the results in a knowledge base. Than this knowledge base is shared with the AgentSlave. The hole communication process uses ACL model proposed by FIPA[6]. The AgentRouter also controls de interfaces of the routers and the firewall rules so it can manipulate the load balancing control dynamically. Fig. 2 shows the AgentRouter logical model.

The AgentSlave is responsible for watching the functionality of the AgentMaster. It also receives the information related to the AgentMaster knowledge base and which is a copy of the most recent information related to the web server CPU usage information and the load balancing structure. If the AgentMaster fails, for instance, one of its interfaces has gone down. Then the AgentSlave checks this failure and if it confirms the failure, it advises the web servers of the new situation, tries to communicate to the AgentMaster and requests it to shutdown the virtual interfaces and then assumes its position in the environment. If the AgentMaster returns it first checks the environment situation so it can take any necessary action.

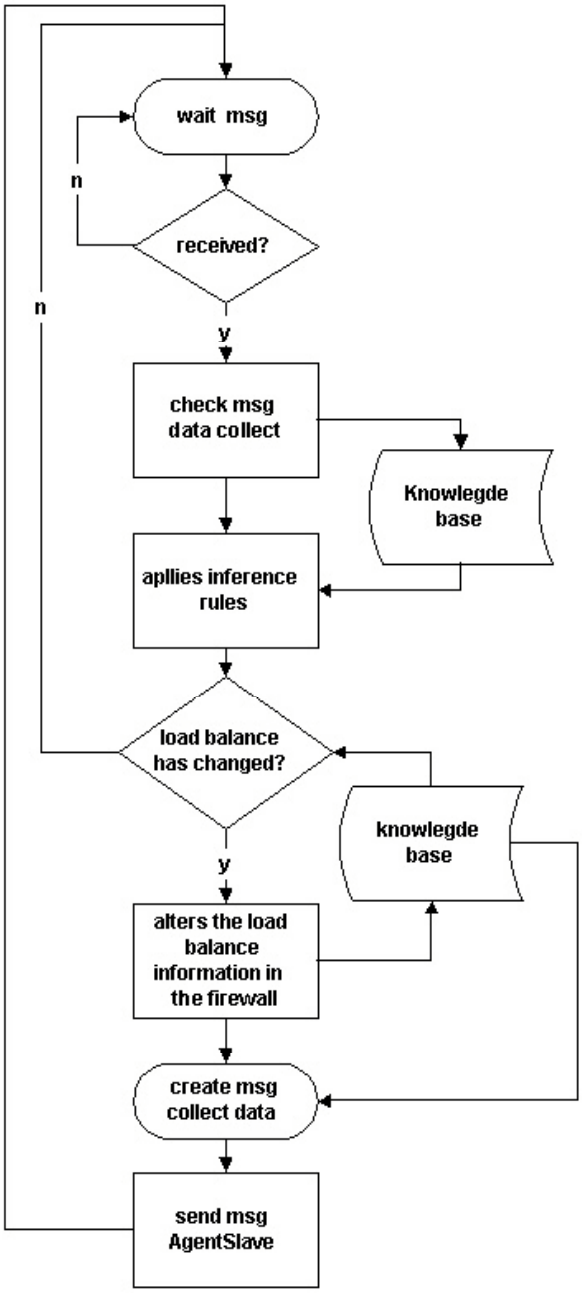


Fig. 2. AgentRouter Logical Model.

To decide which is master or slave in the environment, agents exchange information based on the most recent configuration. Fig. 3. and Fig. 4. demonstrate the logical model of the AgenteSlave.

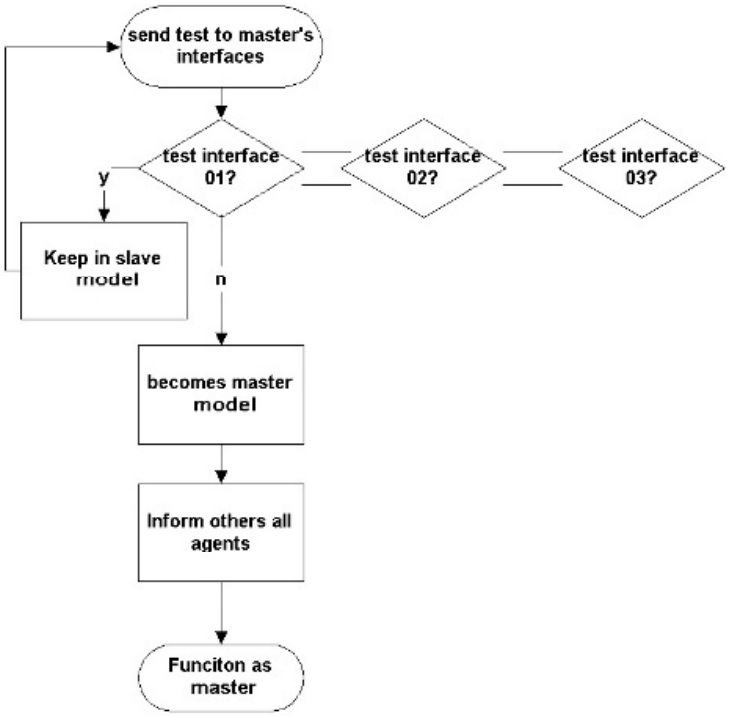


Fig. 3. Failure Check Logical Model of AgentSlave.

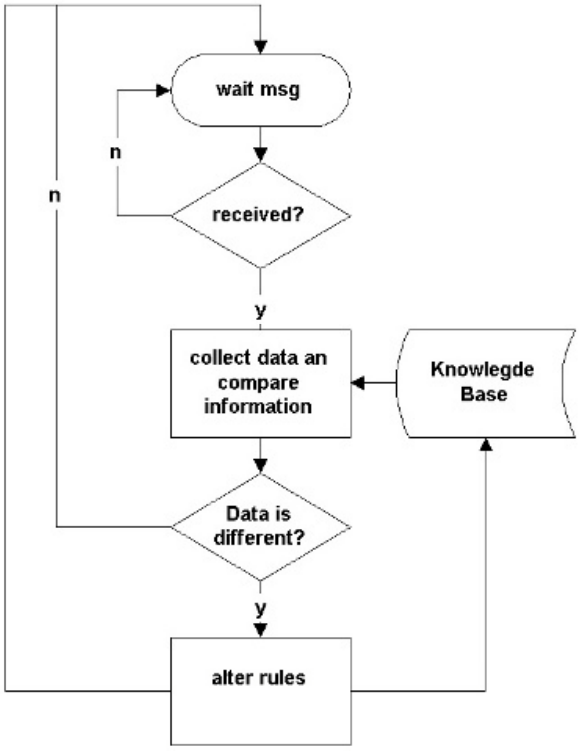


Fig. 4. Received Information Logical Model of AgentSlave.

3.3 AgentWeb Architecture

The AgentWeb informs the AgentRouter the real situation of the workload of the web server and checks for the web service response using http requests and checking if it is not an error message. Based in this information the AgentRouter can decide the load balancing architecture. If the AgentWeb does not send the information about the workload of the web server, the AgentRouter changes the configuration of the load balancing structure to the minimum configuration that web server may receive. Fig. 5. illustrates the logical model of the AgentWeb.

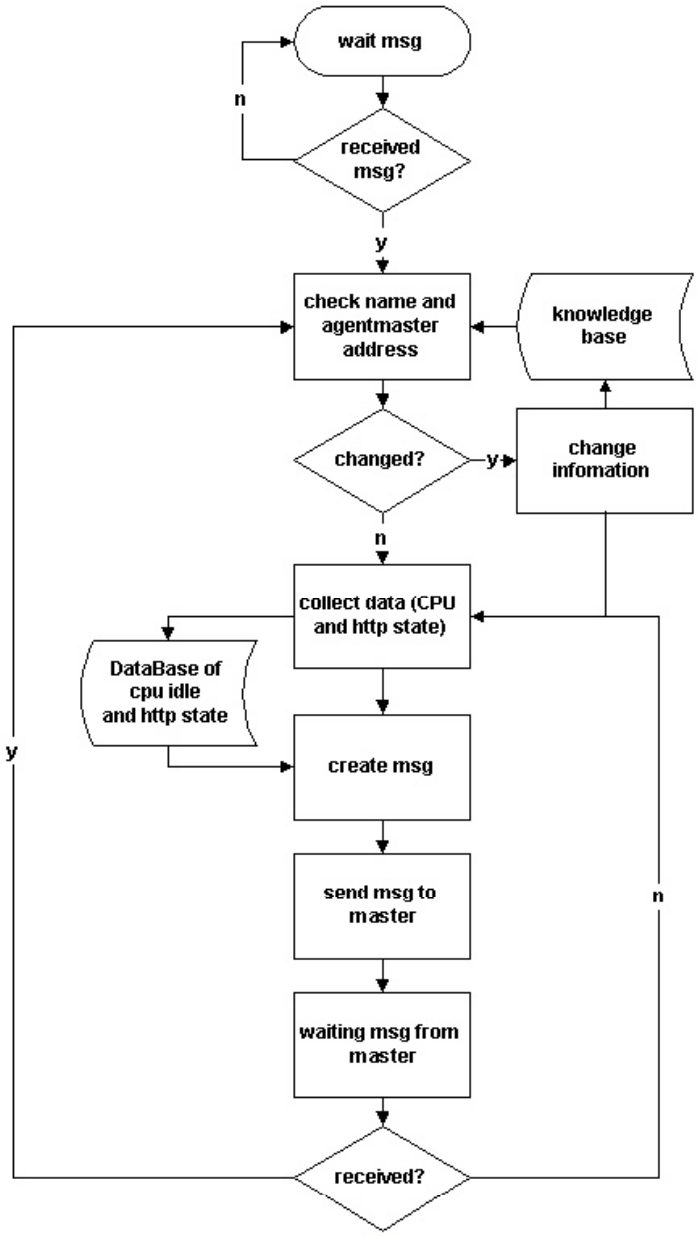


Fig. 5. AgentWeb Logical Model.

3.4 Agents Communication Model

The developed intelligent agents are fully based on JADE framework. To transport the messages agents use the message transport protocol (MTP). MTP itself uses TCP as IP transport protocol. The messages that agents send each other are based in http-formatted messages.

Based on this information, it was implemented a communication model to allow agents exchange information. The agents know and lean each other agents' location. Most of the communication is related to the environment and to check failures and report requested information. Fig. 6. demonstrates the developed model of agents' communication.

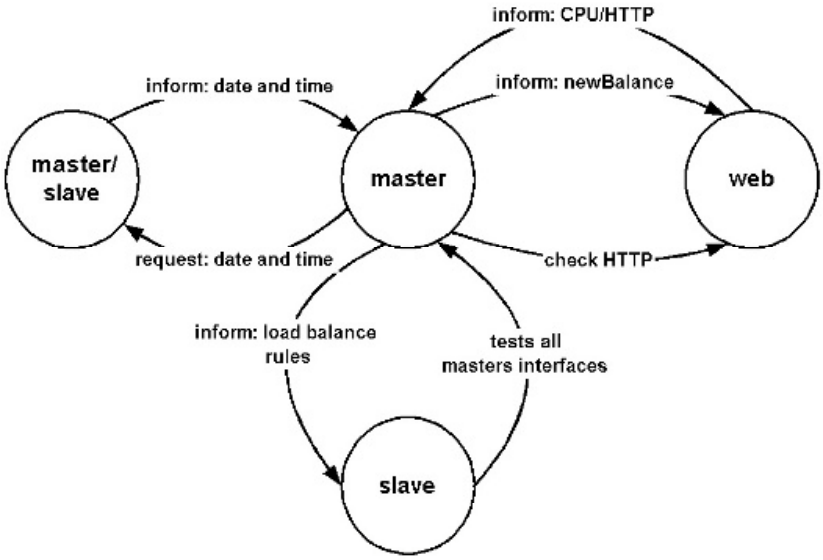


Fig. 6. Communication Model.

4 Tests and Results

To demonstrate that the developed agents were fully operational some tests were created. The characteristics of each test is based in an arithmetic average from three individual other tests because just one or two tests would not be reliable due to the quantity of variables analyzed. The main objective of these tests were to demonstrate that the proposed model works and permits an effective control of the environment and based on environment information the agents can decide what is best for the moment.

4.1 Tests Characteristics

One objective of these tests was to achieve a situation very similar to the Internet behavior in terms of www services. To reach this objective some tools were used to create an environment with a quantity of users that could generate a considerable

amount of web traffic. With this purpose very well defined, one free web application stress tool was used with some parameters defined. These parameters were related to: a. bandwidth control, so the packets would not get lost during the test. b. quantity of virtual users, so we could simulate concurrent users trying to access the resources. c. type of http request, it would be a get or a post request and if it would use Secure Socket Layer (SSL). d. quantity of clients simulating the virtual users. After the initial tests to validate de best environment, the parameters were defined as: a. bandwidth control: limited to 56K per virtual user simulating normal modems. b. 200 virtual users. c. Type of http request: GET request without SSL. d. Four (4) real clients simulating the 200 virtual users.

4.2 Test 1

The objective of the first test was to measure the web servers performance without load balancing and without the agents control. To achieve such objective the stress tool started controlling the real clients. The real clients generated approximately 24.000 http requests in 180 seconds.

The web server logs reported that less then 0.3% requisitions generated some kind of errors. It had approximately 99.7% of requisitions fully attended.

After analyzing the graph generated by the requisitions we could realize a couple of things. First, in the beginning of the test the http requisitions varied more then in the middle and in the last part of the test. This characteristic is related to the stress tool which has a warm up time. Second the test average took about 110 to 160 requisitions per second with a lot of different peaks. Fig. 7. shows the distribution of http requests during the test.

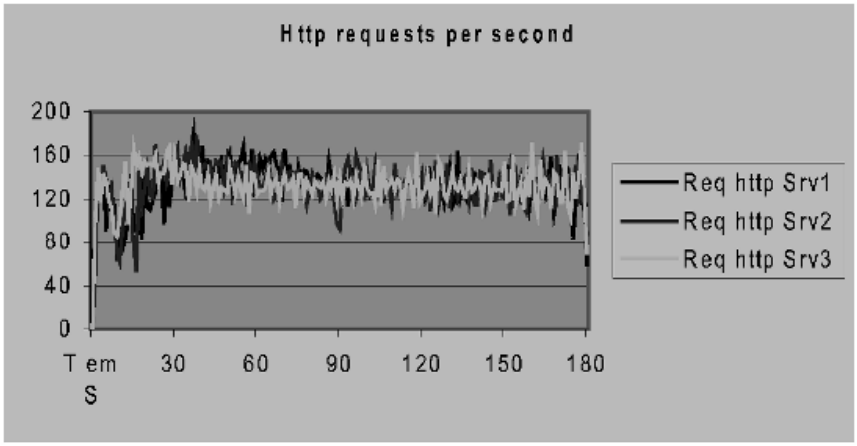


Fig. 7. Http Distribution for Test 1.

After generating the http distribution, it was generated the CPU consumption for each web server to verify the CPU distribution related to the job. Fig. 8. shows the results for CPU distribution for test1.

As shown in Fig. 8. , the CPU_idle stayed bellow 50% of usage. This represents that the web servers were very busy during the test so they could attend the requisitions.

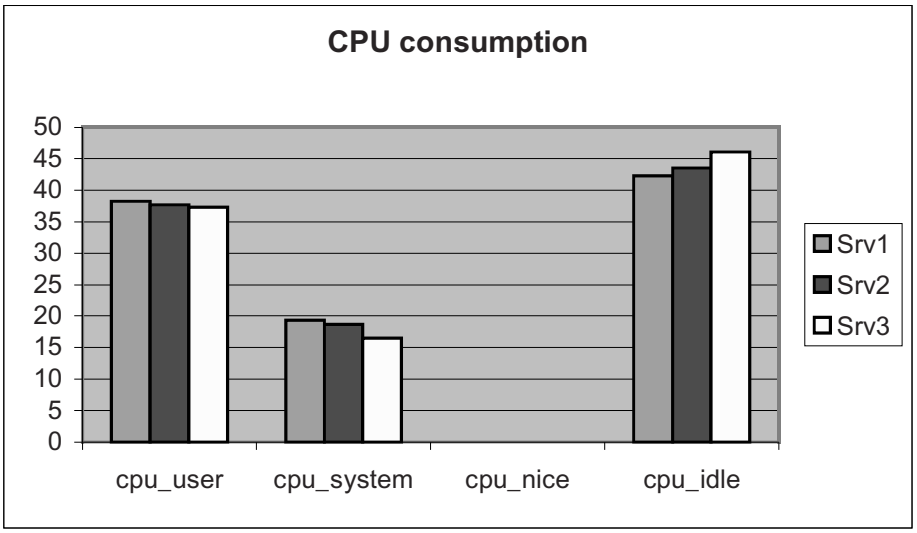


Fig. 8. CPU Consumption for Test 1.

4.3 Test 2

The objective of the second test was to set up a static load balancing structure and measure the behavior of the web servers. It was introduced a traffic load balancer between the clients and the web servers. Again the test reached approximately 24.000 http requests in 180 seconds.

The web server reported that less then 0.1% of the total requests generated some kind of error. It had approximately 99.9% of requisitions fully attended. Fig. 9. shows the behavior of the distributions. Comparing this test the with the first test, the warm up time of the stress tool generated much less variations in the beginning when using a traffic load balancer. Also the average of the http requisitions kept below 60 requests per second with very low peaks difference. This gives more justice in the requisitions distribution.

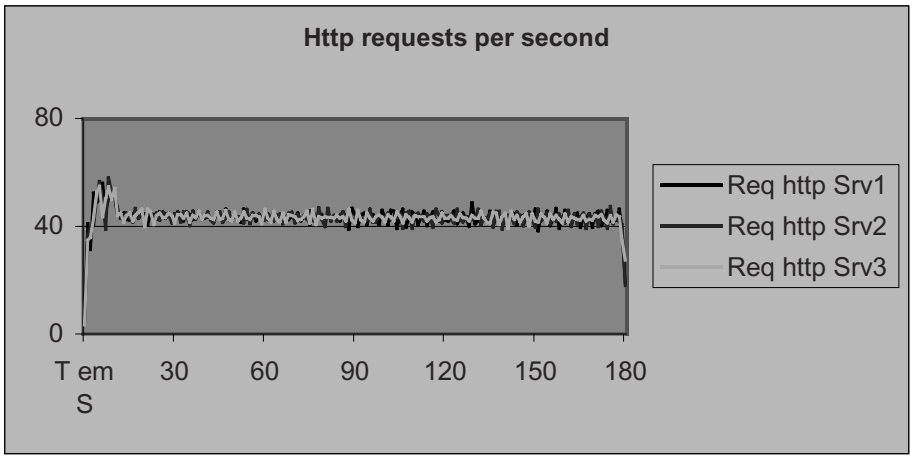


Fig. 9. Http Distribution for Test 2.

Fig. 10. shows the CPU consumption for the test 2. Comparing the values with the first test there is some specific points such as: a. cpu_idle analysis resulted in almost 100% of difference in behavior, so the web servers could work better because of the division of the workload. The first practical conclusion that is shown is that the load balancing structure did work well, but is does not consider the situation of the web servers. This situation demonstrates a static load balancing structure.

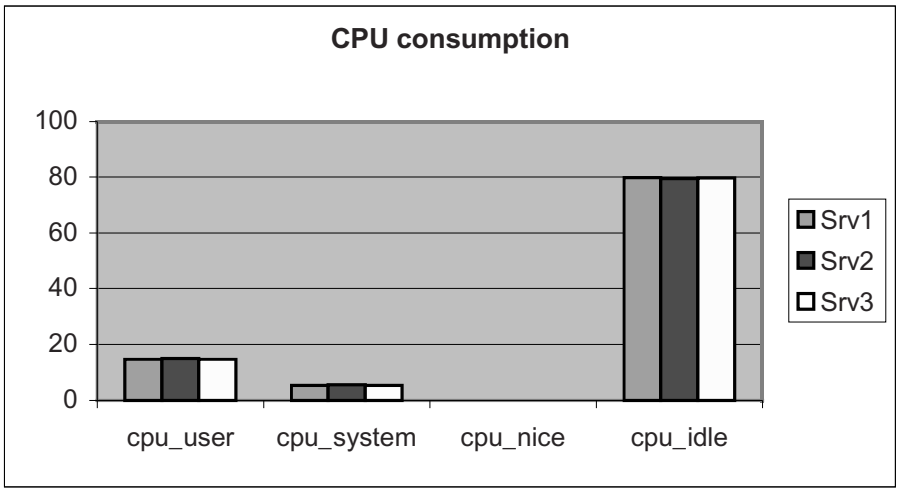


Fig. 10. CPU Consumption for Test 2.

4.4 Test 3

The objective of this test was to introduce the intelligent agents in the environment and measure the behavior of the web servers.

Using the load balancing structure demonstrated in test 2, it was introduced the AgentRouter in the traffic load balancer and it was introduced the AgentWeb in the web servers. Once again the test reached approximately 24.000 http requests in 180 seconds. This test results are shown in Fig. 11. and Fig. 12. .

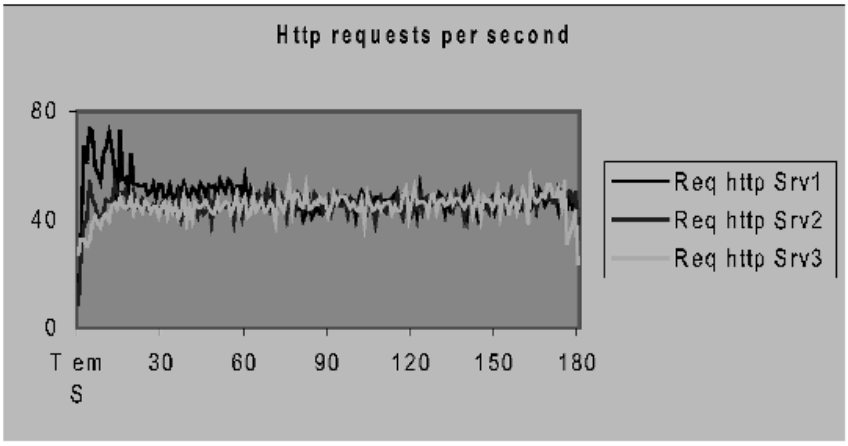


Fig. 11. Http Distribution for Test 3.

Comparing the results of the http distribution with test 3 we can see that in the beginning of the test, during the warm up of the stress tool, the agents realized that web server 1 should receive more requests. During the rest of the test the agents realized that the CPU situation of the web servers were similar and started distributing the http requests more efficiently. The peaks kept in the average in 65 http requests per second approximately.

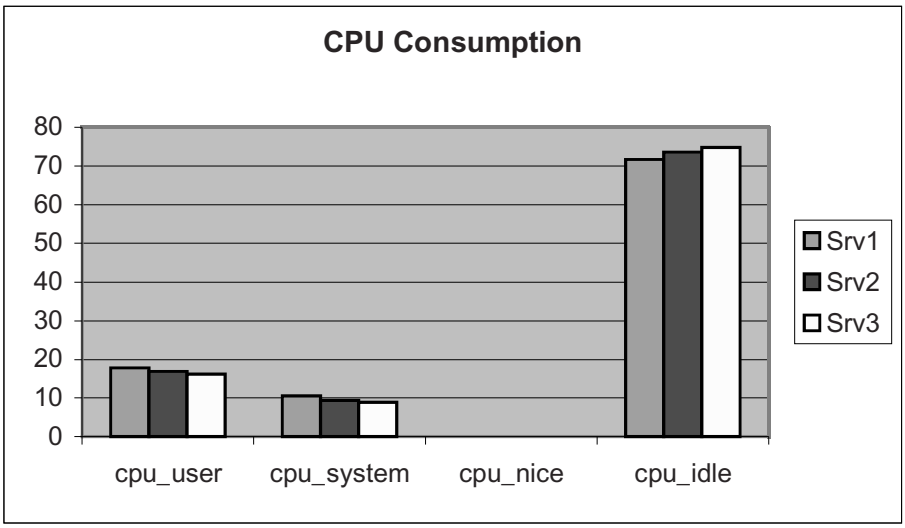


Fig. 12. CPU Consumption for Test 3.

The CPU use kept very close to test 2 results. These differences were expected due to the introduction of the intelligent agents process on the system. The difference is less than 8% when using the intelligent agents. The web server reported that less then 0.1% of the total requests generated some kind of error.

After these test, it was possible to create an environment that did dynamic load balancing using intelligent software agents. The most important consideration is that the system, using the intelligent agents, adapts itself to the real situation of the environment.

4.5 Test 4

After the third test is was possible to check some differences in the environment. But still, it was needed to certify that the agents would react to an abnormal situation of the environment. This was the objective of the fourth test. During the execution of the test, it was started a process into web server 3 to consume all CPU resources and this process kept until the end of the test. This test results are shown in Fig. 13. and Fig. 14.

At the beginning of the first minute of the test it was started the process responsible for consume all CPU resources in web server 3. Throughout the information exchange the agents realized that web server 3 had some kind of problem related to CPU resources, but it still had an operational web service. Agents decided to distribute more requests to web server 1 and 2 due to the problem with web server 3. They also

kept the web server in the load balancing structure because the web service process did not stop answering http requests.

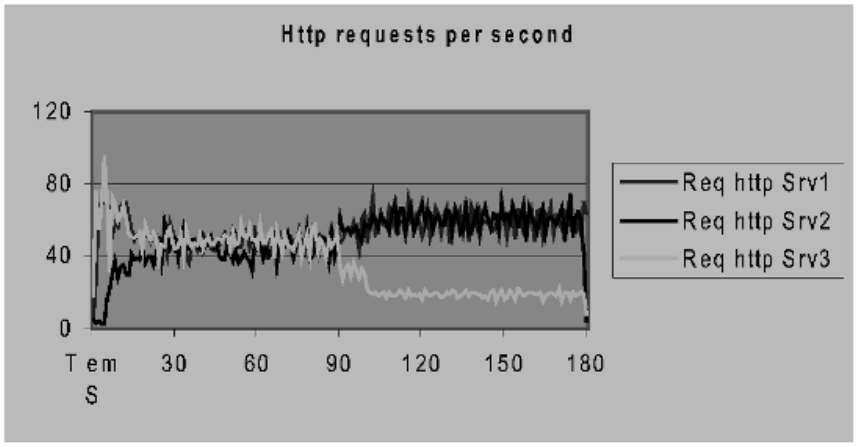


Fig. 13. Http Distribution for Test 4.

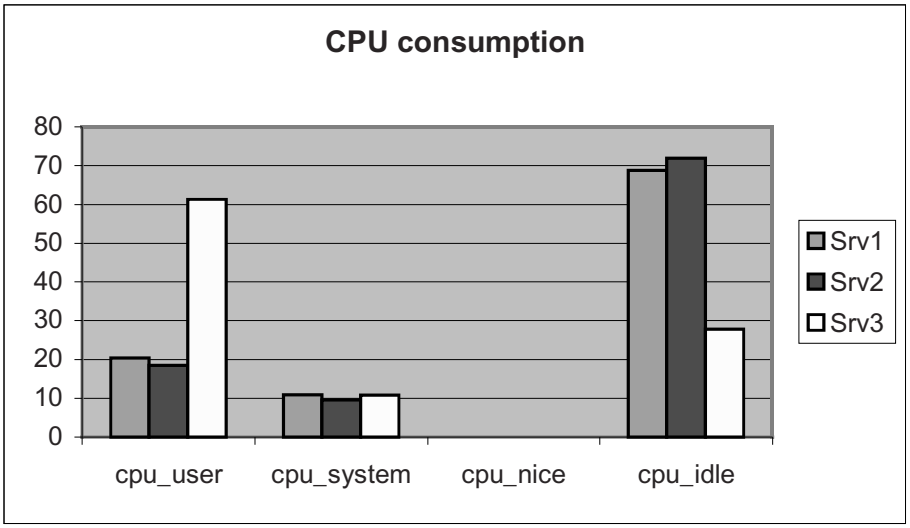


Fig. 14. CPU Consumption for Test 4.

Related to CPU consumption Fig. 14. shows that the web server 3 kept its CPU usage very high.

5 Conclusions

The use of intelligent agents is efficient in terms of dynamic load balancing and service network survival in many different points, such as: a. if service network survival is strongly necessary in the environment, the agents can react according to the real situation of the environment and keep the network service alive to the final users. b. the exchange of information among the agents does not reduces the performance of the environment. It means that the amount of messages passed from

an agent to another does not directly imply in a large quantity of bandwidth usage. c. using dynamic load balancing can give the system more chances of justice related to the distribution of the workload. d. the model that was projected and developed in this paper demonstrated that through the combinations of resources like web clusters, routing procedures, network address translations, firewall operations, it is possible to solve out problems related to high availability systems with a limited amount of resources, including open source systems. e. using intelligent agents in systems may produce good results in terms of quality, performance, fault tolerance systems, network service survival, using limited resources and creating a powerful mechanism of load balancing distribution in web services.

Though it is an effective solution, the proposed model needs some specific optimization related to the integration of the agents with Snort [7] Intrusion Detection System (IDS). Workload distribution performance is another necessary step to improve this model. The service network survival needs better time response when related to changing IP addresses.

The proposed model associated with these future works may become an effective toll that could be used in different organizations where the necessity of workload traffic distribution is needed and the amount of resources is limited. Besides, the use of intelligent agents is a technology that can be applied to help systems administrators in managing their information systems.

Acknowledgements. Luis Javier García Villalba's work is supported by the Spanish Ministry of Science and Technology (MCYT, Spain) under Project TIC2002-04516-C03-03.

This author would like to express his appreciation to the Programa Complutense del Amo for providing him a grant to stay at IBM Research Division. During part of this work he was with the Information Storage Group at the IBM Almaden Research Center, San Jose, California, USA (javiervi@almaden.ibm.com).

References

1. Weber, Taisy Silva, 2000. Tolerância a falhas: conceitos e exemplos. Editora da Universidade Federal do Rio Grande do Sul.
2. Avizienis, A, 1998. Infraestructure-based design of fault-tolerant systems. In Proceedings of the IFIP International Workshop on Dependable Computing and its Applications. Johannesburg, South Africa, pp. 51-69.
3. BRAGA, A.R.,2001. Emprego de Agentes Inteligentes no Balanceamento de Carga na Interface de um Site de Educação à Distância. Editora Universidade de Brasília
4. Elison, B. et al, 1997. Survivable Network Systems: An Emerging Discipline. In CMU Technical Reports, CMU97-13.
5. Bellifemine, Fabio. et al, 1999. JADE – A FIPA-compliant agent framework. In Proceedings of Forth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology. England, London, pp.97-108.
6. The FIPA Application Specification, 2002. The Foundation for Intelligent Physical Agents. Available from http://www.fipa.org.
7. Snort, 2002. The Open Source Network Intrusion Detection System. Available from http://www.snort.org.

MANET - Auto Configuration with Distributed Certification Authority models Considering Routing Protocols Usage

Robson de Oliveira Albuquerque, Maíra Hanashiro, Rafael Timoteo de Sousa Junior, Claudia J. B. Abbas

Universidade de Brasília - Campus Universitário Darcy Ribeiro - Faculdade de Tecnologia - Depto de Engenharia Elétrica e Redes de Comunicação - Laboratório de Redes - sala B1 - CEP: 70910-900 - Brasília - DF – Brazil

Luis Javier Garcia Villalba

Universidad Complutense de Madrid (UCM) - Departamento de Sistemas Informáticos y Programación (DSIP) - Facultad de Informática, Despacho 431- C/ Profesor José García Santesmases s/n - Ciudad Universitaria - 28040 Madrid - Spain

Abstract. In this paper, we discuss about certification, authentication, auto configuration and routing for mobile ad hoc networks (MANETs). The proposal is based on the works of [1] and [3]. We describe distributed certification, MAE authentication, auto configuration process and routing protocols. Then, we show some problems of these models and we propose some solutions considering routing and others protocol modifications.

1 Introduction

Wireless networks are defined as computers networks that are connected to its work area through wireless links, such as radios frequencies and infrared rays. Wireless local area networks (WLAN) arised with the main purpose to overcome the limitations imposed by traditional wired networks, thus permitting faster network installations and mobility.

According to 802.11 [4] standard, established by the IEEE board founded in 1990, WLAN can be sorted in independent networks (Ad Hoc) and access point dependent.

In an infrastructured WLAN (based in access point) all communication among mobile nodes (MN) goes through mobile support stations (MSS) and usually it is directly connected to a wired network. In this situation MN cannot communicate among each other directly.

In Ad Hoc WLAN, refered as Mobile Ad Hoc NETwork (MANET) by IETF, MN can communicate with each other because there is no MSS. In this kind of networks,

de Oliveira Albuquerque R., Hanashiro M., Timoteo de Sousa Junior R., J. B. Abbas C. and Javier Garcia Villalba L. (2005).
MANET - Auto Configuration with Distributed Certification Authority models Considering Routing Protocols Usage.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 57-66
DOI: 10.5220/0002578800570066

MN does not require any physical infrastructure and nodes can move freely because there is no central communication point.

Ad Hoc WLAN are mostly used in situations where it cannot or does not make any sense, install a fixed wired network, such as disaster situations, hurricanes, earthquakes, where rescue teams needs coordination and communication. Soldiers in a battlefield exchanging tactical information, businessman receiving information in business meetings, students using laptops in classrooms. In a near future, Ad Hoc networks shall have an important paper in wearable computers interconnection, sort of future computer that can be attached to human body, for example, a computer jacket.

An Ad Hoc WLAN can operate isolated or it can be an extension of some wired network already installed, which, in this case, needs a communication gateway to connect each other.

As advantages of MANET it has quickly installation (can be installed in areas with no previous infrastructure because it needs no fixed base to route messages), fault tolerant (any malfunction or disconnection of a station can be easily solved with dynamic reconfiguring of the network), connectivity (if two stations are inside the same area where there is reach of radio waves, there is a communication channel), mobility and others.

Based in RFC 2501 [5], some characteristics and fragilities are important in these networks. These characteristics and fragilities are related to dynamic topologies, restricted bandwidth and variable links capacity, power save consumption operation and limited physical security.

Due to these problems, MANET needs proper specifications related to certification, authentication, configuration and routing.

In this paper some proposals related to certification and auto configuration with routing considerations are presented and fundamented in [1] and [2] developed work. Besides that, some problems are emphasized and possible solutions are shown as considerations and possible solutions related to auto configuration and distributed Certification Authority (CA).

2 MANET routing protocols

Routing protocols are responsible for finding, establishing and keeping routes between MN that wishes to communicate. It is very important that routing protocols in MANET creates very few messages as possible, avoiding network overhead and thus not consuming network bandwidth. These factors are directly connected with the velocity that network routes are established and the frequency that they are updated. Different techniques were developed creating protocols that can create and establish routes faster than others. Others can consume less bandwidth but takes more time to establish a specific route.

According to IETF MANET workgroup (IEEE, 2004), there is a desirable quality list that routing protocols are required to supply with: (a) distributed operation, (b) no routing loops, (c) under demand operations, (d) pro-active operation, (e) security, (f) inactivity period operation and (g) unidirectional link support.

Basically MANET routing protocols can be classified as reactive and pro-active. Pro-active are routing protocols that keep information about routes to every MN in the network. Reactive protocols only create a route when it is requested by origin node.

Four routing protocols are specified by IETF with drafts RFC: (a) TBRPF [6], (b) OLSR [7], (c) AODV [8] and (d) DSR [9]. Where (a) and (b) are considered pro-active routing protocols and (c) and (d) are considered as reactive routing protocols.

Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) creates per hop routing by the shortest path for each destination. Each MN running TBRPF generates a topology information tree based in information topology that is saved in a topology table. To minimize network processing, each MN reports only a few portion of is topology table to neighbor MN. TBRFP uses different combinations and periodical updates to keep every MN informed about its own topology tree. To reach and keep robustness in highly mobile environments in the protocol, each MN can send additional information (complete topology tree) to its neighbors.

Differentiated HELLO messages are used to neighbor discovery that contains only information about neighbor change. This modified message results in shorter messages based in link state algorithm.

TBRPF can be divided into two main modules. The first module is called "neighbor discovery" and the second is called "routing" which does the topology discovery and computes the routes to every destination.

Optimized Link State Routing Protocol (OLSR) has as key concept the use of multipoint relays (MPRs). MPRs are MN selected to forward broadcast messages in the routing protocol flooding mechanism. MPRs are spread throughout MANET to provide every MN the partial information about the necessary topology that computes the best route to every MN in the network. MPRs combined with local duplicity avoidance are used to minimize the number of control packets that should be sent in the network.

OLSR is projected to work in high scalable networks where traffic is sporadic and randomly among specifics MN. As a pro-active protocol, it is also adequate to scenarios where pairs of MN changes very often, but no additional control packet is generated in the network since the routes are kept and known by all possible destination.

Ad hoc On-Demand Distance Vector Routing (AODV) is based in Destination-Sequenced Distance Vector Routing Algorithm (DSDV) protocol. In general AOADV tries to cut off the need of broadcast routing messages, which limits its own scalability. Another important AODV point is that it tries to minimize the latency when new routes are required.

AODV is classified as distance-vector algorithm and is considered as a reactive protocol because only one route is created if it is necessary. In general, AODV tries to eliminate the broadcast routing messages flooding, which limits its own scalability. AODV also tries to minimize latency when new routes are requested. It functions is similar to traditional algorithms what can facilitate the interconnection with wired networks. Even though working very closed to traditional protocols, AODV allows multicast and unicast traffic, however the protocol shows only one route to every destination what cannot be a good characteristic.

Dynamic Source Routing (DSR) is a simple and efficient routing protocol designed to multi-hop MANET with up to 200 MN and supports high mobility rates.

It allows the network to organize and auto configure itself without the network infrastructure administration.

DSR is divided into two main modules: “routing discovery” and “routing maintenance” that work together to permit that MN discover and keep updated routes. All aspects of the protocol work under demand, thus not sending periodical messages to routing exchange information. This characteristic of the protocol allows low network bandwidth consumption and power saving.

DSR also permit multiple routes to a specific destination and that every sender selects and control the used routes to forward its packets. Other advantage of the protocol is that is provides loop-free routing information, supports unidirectional links and fast convergence when the network topology changes.

3 MANET routing protocols

To avoid malfunction in MANET it is necessary security in message routing. Besides in [3, 4] and [11] an authentication service for routing protocols is proposed, where Manet Authentication Extension (MAE) is used and attached to every message in the routing protocol. All necessary information to authentication are included in MAE. The main focus of the proposed model was to keep the routing packets and its messages unchanged. MAE format is shown in Fig. 1.

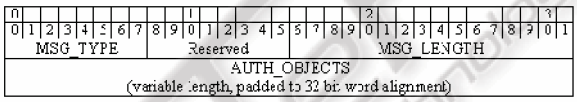


Fig. 1. Mae Syntax

MSG_TYPE field is used to differentiate MAE messages from other routing protocol messages. MSG_LENGTH field indicates the size of MAE in bytes and AUTH_OBJECTS has the objects that have authentication information.

4 Distributed Certification

In MANET, due to its characteristics, is heavily not recommended any centralized service. In [12], [13] and [1] approach a distributed certification model is proposed.

The model proposed uses threshold cryptography theory and pro-active secret key update based in Shamir [14] schema.

In the system point of view, the whole architecture is fully distributed and the service is localized using a coalition approach and the cryptographic system is fully based in RSA model.

It is considered a MANET where every NM v_i has a personal RSA key pair $\{sk_i, pk_i\}$, where $sk_i =< d_i, n_i >$ is the private key and $pk_i =< e_i, n_i >$ is the public key that are used in point-to-point transactions.

A Certification Authority (CA) has a key pair $\{SK, PK\}$, where $SK =< d, n >$ is used to sign all MN certificates. Any certificate in this approach can be verified by the system public key SK, that is know by ever MN in the network.

According to threshold cryptography, SK is divided in the network. Every MN v_i, besides its own key pair, has the partial key P_{vi}. Any subgroup k of n MN can work as a CA. However it is not possible to any MN to know SK, but in the system initialization.

Threshold cryptography is indicated in MANET due to some of its proprieties: (a) the distribution and decentralized control of the keys fits the profile of Ad Hoc networks, (b) security omnipresence is guaranteed since the secret is fully distributed in the network and intrusion detection is more practical and efficient, (c) the limit k is the balance between the service availability and intrusion tolerance. In other words, a group of adversaries need to destroy $(n-k+1)$ partial key holders to bring the system down (once it would block one auto configuration) and at least break k partial keys to steal SK secret.

System is initialization is a very careful step to k choosing. As lower the k value the greater the facility of break SK secret. In other hand the greater the value of k the higher the system security, which reduces fault tolerance at the same time. After all, the most close k is from n, the probability of $(n-k+1)$ MN leaving the network raises, which would forbid the service.

Certificates generated by a CA formed by a subgroup of k MN have the finality of certificate, as in a normal cryptographic system, the public key of every MN. Therefore, every MN has its own $cert_i$ certificate that must be signed by SK, in $< v_i, pk_i, T_{sign}, T_{expire} >$ format, where v_i is the MN identifier, pk_i is its public key, T_{sign} is the signature date and T_{expire} is the expiration certificate date.

To control the certificate validity are used to methods: (a) Implicit certificate revocation that defines that every MN must renew its certificate at least ever period T_{renew} where $T_{expire} = T_{sign} + T_{renew}$, (b) explicit certificate revocation where a certificate is assumed by Certificate Revocation List (CRL) is not valid anymore even its T_{expire} is valid. This implies directly that only revoked certificates that did not expire must be in CRL.

This model was implemented in [1] which involves only subgroups, k size, of partial key holders. The basic operations include: (a) secret key negotiation, where the secret key can be obtained by on MN with the system initialization or with the auto configuration service. In the first case, both keys and certificates are distributed to MN by a central negotiator before MANET formation. In the second case, an auto initialization algorithm where k MN can provide a partial key to new MN in the network, (b) the secret key update, instead of changing the system key from time to time, only changes the partial key with the main purpose of protecting the secret key from being broken. The system supports until $k-1$ partial secret breaks because SK is obtained with k keys. If in a update situation there is less than k discovered keys, SK is protected and does not need to be changed, (c) the certification service permits, that when a MN requests using the certification service, one subgroup of k (coalition) partial secret key holders is created and every MN v_i generates a partial signed certificate to the requesting MN. MN then generates its certificate by grouping k received certificates that represents a signed certificate from SK. This service includes emission, renovation and revocation of certificates, besides, even before the MANET formation, a security policy for each step should be defined.

5 Auto Configuration

A MN to communicate in a network must have a unique identifier, usually the IP address. However, in MANET the topology changes dynamically thus creating a difficult environment for centralized administration that can distribute IP address or any other identifier. This situation leads to a distributed, dynamically and automatic service.

Together with security and routing protocols, auto configuration provides a service that can became MANET more efficient and robust. Even though there are many approaches related to auto configuration, none has been standardized.

In [14] is proposed an auto configuration model that uses message authentication considering the distributed CA model in [12, 13] and [1]. In [2] approach a protocol for auto configuration is developed considering a distributed CA, which avoids that any intruder MN can produce messages or even change the messages already created with the purpose to break the protocol or get the service unavailable. To reach this situation in MANET, according to [2], the MN where already configured with a valid certificate before they can request and join the auto configuration service.

Therefore to a MN request an IP address or even respond to MN client solicitation, MN must have a valid certificate. The authentication service of the auto configuration mechanism is supplied by MAE, which has all the necessary information to guarantee authenticity, integrity, non-repudiation in all MAE protected messages.

MAE used for the proposed auto configuration model is the same proposed to protect the routing messages in MANET routing protocols.

As referenced before MAE has authentication objects which includes Digital Signature (DS) that is mandatory and authenticate all non-mutable fields of auto configuration messages. MAE should have one more object, that can be the certificate. The message sender must use DS with its private key because the certificate that goes with MAE has the sender public key that can be used to certify the message sender. If the MN certificate is not locally available, MAE can have a CERT object, which carries with the message the certificate that created and signed MAE. Additional objects are used to provide additional services that are beyond the protocol auto configuration approach.

Every NM that is valid and trustable belonging to MANET has an IP address identifying its interfaces and a subset of free IP address (FIA) to offer to MN clients that wishes do associate to the network.

Inside an individual MANET, A MN FIA must be distinguished from others MN FIA thus avoiding that the same IP address can be distributed by more than one MN, besides that, every MANET has a unique identifier defined as partition ID (PID), which, in this situation permits that to MN that has the same PID are in the same MANET. PID also helps distinguishing different MANET in a specific area and also helps different MANET to be brought together.

Dynamic Configuration Distribution Protocol (DCDP) is used to distribute network configuration information such as IP address, network mask and default gateway, which uses binary division to provide to MN different IP address in the network. Binary division assures that all MN receives distinguished IP address, thus avoiding IP address conflicts even in a MANET join situation.

In [2] to obtain and associate an IP address the MN must have received its valid certificate. When a MN wishes to join a MANET so it can obtain an IP address, it sends an ADDR_REQ message in broadcast using its MAC address as source address.

Any MN belonging to the MANET answers the message with ADDR_REP that contains FIA with the biggest free IP quantity because a MN can have more than one FIA with different quantities. The MN can receive more than one answer from different other MN and then selects the MN that has the biggest FIA sending a SERVER_POOL message directly to the chosen MN server, discarding all other received messages.

The SERVER_POOL message confirms the MN intention of getting an IP address. The elected MN server then divides its FIA, sending one half to the MN that requested it and keeping the other half so it can answer future requests. The MN that received the FIA throughout IP_ASSIGNED message assigns the free IP address in its own FIA. The first IP address the MN uses for itself associating it with its interface and using all the rest as FIA to answers MN client requests.

If a MN has more than on FIA, for security and implementation facility reasons, the MN must mark in which FIA is its own address. The process is finished using an IP_ASSIGNEMENT_OK message to the server MN.

6 Related problems and proposed solutions

In [1] a MANET distributed CA was created and implemented. The proposed model relies in k size. This implies directly that k MN must be reached so a MN can have its certificate signed. If k MN are not reached, the MN cannot join MANET because it cannot sign its certificate. A routing protocol should then be used to reach k MN thus permitting the certificate signature.

Another problem related to k is that it has a fixed value that is defined considering a relative size so that k cannot have a big value (close to the total amount of MN in the MANET) and neither very short size (related to the quantity of MN in MANET). However the size of MANET is highly variable thus implying that an adequate defined k value may become inadequate considering that a MN can leave or join the MANET at anytime.

An initial solution is that k may vary in function of the size of the percentage of the network, but alter k is important define maximum and minimum values (both related to a percentage of the size of the network) of MN in function of the security necessity of the network and these values should be monitored as the quantity of MN in the MANET raise or reduce, thus implying directly that if a minimum or a maximum value is overpast is necessary a redefinition of k. According to the analyses of the results obtained in [1], the value of k can be defined as an average of the maximum and the minimum size.

Considering that k may vary from time to time, the model needs improvements in the CRL because the number of revoked certificates would be much bigger because

the certificates are fully dependent on k. At this point we have the relation that the most k varies the most will be the emission of revoked certificates and the most will be the emission and requests of new certificates. This generates more traffic in the network and thus forcing the MN to process new certificates raising power consumption. Besides the variation of k, as MN enter and leaves the network, the certificates are automatically revoked but new certificates needs a new CA initialization. But according to [1] the process of CA initialization is centralized, contradicting the MANET's necessity.

In [1] model approach, to solve out this problem here presented is necessary the creation of a model of distributed CA initialization that implies in new mathematical models to the generation of a distributed key.

In other hand if k MN has to be reached, these MN can be reached using routing protocols to the signature of a previous requested certificate. This problems requires that a MN can work as a proxy, asking in the MN's name that others $k-1$ MN sign the certificate request. Considering that the proxy MN already has a valid configuration in the network related to IP address it could request the certificate to be signed using routing measures if $k-1$ could not be reached for itself.

Another approach considers that it could be used a temporary IP address to request the certificate signature. This implies that a topology change is required because of the temporary IP chosen by the MN. To solve out this problem a range of IP network address (even in CIDR) could be allocated and announced in the network informing that if a MN wishes to sign a certificate so it can join MANET, it then should use an IP address range reserved to that finality.

Considering this situation OLSR could be used as routing protocols because of its pro-active characteristic, besides the information messages to the reserved IP range could be announced by MPRs. A time-to-live (TTL) should be limited to 2 or 3 hops because is highly probable that $k-1$ nodes could be reached by routing. Another consideration is that it would limit the traffic related to certification signatures.

Another point is that any pro-active routing protocol could be used in this situation because the routing information would be easily created because the IP range would be well-known in the network.

In [2] the distributed CA approach implemented in [1] was used and the routing considerations where not applied limiting the reach of the auto configuration model proposed. This returns to the considered approach pointed herein because in (BUIATI, 2004) is assumed that the MN already has a valid signed certificate. So the proposed solution to [1] can be easily applied in [2].

Another problem in [2] is that the auto configuration model relies in that every message sent in the network is broadcast messages. This process makes the proposed auto configuration model not scalable because in huge MANET the amount of messages would increase significantly creating problems related to unnecessary bandwidth consumption and increasing power consumption by the MN.

To solve out this specific problem the protocol should be changed so that only the first message is broadcasted to reach all close MN. In this message the MAC address of the MN goes with the frame. As the MN server receives the sender MAC address

the other messages in the communication process can be done in the unicast approach thus avoiding the flooding of the network.

7 Conclusions

MANET is increasing highly but some problems should be fixed out due to its characteristics. Related problems about auto configuration, routing measures, distributed CA are increasing as MANET standards are developed.

The approach studied in this paper is related to the problems found in [1] and [2] and the proposed solutions considering routing and others protocol modifications so both models can be more heavily developed and studied.

In [1] approach the proposed solution relies in static k, but in MANET the number o MN cannot be easily predicted. In other hand k is defined considering n. As n increases or reduces, k cannot vary because the whole process needs an initialization to secret key creation that is centralized. This approach was not considered in the implemented model and thus pointed herein to future researches in this subject. The initial proposed solution is based in new idea that relies in a fully distributed CA initialization approach so k can vary according to the necessity of MANET. It is important to say that this model should be heavily studied to validate the proposed solution.

In [1] the routing measures to reach k is not considered because it assumes that k are close of the requesting MN, which in MANET may not be true due to its mobility. An initial proposed solution considers that routing protocols can be used as proxy to reach k MN in order to produce a signed certificate.

In [2] the model is based in broadcasts messages during the whole auto configuration process. This paper proposes that the protocol should be changed in order to avoid unnecessary bandwidth consumption and thus avoiding power consumption. Once the first message is sent the MAC address of the sender can be easily obtained and the consequent communication process can be done using unicast approach.

Both [1] and [2] are heavily fundamented and very well work were developed permitting that new approaches and researches could be conducted using both proposed models in order to allow secure auto configuration and distributed CA in MANET.

References

1. SILVEIRA, F. AND HANASHIRO, M., *Serviços de Certificação para Redes Móveis Ad Hoc*. UnB, Brazil, 2003.
2. BUIATI, F.M., *Protocolo Seguro para Autoconfiguração de Endereços de Redes Móveis Ad Hoc*, UnB, Brazil, 2004.

3. PUTTINI, R.S., ME, L., e SOUZA, R. T. de, *Certification and Authentication for Securing MANET Routing Protocols*.
4. IEEE Standard 802.11, *Wireless LAN media access control (MAC) and physical layer (PHY) specifications*, First edition, 1999-08-20
5. CORSON, S. e MARKER, J., *Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation consideration.* RFC 2501 (informational), IETF, 1999.
6. OGIER, R., LEWIS, M., TEMPLIN, F. e BELLUR, B., *Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), INTERNET DRAFT, MANET working group*, <draft-ietf-manet-tbrpf-06.txt>, November 2002.
7. CLAUSEN, T. e JACQUET, P. *Optimized Link State Routing Protocol, IETF Internet Draft*, MANET working group, version 11, Jul. 2003.
8. PERKINS,C. E., ROYER, E. M. e DAS, S. R.. *Ad hoc on-demand distance vector (AODV) routing. IETF INTERNET DRAFT*, MANET working group, Jan. 2002. draftietfmanetaodv10.txt.
9. JOHNSON, D. B. et al, *The dynamic source routing protocol for mobile ad hoc networks (DSR), INTERNET DRAFT*, MANET working group, < draft-ietf-manet-dsr-07.txt>, Feb. 2002.
10. PUTTINI, R.S., ME, L., e SOUZA, R. T. de, *An Authentication Protocol to MANET.*
11. BUIATI, F.M., PUTTINI, R.S. e SOUZA, R.T.J. de, *Secure Autoconfiguration for Móbile Ad Hoc*, 2nd International Information and telecommunication Technologies Symposium I2TS 2003.
12. LUO, H. AND LU, S. *Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks*. Technical Report TR-200030, Dept. of Computer Science, UCLA, 2000.
13. KONG, J., ZERFOS, P., LUO, H., LU, S. AND ZHANG, L., *Providing robust and ubiquitous security support for MANET*, IEEE ICNP 2001, 2001.
14. SHAMIR, A. *How to Share a Secret*. Communications of the ACM, 22(11):612-613, 1979.

SisBrAV – Brazilian Vulnerability Alert System

Robson de Oliveira Albuquerque, Daniel Silva Almendra, Leonardo Lobo Pulcineli, Rafael Timoteo de Sousa Junior, Claudia J. B. Abbas

Universidade de Brasília - Campus Universitário Darcy Ribeiro - Faculdade de Tecnologia - Depto de Engenharia Elétrica e Redes de Comunicação - Laboratório de Redes - sala B1 - CEP: 70910-900 - Brasília - DF – Brazil

Luis Javier Garcia Villalba

Universidad Complutense de Madrid (UCM) - Departamento de Sistemas Informáticos y Programación (DSIP) - Facultad de Informática, Despacho 431- C/ Profesor José García Santesmases s/n - Ciudad Universitaria - 28040 Madrid - Spain

Abstract. This paper describes the project and implementation of a vulnerability search and alert system based on free software. SisBrAV (acronym in Portuguese for Brazilian Vulnerability Alert System), will consist in a spider mechanism that explores several security-related sites for information on vulnerabilities and an intelligent interpreter responsible for analyzing and sorting the relevant data, feeding it into a database. With that information in hands, an email notifier sends out alerts, in Portuguese, about new vulnerabilities to registered clients, according to the operating systems and services run in their environment. In addition to the email notifier, a web server will also be implemented, for systems administrators to perform an on-demand custom search in the vulnerabilities database.

1 Introduction

In a daily basis, a large number of vulnerabilities, which affect a variety of systems and services, are detected. Manufacturers and developers work extensively in order to release, as fast as possible, a patch that fixes the problems found in their products. On the other hand, the hacker community is continually growing, producing malicious codes, exploits and viruses that take advantage of those vulnerabilities very rapidly. With the incredibly large quantity of information that can be found on the Internet today, as well as the increasing number of hacker sites that provide easy access to lots of malicious tools and exploit codes, it is of great importance that every enterprise's systems security team be well advised and informed about what are the threats to their environment and what they can do to avoid them, protecting their systems, services and network quickly and proactively. Even individuals with one or two PCs at home should be concerned with their system's vulnerabilities, applying the latest patches in their software, avoiding any security problem that may happen.

de Oliveira Albuquerque R., Silva Almendra D., Lobo Pulcineli L., Timoteo de Sousa Junior R., J. B. Abbas C. and Javier Garcia Villalba L. (2005).
SisBrAV – Brazilian Vulnerability Alert System.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 67-76
DOI: 10.5220/0002578700670076

Existing vulnerability database systems are created and maintained by human administrators, who are responsible for searching, analyzing and evaluating new vulnerabilities everyday, and then updating the database regularly with new entries, in a pretty much manual process. The initial idea of the project depicted in this paper, was that there could be an automatic process of gathering the relevant vulnerabilities information, sorting it according to predefined rules, feeding a database and generating email alerts for specific recipients whose environment could be affected. The solution should be based on free software and should also be portable to many platforms. SisBrAV project is, thus, the result of that idea.

2 Related Issues

Up to this date, in Brazil, there isn't any system such as SisBrAV, which automatically looks for new vulnerabilities and informs the users about them. In the other hand, a large number of security sites can be found in the Internet, and almost all of them have a vulnerability alert section, updated daily, enclosing vulnerabilities for many systems and programs. Thus, information regarding vulnerabilities can be easily accessed through the Internet, but it is very difficult to glimpse which vulnerabilities represent real threats among the large number encountered. So, there is plenty of information, but a lack of simplicity in the process of filtering these pieces of information, in order to keep only in the important ones.

The main challenge in the SisBrAV project is the sorting process, since the system will search for vulnerabilities in many sources, and each of them organizes the information in a particular way. The interpretation of the data collected must be very precise, as well as the sorting process, since the clients must be informed only about the threats to his specific environment. The importance score for each vulnerability must also be precisely assigned, making it possible for the client to assign different priorities when establishing security countermeasures for the vulnerabilities he has been informed about.

Two other elements are also critical for the efficiency of the SisBrAV system: the organization of the vulnerability information and the generation of customized alerts to each client according to his systems and services. The information must be sorted in an accurate but simple manner, and the alerts must be clear and succinct, as well as they must be sent only to the clients whose environment is threatened by the vulnerabilities.

SisBrAV will implement a module for each function it performs. The following section will describe how all these modules work and what functions they perform.

3 SisBrAV Modules

SisBrAV will be consisted of 5 modules. The Vulnerability Search Mechanism (VSM) module will consist in a spider that accesses and indexes many vulnerability documents in several security sites. The Interpreter, Parser and Sorter (IPS) module will be a program that analyses the data provided from the spider, defining priorities

and classifying the entries, according to predefined rules. The Central Database (CDB) module will store all vulnerability data, clients' info and keywords for English-Portuguese translation. The Email Notifier (E-Note) module will alert by email the registered clients about new/updated vulnerabilities specific to each client's environment. At last, the Vulnerability Web Server (VWS) module will be a server, accessible by any registered client, to perform an on-demand, customized vulnerability search in the Vulnerability Database. The details of each module are depicted in the next sections.

3.1 Vulnerability Search Mechanism

The vulnerability search and indexing process is made by a spider mechanism. A spider is a program that explores the Internet by retrieving a document and recursively retrieving some or all the documents that are referenced in it. It acts as an untiring human being who follows all links he finds in a web site, and all the links in the subsequent documents he sees. The spider indexes (fully or partially) all the documents that it accesses into a database, which can afterwards be used by a search engine.

The spider tool used in SisBrAV will be htdig, which is one of the programs that constitute the Ht://Dig package (6). Ht://Dig is a free web search engine, created in accordance to the GPU (General Public License) rules, and is consisted of many individual programs, like htdig, htdump and others.

The most recent stable version of Ht://Dig is 3.1.6, so this will probably be the version implemented in SisBrAV. A brief description of the htdig program is necessary, for there are some options which are used in the system, for its best performance and accuracy.

Htdig is a spider program (or search robot), which does what is called the "digging" process, retrieving HTML documents using the HTTP protocol, gathering information from these documents and indexing them, creating specific database files which can then be used to perform a search through these documents.

Htdig has many options, which are/will be used in the SisBrAV system, either to produce a desired result or for debugging purposes. The –c <configfile> option specifies another configuration file instead of the default. Another important option is the -h <maxhops> option, used to restrict the dig to documents that are at most maxhops links away from the starting document. This option is used every time the initial digging is run, to assure that htdig will index only the relevant documents for each site. The –i option is used to perform an initial digging. With this option, the old databases are removed, and new ones are created. There are also some options very useful for debugging, such as –s and –v, used to print statistics about the dig after completion and to set the verbose mode, respectively. For test purposes, one important option is the -t option, which tells htdig to create an ASCII version of the document database, making it easier to parse with other programs, so that information can be extracted from it for purposes other than searching. It generates the files db.docs and db.worddump, which formats will be explained later.

Finally, the url_file argument can also be passed, telling htdig to get the URLs to start indexing from the file provided, overriding the default start_url in the configuration file.

As said before, when using htdig with the –t option, it produces two ASCII files, db.docs and db.worddump. The db.docs file contains a series of lines, each of them relating to an indexed document. The lines contain information such as the document URL, title, modification time, size, meta description, excerpt, and many other useful document information. The db.wordlist file has a line for each word found in the indexed documents. Each line in this file contains a word, followed by the document ID where it was found, its location in the document, its weight, number of occurrences, etc.

The default configuration file for htdig is the file htdig.conf. That's where all configuration options for htdig (and the other tools, if they are used) are set. Since all of its parameters will probably be left with their default values, this file's content will not be copied in this paper. At first, the security sites indexed by SisBrAV's htdig will be the ones listed in the items (5), (7), (8), (9) and (10) in the References section. The number of sites can be (and will be) expanded to a much higher number, but initially only these five sites were chosen. The way htdig indexes each site will be almost the same: the only parameter that will differ from one site to another is the number of hops from the starting URL. For example, if maxhops is set to 2, htdig will index the starting URL, then it will follow all the links in that URL and index all the documents, and finally it will also follow the links in these documents, indexing the documents it finds, and then stop the digging process. Since each site has its way of displaying their documents, the number of hops necessary to gather all relevant vulnerability information will vary from site to site.

To solve this issue, a simple UNIX bash script will be used to read a file that contains lines with an URL and a number (which defines the maximum hops from the initial URL), separated by a TAB. The script will produce different htdig commands, according to the number of maximum hops defined. The number of maximum hops for each site is defined by the SisBrAV administrators, who inspect the sites and check the number of levels the spider will have to crawl down in order to obtain the maximum amount of relevant information about the vulnerabilities, and the minimum unnecessary information.

Htdig generates several Berkeley DB type files. These files will then be analyzed by the IPS Module, as explained in the next section.
by the IPS Module, as explained in the next section.

3.2 Interpreter, Parser and Sorter

The IPS Module will probably be written in Java. It will use an heuristics algorithm to perform the content analysis of the data stored in the Berkeley DB files created by htdig, in order to feed the Central Database with accurate vulnerability information. The data is parsed and the vulnerabilities are grouped between previously determined, hierarchically distributed classes.

At first, the IPS program will perform the sorting process. Initially, it analyses all the entries in the database, to find ambiguous or duplicated information for a same vulnerability. Then, it parses the content of the information, in order to group the vulnerability entries in classes, according to its main aspects: remote/local, type, low/medium/high importance score, etc. It also determines the systems/services in which that vulnerability occurs. If there is more than one entry for the same vulnerability, it correlates all the information found in the entries, to make sure the attributes are set as precisely as possible. For example, if a given vulnerability is

issued in three different sites, and one of them scores the vulnerability as of medium importance and the others say its importance is high, the IPS will set this attribute to "high".

The hierarchical class tree used to group the vulnerabilities is described in figure 1.

Each document indexed by the spider in the VSM module will be related to a specific vulnerability. The IPS module performs the vulnerability sorting process for each document, by following the tree shown in the above figure. Initially, the algorithm determines if the vulnerability is local or remote, according to the information found in the document. It then classifies the vulnerability into a specific vulnerability type, among the predefined types registered in the system, such as Denial of Service, Buffer Overflow, Password Retrieval, Authentication Bypass, etc. Afterwards, an importance score is assigned to the vulnerability. At last, the IPS finds out what operating systems – and their versions – are affected by the vulnerability, and what programs/services – and their versions – are threatened by it. As well as the vulnerability types, there will also be a large list of systems and services (and their respective versions), which IPS will use in the sorting process.

After a given vulnerability is sorted, the IPS checks if there is any other vulnerability with exactly the same characteristics, affecting the same systems/services. If so, it performs a series of tests, to check if both entries refer to the same vulnerabilities. In these tests, other information is analyzed, such as the vulnerability date, the document URL (if the root site is the same, it's probably not the same vulnerability, since a security site must not have duplicated documents for the same vulnerability), and other information.

After the vulnerabilities have been classified, the IPS feeds the Central Database with that data.

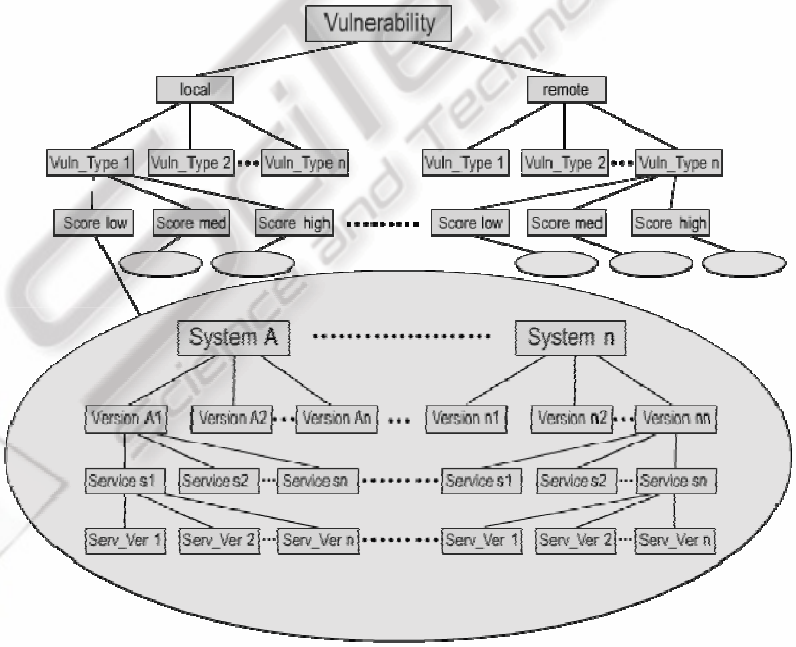


Fig. 1. IPS – Hierarchical Classes Tree: the vulnerabilities are sorted and grouped into different classes

Since the database is not hierarchical, but relational, the IPS will also have to convert the results of the sorting process before actually feeding the Central Database.

3.3 Central Database

In order to store all the information regarding vulnerabilities and their attributes, clients' profiles, systems and services data, as well as the English-Portuguese translation data, SisBrAV will have a Central Database. It is most likely that it will be implemented using a MySQL server, which is GPU compliant, and its architecture will follow the SQL ANSI standard, to guarantee its portability and scalability.

The CDB will be divided into three smaller databases, each one storing specific information, although the three of them relate to each other. The first database is the Vulnerability Database, which will contain all the vulnerability information already sorted into defined groups, as seen in the IPS section. The second base is the Client Database, which will keep the client-related data, such as their names, contact information and the systems and software running in their environment. At last, the third database will be the English-Portuguese Translation Database, storing a number of keywords, each one relating to keywords in the other idiom, according to certain parameters.

Mostly based on the schema designed by the Open Source Vulnerability Database Team (5), the Vulnerability Database is the most important part of the whole SisBrAV system, for it is the central repository of all vulnerability information. Its structure, which is still being developed, will probably keep the main OSVDB structure, although there will be some changes in certain tables, and other tables will be removed or added. The Vulnerability Database, when fully implemented, will be similar to figure 2.

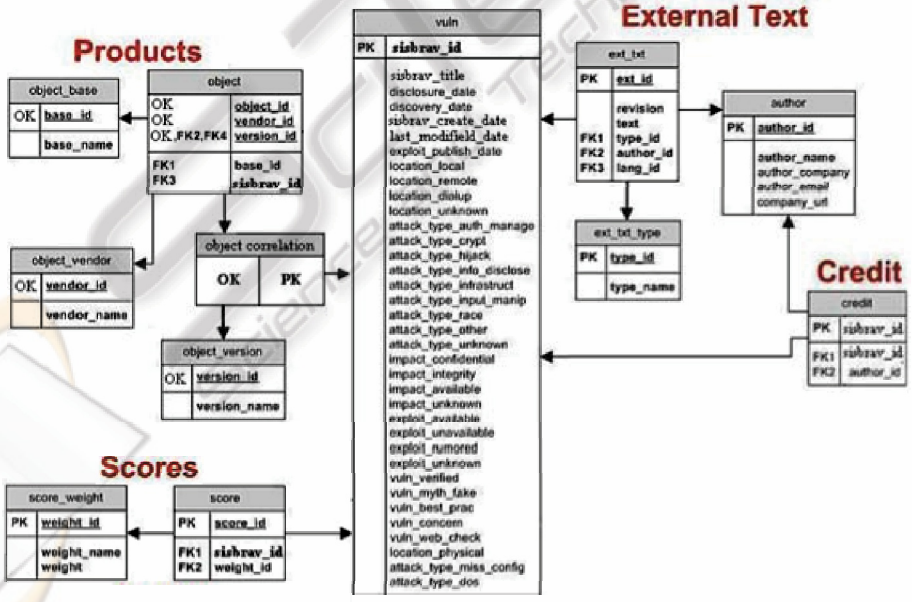


Fig. 2. Vulnerability Database Schema

The External Text section in this database consists in tables that describe certain aspects about a vulnerability. They exist inside the database, but usually describe information that one would use externally to the database. For example, a Solution Description, or a Vulnerability Description is an external text.

The tables in the above schema also deserve some explanation. The vuln table is the main table in the schema. It's where the SisBrAV IDs live. Other information stored in this table includes various dates and vulnerability classification data. The ext_txt_type table defines the types of external texts. For example, Vulnerability Description, Solution Description, Technical Description, Manual Testing Notes. The ext_txt table stores the external text blobs for any type of text that is larger than 1024 characters. Other information stored is the language, type, author, and revision. When the texts are updated/fixed/modified the new text is reinserted into this table and the revision number is incremented. The contributors for anything in the ext_txt table are identified in the author table, making it possible to have a contributor's line to any SisBrAV ID. The authors are used to track the external text authors, as well as the credited researcher of each vulnerability. In the Products section, the object correlation table performs a link between the PK of the vuln table and a key named Object Key (OK).

As a result, it is possible for other tables to link to the Products tables without using a PK. The object table binds vendor, base, version and vulnerability together, storing product information.

The name object might seem sort of vague, but it means the object that the vulnerability exists within. The object_base table contains product names. For example, Windows, Exchange, Apache, and MySQL are all examples of product names. The object_vendor table contains the vendor names. For example, Microsoft, Sun Microsystems, and Apache Software Foundation are all examples of vendor names. The object_version table contains the version names. For example, 1.0, 2.0, 0.1, XP, 2000, or 95 are all examples of version names. Another crucial table is the score table, used to bind a scoring weight to a vulnerability. It is intended to allow every vulnerability in the database to be associated with one scoring weight. The score_weight table is used to store any type of scoring information needed for scoring calculations. Finally, the credit table adds support for identifying credit for discovering a vulnerability. Instead of storing author like information, a reference to the author table is made, as the data is extremely similar.

The second part of the CDB is the Client Database, responsible for storing all client-related data, involving personal/enterprise identification information, contact emails, products (systems and services) running, etc. Its structure is shown in figure 3.

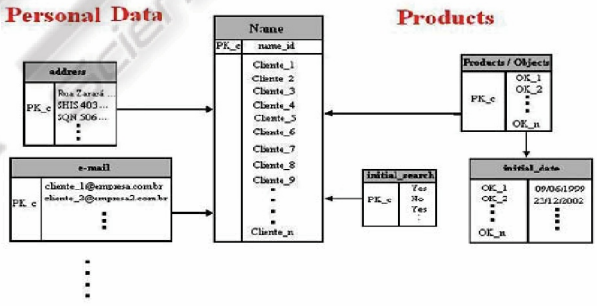


Fig. 3. Client Database Schema

In the above schema, the Products section refers to the products that each client registers, in order to receive only vulnerability alerts related to the systems running in his/her environment. The initial date to look in for vulnerabilities is also stored for each client. Specific product characteristics data is kept in the Vulnerability Database, as it was shown in the previously. All data related to a client himself is found in the Personal Data section. Contact emails, telephone numbers, addresses and personal/enterprise information, as well as the clients' passwords are entered in this part of the database.

The Name table consists in the main table of the Client Database, containing each client's account ID. All the clients are bound to their products through the Products/Objects table. The initial_date table stores the initial date to search for vulnerabilities, for each product a client registers in the database, while the Initial_search table contains entries that specify if a client is a new registered client in the system (represented by a "Yes" entry) or not ("No"). These entries are used by E-Note, to define if an initial search must be executed or not. At last, the Personal Data tables, such as address, e-mail and others, store client specific information, such as email, telephone numbers, address, personal/enterprise information and login username and password.

The last subpart of the CDB is the English-Portuguese Translation Database, which is still being designed. It will contain a large number of keywords in English and in Portuguese, in addition to semantics and syntax rules, making it possible for the E-Note module to translate the main description of a vulnerability entry to compose a mainly Portuguese email alert.

3.4 Email Notifier (E-Note)

This program will look for updated vulnerability information in the database. After retrieving the information, the program checks, for each registered client, if there are any new/updated vulnerabilities which affect the client's environment. If so, an email message – in Portuguese – is formatted, to inform the client about the new vulnerabilities discovered in his systems and services.

This message consists in a brief explanation of the vulnerability, in Portuguese, and one or more links for further information on that issue.

When a client registers in the SisBrAV system, he will have to inform what systems he has and what programs he runs, thus defining the scope of vulnerabilities SisBrAV should be concerned with, when generating alerts to that specific client. Besides that, the client also defines the start date, determining the initial point from which the system should begin the search in the vulnerability database. With that data in hands, E-Note will search in the database only the information that is really necessary for that client, generating a customized email message to him.

The E-Note module will also be written in Java, to guarantee its portability. E-Note is divided in two programs: one program performs the search in the database and the other sends the email alert.

For each new client added in the system, all the data about his systems and services is stored in the clients table, in the SisBrAV database, and a flag is set for this client, with a logical value that represents "NEW". The start date from which he wants to be informed about existing vulnerabilities is also stored in the database clients table.

Every time E-Note is run, it checks if there are any new clients in order to search for all the vulnerability entries that occur specifically in their systems and are newer than the start date defined by the client. It then generates the email alert to those clients, notifying about all vulnerabilities found. Afterwards, the "NEW" flag in the clients' entry in the database is set to a value that stands for "OLD".

For existing clients, the E-Note will simply check if there are new/updated vulnerabilities regarding their systems/services. If so, it generates the email alert for the specific clients whose systems are affected.

Due to the fact that the vulnerability information stored in the database is mainly in English, the vulnerabilities selected by E-Note are also in English. To make it possible for E-Note to generate Portuguese messages, an English-Portuguese translation database will bind English keywords to previously defined Portuguese sentences. E-Note performs, thus, a simple translation in the main vulnerability description. The main aspects – remote/local, high/low importance, etc – of the vulnerability are also translated. For example, if the main description of a vulnerability is "HP-UX DCE Remote Denial of Service Vulnerability", and its importance is critical, the Portuguese message would be "HP-UX DCE: Vulnerabilidade Remota de Negação de Serviço. Importância: Crítica". The translation database is in the format described in the previous section.

Along with the main description of the vulnerability, the email also contains links to the sites where that vulnerability is described and discussed.

3.5 Vulnerability Web Server

The idea of SisBrAV is not only to inform its users, emailing them alerts about vulnerability issues. The registered clients will also be able to perform a custom search in the Vulnerability Database through the web. With that functionality in mind, the fifth module of SisBrAV will be a Web Server that will handle these web requests.

The users will access an authentication site, where they provide their username and password (which are created and informed to him/her during the registering process). If successfully authenticated, they will be redirected to a customized database search page.

The site interface is being designed to be friendly and simple, although its security will be fundamental. The web site will probably be based in PHP, due to the fact that this language is very portable, and through its use, the database access can be implemented in a secure and simple manner. The web server chosen for the SisBrAV system was Apache, mainly because it is a multi-platform server, and also because fully supports the web publishing technology which will probably be used (PHP).

There are also other technologies which utilization is currently in discussion, such as Java servlets or JSP, because through using it would be easier to integrate the VWS module to the other modules in SisBrAV. XML is also in discussion, since it is another efficient way of implementing the database access from web. If JSP ends up being implemented, Tomcat (which is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies, fully integrated with Apache) will also be used.

4 Conclusions

In the current scenario, it is really important for anyone connected to the World Wide Web to protect his/her systems and data against the threats that continually arise. Besides having a nice antivirus tool, a firewall efficiently configured and other security technologies implemented in their network, users and enterprises must keep all of their Operating Systems, services and other software up-to-date, by applying all their latest patches and fixes. With that in mind, it's of great importance that systems and network administrators be informed quickly about any vulnerability that may be encountered in their systems, so that they can act proactively to build up defense countermeasures to guarantee the security of their environment.

SisBrAV will be an important security innovation, since it implements an idea of an automatic vulnerability searching and alerting mechanism, with very little human administration needed. Since it will have many trustable security sites as sources where it will look for vulnerabilities information, SisBrAV will be a very reliable system, extending the horizons of systems and network security. In addition to that features, it is also important to remember SisBrAV, being a Brazilian project, will implement a translation feature in order to produce Portuguese email alerts, so that Brazilian clients will feel comfortable with it. In the future, the language support can be expanded to other idioms.

Nowadays, where free software gradually gains space in the software business, a program must support many platforms, so that it can be installed in a variety of systems and interact with different technologies without incurring into stability loss or performance troubles. SisBrAV is being designed using only free software products and platform independent languages, resulting in a solution with great portability and scalability.

References

1. Deitel, H. M. – *Java, Como Programar* / H. M. Deitel e P. J. Deitel; trad. Carlos Arthur. Lang Lisboa. – 4.ed. – Porto Alegre: Bookman, 2003.
2. *SQL Tutorial*. Available from: http://www.w3schools.com/sql.
3. *PHP/MySQL* Tutorial. Available from: http://www.freewebmasterhelp.com/tutorials/phpmy
4. *Portal Java Home Page*. Available from: http://www.portaljava.com/home/index.php.
5. *Open Source Vulnerability Database*. Available from: http://www.osvdb.org.
6. *http://Dig Project Home Page*. Available from: http://www.htdig.org.
7. *Internet Security Systems X-force Home Page*. Available from: http://xforce.iss.net.
8. *Cert Knowledge Base*. Available from: http://www.cert.org/kb.
9. *SANS Newsletters*. Available from: http://www.sans.org/newsletters.
10. *Security Focus Home Page*. Available from: http://www.securityfocus.com.

Proposal of a System for Searching and Indexing Heterogeneous Vulnerabilities Databases

Robson de Oliveira[1], Fabio Buiati[1], Luis Javier García Villalba[1], Daniel Almendra[2], L. Pulcineli[2], Rafael de Sousa Jr.[2], and Cláudia Jacy Barenco Abbas[2]

[1] Grupo de Análisis, Seguridad y Sistemas (GASS),
Departamento de Sistemas Informáticos y Programación (DSIP),
Facultad de Informática , Universidad Complutense de Madrid (UCM),
C/ Profesor José García Santesmases s/n, Ciudad Universitaria, 28040 Madrid, Spain
{robson, fabio}@fdi.ucm.es, javiergv@sip.ucm.es
[2] Universidade de Brasília, Campus Universitário Darcy Ribeiro,
Faculdade de Tecnologia, Depto. de Engenharia Elétrica e Redes de Comunicação,
Laboratório de Redes - sala B1, CEP: 70910-900 - Brasília - DF – Brazil
{danielalmendra, pulcineli}@terra.com.br, desousa@unb.br,
barenco@redes.unb.br

Abstract. This paper describes the project and implementation of a vulnerability search and alert system based on free software. SisBrAV (acronym in Portuguese for Brazilian Vulnerability Alert System), will consist in a spider mechanism that explores several security-related sites for information on vulnerabilities and an intelligent interpreter responsible for analyzing and sorting the relevant data, feeding it into a database. With that information in hands, an email notifier sends out alerts, in Portuguese, about new vulnerabilities to registered clients, according to the operating systems and services run in their environment. In addition to the email notifier, a web server will also be implemented, for systems administrators to perform an on-demand custom search in the vulnerabilities database.

Keywords: Vulnerability Database, Internet spiders, Security Information System, Open Source Solution.

1 Introduction

In a daily basis, a large number of vulnerabilities, which affect a variety of systems and services, are detected. Manufacturers and developers work extensively in order to release, as fast as possible, a patch that fixes the problems found in their products. On the other hand, the hacker community is continually growing, producing malicious codes, exploits and viruses that take advantage of those vulnerabilities very rapidly. With the incredibly large quantity of information that can be found on the Internet today, as well as the increasing number of hacker sites that provide easy access to lots of malicious tools and exploit codes, it is of great importance that every enterprise's systems security team be well advised and informed about what are the threats to their environment and what they can do to avoid them, protecting their systems, services and network quickly and proactively. Even individuals with one or two PCs at home

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3984, pp. 819 – 828, 2006.

should be concerned with their system's vulnerabilities, applying the latest patches in their software, avoiding any security problem that may happen.

Existing vulnerability database systems are created and maintained by human administrators, who are responsible for searching, analyzing and evaluating new vulnerabilities everyday, and then updating the database regularly with new entries, in a pretty much manual process. The initial idea of the project depicted in this paper, was that there could be an automatic process of gathering the relevant vulnerabilities information, sorting it according to predefined rules, feeding a database and generating email alerts for specific recipients whose environment could be affected. The solution should be based on free software and should also be portable to many platforms. SisBrAV project is, thus, the result of that idea.

2 Related Issues

Up to this date, in Brazil, there isn't any system such as SisBrAV, which automatically looks for new vulnerabilities and informs the users about them. In the other hand, a large number of security sites can be found in the Internet, and almost all of them have a vulnerability alert section, updated daily, enclosing vulnerabilities for many systems and programs. Thus, information regarding vulnerabilities can be easily accessed through the Internet, but it is very difficult to glimpse which vulnerabilities represent real threats among the large number encountered. So, there is plenty of information, but a lack of simplicity in the process of filtering these pieces of information, in order to keep only in the important ones.

The main challenge in the SisBrAV project is the sorting process, since the system will search for vulnerabilities in many sources, and each of them organizes the information in a particular way. The interpretation of the data collected must be very precise, as well as the sorting process, since the clients must be informed only about the threats to his specific environment. The importance score for each vulnerability must also be precisely assigned, making it possible for the client to assign different priorities when establishing security countermeasures for the vulnerabilities he has been informed about.

Two other elements are also critical for the efficiency of the SisBrAV system: the organization of the vulnerability information and the generation of customized alerts to each client according to his systems and services. The information must be sorted in an accurate but simple manner, and the alerts must be clear and succinct, as well as they must be sent only to the clients whose environment is threatened by the vulnerabilities.

SisBrAV will implement a module for each function it performs. The following section will describe how all these modules work and what functions they perform.

3 SisBrAV Modules

SisBrAV will be consisted of 5 modules. The Vulnerability Search Mechanism (VSM) module will consist in a spider that accesses and indexes many vulnerability documents in several security sites. The Interpreter, Parser and Sorter (IPS) module

will be a program that analyses the data provided from the spider, defining priorities and classifying the entries, according to predefined rules. The Central Database (CDB) module will store all vulnerability data, clients' info and keywords for English-Portuguese translation. The Email Notifier (E-Note) module will alert by email the registered clients about new/updated vulnerabilities specific to each client's environment. At last, the Vulnerability Web Server (VWS) module will be a server, accessible by any registered client, to perform an on-demand, customized vulnerability search in the Vulnerability Database. The details of each module are depicted in the next sections.

3.1 Vulnerability Search Mechanism (VSM)

The vulnerability search and indexing process is made by a spider mechanism. A spider is a program that explores the Internet by retrieving a document and recursively retrieving some or all the documents that are referenced in it. It acts as an untiring human being who follows all links he finds in a web site, and all the links in the subsequent documents he sees. The spider indexes (fully or partially) all the documents that it accesses into a database, which can afterwards be used by a search engine. The spider tool used in SisBrAV will be htdig, which is one of the programs that constitute the Ht://Dig package (6). Ht://Dig is a free web search engine, created in accordance to the GPU (General Public License) rules, and is consisted of many individual programs, like htdig, htdump and others. The most recent stable version of Ht://Dig is 3.1.6, so this will probably be the version implemented in SisBrAV. A brief description of the htdig program is necessary, for there are some options which are used in the system, for its best performance and accuracy.

Htdig is a spider program (or search robot), which does what is called the "digging" process, retrieving HTML documents using the HTTP protocol, gathering information from these documents and indexing them, creating specific database files which can then be used to perform a search through these documents.

Htdig has many options, which are/will be used in the SisBrAV system, either to produce a desired result or for debugging purposes. The –c <configfile> option specifies another configuration file instead of the default. Another important option is the -h <maxhops> option, used to restrict the dig to documents that are at most maxhops links away from the starting document. This option is used every time the initial digging is run, to assure that htdig will index only the relevant documents for each site. The –i option is used to perform an initial digging. With this option, the old databases are removed, and new ones are created. There are also some options very useful for debugging, such as –s and –v, used to print statistics about the dig after completion and to set the verbose mode, respectively. For test purposes, one important option is the -t option, which tells htdig to create an ASCII version of the document database, making it easier to parse with other programs, so that information can be extracted from it for purposes other than searching. It generates the files db.docs and db.worddump, which formats will be explained later. Finally, the url_file argument can also be passed, telling htdig to get the URLs to start indexing from the file provided, overriding the default start_url in the configuration file.

As said before, when using htdig with the –t option, it produces two ASCII files, db.docs and db.worddump. The db.docs file contains a series of lines, each of them

relating to an indexed document. The lines contain information such as the document URL, title, modification time, size, meta description, excerpt, and many other useful document information. The db.wordlist file has a line for each word found in the indexed documents. Each line in this file contains a word, followed by the document ID where it was found, its location in the document, its weight, number of occurrences, etc. The default configuration file for htdig is the file htdig.conf. That's where all configuration options for htdig (and the other tools, if they are used) are set. Since all of its parameters will probably be left with their default values, this file's content will not be copied in this paper. At first, the security sites indexed by SisBrAV's htdig will be the ones listed in the items (5), (7), (8), (9) and (10) in the References section. The number of sites can be (and will be) expanded to a much higher number, but initially only these five sites were chosen. The way htdig indexes each site will be almost the same: the only parameter that will differ from one site to another is the number of hops from the starting URL. For example, if maxhops is set to 2, htdig will index the starting URL, then it will follow all the links in that URL and index all the documents, and finally it will also follow the links in these documents, indexing the documents it finds, and then stop the digging process. Since each site has its way of displaying their documents, the number of hops necessary to gather all relevant vulnerability information will vary from site to site.

To solve this issue, a simple UNIX bash script will be used to read a file that contains lines with an URL and a number (which defines the maximum hops from the initial URL), separated by a TAB. The script will produce different htdig commands, according to the number of maximum hops defined. The number of maximum hops for each site is defined by the SisBrAV administrators, who inspect the sites and check the number of levels the spider will have to crawl down in order to obtain the maximum amount of relevant information about the vulnerabilities, and the minimum unnecessary information. Htdig generates several Berkeley DB type files. These files will then be analyzed by the IPS Module, as explained in the next section.

3.2 Interpreter, Parser and Sorter (IPS)

The IPS Module will probably be written in Java. It will use an heuristics algorithm to perform the content analysis of the data stored in the Berkeley DB files created by htdig, in order to feed the Central Database with accurate vulnerability information. The data is parsed and the vulnerabilities are grouped between previously determined, hierarchically distributed classes. At first, the IPS program will perform the sorting process. Initially, it analyses all the entries in the database, to find ambiguous or duplicated information for a same vulnerability. Then, it parses the content of the information, in order to group the vulnerability entries in classes, according to its main aspects: remote/local, type, low/medium/high importance score, etc. It also determines the systems/services in which that vulnerability occurs. If there is more than one entry for the same vulnerability, it correlates all the information found in the entries, to make sure the attributes are set as precisely as possible. For example, if a given vulnerability is issued in three different sites, and one of them scores the vulnerability as of medium importance and the others say its importance is high, the IPS will set this attribute to "high". The hierarchical class tree used to group the vulnerabilities is described in Fig. 1. Each document indexed by the spider in the

VSM module will be related to a specific vulnerability. The IPS module performs the vulnerability sorting process for each document, by following the tree shown in the above figure. Initially, the algorithm determines if the vulnerability is local or remote, according to the information found in the document. It then classifies the vulnerability into a specific vulnerability type, among the predefined types registered in the system, such as Denial of Service, Buffer Overflow, Password Retrieval, Authentication Bypass, etc. Afterwards, an importance score is assigned to the vulnerability. At last, the IPS finds out what operating systems – and their versions – are affected by the vulnerability, and what programs/services – and their versions – are threatened by it. As well as the vulnerability types, there will also be a large list of systems and services (and their respective versions), which IPS will use in the sorting process.

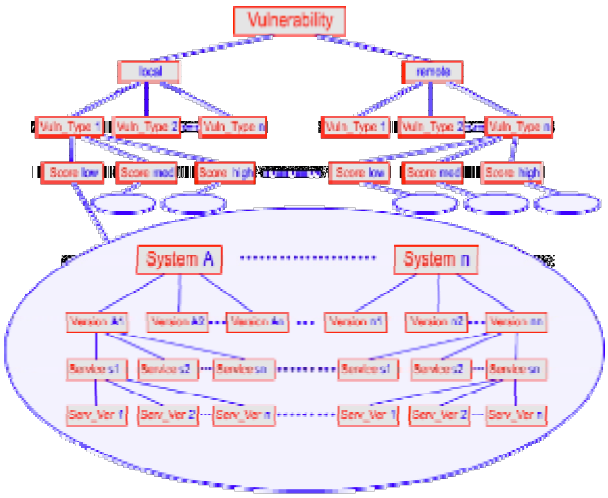


Fig. 1. IPS – Hierarchical Classes Tree

After a given vulnerability is sorted, the IPS checks if there is any other vulnerability with exactly the same characteristics, affecting the same systems/services. If so, it performs a series of tests, to check if both entries refer to the same vulnerabilities. In these tests, other information is analyzed, such as the vulnerability date, the document URL (if the root site is the same, it's probably not the same vulnerability, since a security site must not have duplicated documents for the same vulnerability), and other information. After the vulnerabilities have been classified, the IPS feeds the Central Database with that data. Since the database is not hierarchical, but relational, the IPS will also have to convert the results of the sorting process before actually feeding the Central Database.

3.3 Central Database (CDB)

In order to store all the information regarding vulnerabilities and their attributes, clients' profiles, systems and services data, as well as the English-Portuguese translation data, SisBrAV will have a Central Database. It is most likely that it will be

implemented using a MySQL server, which is GPU compliant, and its architecture will follow the SQL ANSI standard, to guarantee its portability and scalability. The CDB will be divided into three smaller databases, each one storing specific information, although the three of them relate to each other. The first database is the Vulnerability Database, which will contain all the vulnerability information already sorted into defined groups, as seen in the IPS section. The second base is the Client Database, which will keep the client-related data, such as their names, contact information and the systems and software running in their environment. At last, the third database will be the English-Portuguese Translation Database, storing a number of keywords, each one relating to keywords in the other idiom, according to certain parameters. Mostly based on the schema designed by the Open Source Vulnerability Database Team (5), the Vulnerability Database is the most important part of the whole SisBrAV system, for it is the central repository of all vulnerability information. Its structure, which is still being developed, will probably keep the main OSVDB structure, although there will be some changes in certain tables, and other tables will be removed or added. The Vulnerability Database, when fully implemented, will be similar to Fig. 2.

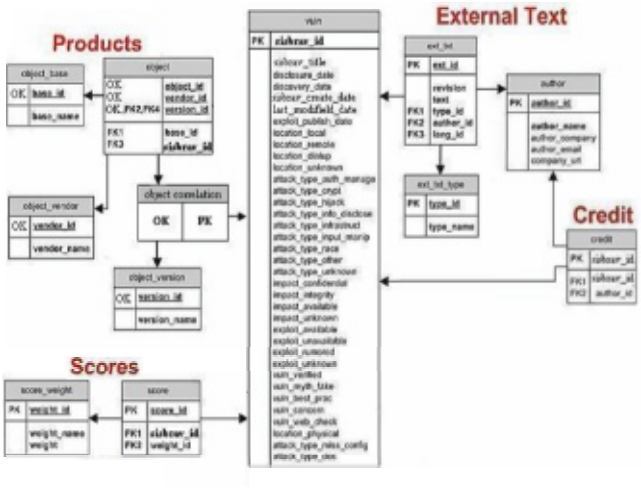


Fig. 2. Vulnerability Database Schema

The External Text section in this database consists in tables that describe certain aspects about a vulnerability. They exist inside the database, but usually describe information that one would use externally to the database. For example, a Solution Description, or a Vulnerability Description is an external text.

The tables in the above schema also deserve some explanation. The vuln table is the main table in the schema. It's where the SisBrAV IDs live. Other information stored in this table includes various dates and vulnerability classification data. The ext_txt_type table defines the types of external texts. For example, Vulnerability Description, Solution Description, Technical Description, Manual Testing Notes. The ext_txt table stores the external text blobs for any type of text that is larger than 1024

characters. Other information stored is the language, type, author, and revision. When the texts are updated/fixed/modified the new text is reinserted into this table and the revision number is incremented. The contributors for anything in the ext_txt table are identified in the author table, making it possible to have a contributor's line to any SisBrAV ID. The authors are used to track the external text authors, as well as the credited researcher of each vulnerability. In the Products section, the object correlation table performs a link between the PK of the vuln table and a key named Object Key (OK). As a result, it is possible for other tables to link to the Products tables without using a PK. The object table binds vendor, base, version and vulnerability together, storing product information. The name object might seem sort of vague, but it means the object that the vulnerability exists within. The object_base table contains product names. For example, Windows, Exchange, Apache, and MySQL are all examples of product names. The object_vendor table contains the vendor names. For example, Microsoft, Sun Microsystems, and Apache Software Foundation are all examples of vendor names. The object_version table contains the version names. For example, 1.0, 2.0, 0.1, XP, 2000, or 95 are all examples of version names. Another crucial table is the score table, used to bind a scoring weight to a vulnerability. It is intended to allow every vulnerability in the database to be associated with one scoring weight. The score_weight table is used to store any type of scoring information needed for scoring calculations. Finally, the credit table adds support for identifying credit for discovering a vulnerability. Instead of storing author like information, a reference to the author table is made, as the data is extremely similar.

The second part of the CDB is the Client Database, responsible for storing all client-related data, involving personal/enterprise identification information, contact emails, products (systems and services) running, etc. Its structure is shown in Fig. 3.

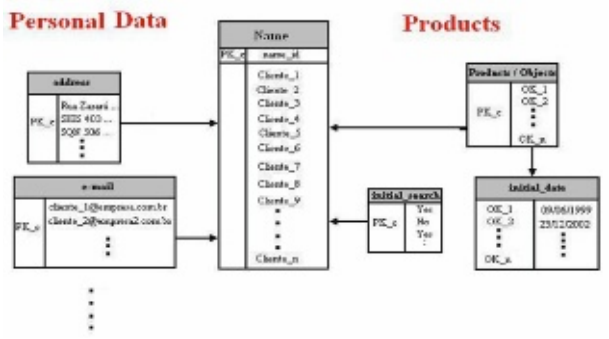


Fig. 3. Client Database Schema

In the above schema, the Products section refers to the products that each client registers, in order to receive only vulnerability alerts related to the systems running in his/her environment. The initial date to look in for vulnerabilities is also stored for each client. Specific product characteristics data is kept in the Vulnerability Database, as it was shown in the previously. All data related to a client himself is found in the

Personal Data section. Contact emails, telephone numbers, addresses and personal/ enterprise information, as well as the clients' passwords are entered in this part of the database.

The Name table consists in the main table of the Client Database, containing each client's account ID. All the clients are bound to their products through the Products/Objects table. The initial_date table stores the initial date to search for vulnerabilities, for each product a client registers in the database, while the Initial_search table contains entries that specify if a client is a new registered client in the system (represented by a "Yes" entry) or not ("No"). These entries are used by E-Note, to define if an initial search must be executed or not. At last, the Personal Data tables, such as address, e-mail and others, store client specific information, such as email, telephone numbers, address, personal/enterprise information and login username and password.

The last subpart of the CDB is the English-Portuguese Translation Database, which is still being designed. It will contain a large number of keywords in English and in Portuguese, in addition to semantics and syntax rules, making it possible for the E-Note module to translate the main description of a vulnerability entry to compose a mainly Portuguese email alert.

3.4 Email Notifier (E-Note)

This program will look for updated vulnerability information in the database. After retrieving the information, the program checks, for each registered client, if there are any new/updated vulnerabilities which affect the client's environment. If so, an email message – in Portuguese – is formatted, to inform the client about the new vulnerabilities discovered in his systems and services. This message consists in a brief explanation of the vulnerability, in Portuguese, and one or more links for further information on that issue.

When a client registers in the SisBrAV system, he will have to inform what systems he has and what programs he runs, thus defining the scope of vulnerabilities SisBrAV should be concerned with, when generating alerts to that specific client. Besides that, the client also defines the start date, determining the initial point from which the system should begin the search in the vulnerability database. With that data in hands, E-Note will search in the database only the information that is really necessary for that client, generating a customized email message to him.

The E-Note module will also be written in Java, to guarantee its portability. E-Note is divided in two programs: one program performs the search in the database and the other sends the email alert.

For each new client added in the system, all the data about his systems and services is stored in the clients table, in the SisBrAV database, and a flag is set for this client, with a logical value that represents "NEW". The start date from which he wants to be informed about existing vulnerabilities is also stored in the database clients table.

Every time E-Note is run, it checks if there are any new clients in order to search for all the vulnerability entries that occur specifically in their systems and are newer than the start date defined by the client. It then generates the email alert to those clients, notifying about all vulnerabilities found. Afterwards, the "NEW" flag in the clients' entry in the database is set to a value that stands for "OLD". For existing

clients, the E-Note will simply check if there are new/updated vulnerabilities regarding their systems/services. If so, it generates the email alert for the specific clients whose systems are affected.

Due to the fact that the vulnerability information stored in the database is mainly in English, the vulnerabilities selected by E-Note are also in English. To make it possible for E-Note to generate Portuguese messages, an English-Portuguese translation database will bind English keywords to previously defined Portuguese sentences. E-Note performs, thus, a simple translation in the main vulnerability description. The main aspects – remote/local, high/low importance, etc – of the vulnerability are also translated. For example, if the main description of a vulnerability is "HP-UX DCE Remote Denial of Service Vulnerability", and its importance is critical, the Portuguese message would be "HP-UX DCE: Vulnerabilidade Remota de Negação de Serviço. Importância: Crítica". The translation database is in the format described in the previous section.

Along with the main description of the vulnerability, the email also contains links to the sites where that vulnerability is described and discussed.

3.5 Vulnerability Web Server (VWS)

The idea of SisBrAV is not only to inform its users, emailing them alerts about vulnerability issues. The registered clients will also be able to perform a custom search in the Vulnerability Database through the web. With that functionality in mind, the fifth module of SisBrAV will be a Web Server that will handle these web requests. The users will access an authentication site, where they provide their username and password (which are created and informed to him/her during the registering process). If successfully authenticated, they will be redirected to a customized database search page. The site interface is being designed to be friendly and simple, although its security will be fundamental. The web site will probably be based in PHP, due to the fact that this language is very portable, and through its use, the database access can be implemented in a secure and simple manner. The web server chosen for the SisBrAV system was Apache, mainly because it is a multi-platform server, and also because fully supports the web publishing technology which will probably be used (PHP). There are also other technologies which utilization is currently in discussion, such as Java servlets or JSP, because through using it would be easier to integrate the VWS module to the other modules in SisBrAV. XML is also in discussion, since it is another efficient way of implementing the database access from web. If JSP ends up being implemented, Tomcat (which is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies, fully integrated with Apache) will also be used.

4 Conclusions

In the current scenario, it is really important for anyone connected to the World Wide Web to protect his/her systems and data against the threats that continually arise. Besides having a nice antivirus tool, a firewall efficiently configured and other security technologies implemented in their network, users and enterprises must keep

all of their Operating Systems, services and other software up-to-date, by applying all their latest patches and fixes. With that in mind, it's of great importance that systems and network administrators be informed quickly about any vulnerability that may be encountered in their systems, so that they can act proactively to build up defense countermeasures to guarantee the security of their environment. SisBrAV will be an important security innovation, since it implements an idea of an automatic vulnerability searching and alerting mechanism, with very little human administration needed. Since it will have many trustable security sites as sources where it will look for vulnerabilities information, SisBrAV will be a very reliable system, extending the horizons of systems and network security. In addition to that features, it is also important to remember SisBrAV, being a Brazilian project, will implement a translation feature in order to produce Portuguese email alerts, so that Brazilian clients will feel comfortable with it. In the future, the language support can be expanded to other idioms. Nowadays, where free software gradually gains space in the software business, a program must support many platforms, so that it can be installed in a variety of systems and interact with different technologies without incurring into stability loss or performance troubles. SisBrAV is being designed using only free software products and platform independent languages, resulting in a solution with great portability and scalability.

Acknowledgements

GASS's work is supported by the Spanish Ministry of Education and Science (MEC) under project TSI2005-00986.

References

1. Deitel, H. M. – Java, Como Programar / H. M. Deitel e P. J. Deitel; trad. Carlos Arthur. Lang Lisboa. – 4.ed. – Porto Alegre: Bookman, 2003.
2. SQL Tutorial. Available from: http://www.w3schools.com/sql.
3. PHP/MySQL Tutorial. Available from: http://www.freewebmasterhelp.com/tutorials/phpmysql.
4. Portal Java Home Page. Available from: http://www.portaljava.com/home/index.php.
5. Open Source Vulnerability Database. Available from: http://www.osvdb.org.
6. Ht://Dig Project Home Page. Available from: http://www.htdig.org.
7. Internet Security Systems X-force Home Page. Available from: http://xforce.iss.net.
8. Cert Knowledge Base. Available from: http://www.cert.org/kb.
9. SANS Newsletters. Available from: http://www.sans.org/newsletters.
10. Security Focus Home Page. Available from: http://www.securityfocus.com.

Enhancing an Integer Challenge-Response Protocol

Robson de Oliveira Albuquerque[1], Luis Javier García Villalba[1], and Rafael Timóteo de Sousa Jr.[2]

[1] Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431
Universidad Complutense de Madrid (UCM)
C/ Profesor José García Santesmases s/n
Ciudad Universitaria
28040 Madrid, Spain
{robson,javiergv}@fdi.ucm.es
[2] Departamento de Engenharia Elétrica
Faculdade de Tecnologia
Universidade de Brasília
Campus Universitário Darcy Ribeiro
Asa Norte – Brasília – DF
CEP 70.910-900, Brazil
desousa@unb.br

Abstract. In a decentralized network, such as a peer-to-peer or a spontaneous network, a significant trust factor for a peer is to gain a sufficient level of certainty on the other peers' real identity. In this paper we evaluate a proposed peer identification protocol that was designed for such environments and operates based on iterated challenge-response exchanges among peers. For this purpose, we introduce a new attack against this protocol and use the birthday paradox to model the number of operations until the proposed attack is successful. The modeling process, which results in the estimation of the upper bound effort for this successful attack, gives way to the definition of enhancements for the identification protocol. As a result, we define a new identification protocol based on multiple integer challenge-responses that, though not being a cryptographic protocol, represents for an attacker a challenge harder than breaking a symmetric cryptographic key by brute force attack. Our proposed attack shows how to break the GCP protocol without any previous knowledge on target secret information.

Keywords: Information Security, Spontaneous Self-Organized Networks, Peer Identification, Identification Protocol, Trust.

1 Introduction

The trust management in spontaneous self-organized networks is a mechanism used to decrease the complexity of the access control and authorization of operations, allowing an approach to cope with the decentralized structure of these networks, the dynamic of arrival and departure of users, the mobility and the possibility of a large number of participant entities.

O. Gervasi et al. (Eds.): ICCSA 2008, Part II, LNCS 5073, pp. 526–540, 2008.

In these spontaneous networks, the network control subsystem as well as the applications constitute distributed systems, which organize themselves autonomously in order to provide a communication and processing structure to the participant entities, even thought they do not have a previously determined structure neither a centralized control. These characteristics are commonly presented in mobile *ad-hoc* networks, computational grids, peer-to-peer systems, agent communities etc.

In these distributed self-organized systems the problem of identifying the entities that initially do not know or do not have the possibility to recognize each other is always present. In such a scenario, it is possible (and actually, very easy) that a remote entity presents more than one identity, which characterizes a Sybil attack [1]. Thus, in order to control the access or to authorize operations, an entity must be sure that different identities belong to different entities. Despite that, there is also the possibility of a group of remote entities to form an attack coalition. Trust [4], [8] has an important role in pointing out directions and tries to find real and effective solutions for identity and security problems.

The Green Card Protocol (GCP) [2] proposes a way to an entity, in a distributed system, for acquiring a sufficient level of trust in the real identity of another entity. GCP is based on an iterated challenge-answer process between two entities, with no restriction to the communication subsystem used, what tries to simplify the identification process.

Having the GCP way of operation in mind, the formal model of the environment presented in [1] for the multi-identity attack is considered adequate, in this article, for the analysis of an attack against the GCP. This formal model establishes the basic characteristics of a distributed self-organized system, which includes these reasonable assumptions:

- A group of entities, including a subgroup of well behaving entities and another sub group of misbehaving entities (possibly working in coalition).
- A broadcast communication medium that can be used by any entity, meaning that every entity may send a message that will be broadcasted to all other entities and every message can be observed by all entities.
- The messages are received by all other entities within a finite time interval. The delivery of the messages is guaranteed, but there is no guarantee that the entities will receive the messages in the same order.
- There is a minimum restriction of the computational resources available for each entity; however, this restriction allows the entities to establish point-to-point private and authenticated communication paths using public key cryptography.

Considering the context presented above, this article proposes a new attack plan against the GCP. This proposed plan is based on the mathematics behind the Birthday Paradox [3], a formulation known as the base of the birthday cryptographic attack. Then, the Birthday Paradox solution is used to estimate the upper bound effort for a successful attack against the GCP, using the attack plan presented, and this effort is shown to be much smaller than that it takes to accomplish a brute force attack completely. Moreover, the GCP protocol was implemented to allow a verification of the characteristics of the attack plan, through a distributed simulation.

As a result of this analysis, given the required enhancements for the GCP protocol, we define a new identification protocol based on multiple integer challenge-responses that, though not being a cryptographic protocol, represents for an attacker a challenge harder than breaking a symmetric cryptographic key by brute force attack.

The article is organized as follows. Section 2 describes the GCP and section 3, the Birthday Paradox. Section 4, 5 and 6 presents the attack plan against the GCP and the estimative of the upper bound effort for a successful attack, with some implementation data. Section 7 presents further discussion that is important about enhancing security in identification and gives the specification of a new approach for a challenge-based identification protocol. The conclusions, in section 8, address other possible security issues and discuss some necessities for strengthening this protocol. Also we point directions for further work with the integration of the notion of trust into an identification protocol.

2 Characteristics of the GCP Protocol

The GCP proposes a way to dynamically manage the identification of various entities (nodes) in completely decentralized environments, characterized by the absence of an infrastructure responsible for providing at all times the authentication of the nodes (in centralized or hybrid environments this authentication is usually provided by a central data base server, that verifies the logins and passwords or codes generated by cryptography functions).

Facing the absence of a centralized authentication entity, the responsibility of identifying other entities is given to the entity itself in an individual process. And that requires an exchange of information among the participants to provide some guarantee on the authenticity of the identity presented by each one. An entity A may verify the authenticity of the identity of B, by comparing the data provided by B with the data stored about B in a local database in A. To do so, A will send challenges to B, who must give the answers expected by A for each challenge.

Each entity is responsible for generating independently its identification data, which is its innate knowledge. For an entity A the innate knowledge K={<*q*,*r*>}, is given by a set of tuples *q* e *r*, with the following characteristics:

1. The elements *q* and *r* are non negative integers. The field *q* represents the challenge and *r* the answer.
2. The possible values of q are limited to q < k, k being a constant value known by all entities. The tuples that form K can be represented in memory by tables with size k.
3. The value r is randomly generated. Although it is not specified in the protocol, it is assumed the r is within the interval [0,d}, with d=2^n, where n is the number of bits used to represent r. There is no loss of generality with this assumption, since n may be arbitrarily large and, can express in a simple way the characteristics of the protocol, like the memory usage by K (if the protocol is used by E entities, the memory usage in bits for each entity is E*k*n. For example, if there are 500 entities, using 256 tuples with 16 bits (2 bytes) for the values of r, it is necessary a database of 256 Kbytes for each entity).

The protocol uses this knowledge to exchange the challenges and answers in an iterated way between two entities. The content of the exchanged messages are the integer numbers q and r, which are the challenges and answers. For each challenge q sent by an entity A, an entity B must send back a correspondent answer r, as expected by A. Each correct answer provided by B, results in an increase of the trust A has in the authenticity of B's identity. The protocol is symmetric and has a piggybacking functionality, through which entity B, while responding a challenge from A, also sends a challenge to be answered by A. In each iteration, each entity chooses the challenge it will send randomly, that is, the challenge from A to B consists in a random choice from a position in the table of tuples which B must have previously acquired from A.

If an entity B is already identified by an entity A, the GCP also suggests a way for entity B to introduce a third entity C to A, thus creating a sort of presenting mechanism. This process brings up the issue of transitivity impossibility [4] in the trust relation from C to B to A, assuming that if A trusts in B, B trusts in C, it does not mean that A trusts in C. This is an important trust issue that is not explicit in the GCP specification.

It is also important to note that the GCP is not a cryptographic protocol, that is, there is no assumptions that the creation and communication of the tuples $<q,r>$ demand cryptographic functions or operations. The GCP specification [2] presents the following arguments in favor of the protocol, especially concerning the robustness against impostors and the computational cost:

- Since r is a randomly generated number, it is possible, from a statistical point of view, to keep extremely low the probability of an imposter guessing the value of r correctly to a given challenge q, without knowing the tuple $<q,r>$.
- In an iteration of challenge-answer, a single incorrect reply results in the immediate rejection of the identity. But a correct reply does not mean the validation of the identity, but it only increases the probability of the identity being associated with the correct entity. Thus, it is possible to choose a number of iterations that provides an acceptable certainty of the identity.
- Considering that two entities A and B have already mutually validated their identities, the use of the protocol allows A and B to verify jointly if an entity C is an impostor. In case there are inconsistency between the data kept by A and B about C, then C may not be trustful.
- The protocol requires, in computational terms, processing power to compare integer numbers and storage capacity for databases containing the tuples of integers.

Considering all the aspects mentioned here, the only chance for an impostor not being discovered would be to answer correctly a sequence of challenges. This breach is exploited in the attack plan proposed in the following section.

3 Attack Plan Based on the Birthday Paradox

The attack plan is based on the use of an impostor with multiples identities, i.e. a Sybil class impostor SCI (or, equivalently, a coalition of imposters, since GCP provides the

means for coalitions, due to the possibility of a third entity to be introduced between two already mutually trusted entities). In this manner, the imposter SCI can start an arbitrary number of iterative sections, presenting itself as B, trying to identify itself to an entity A.

The statistic analysis of the possibilities of SCI guessing correct answers indicates a possible attack against the GCP based on the probabilistic phenomenon called Birthday Paradox 3. It is interesting to note that such denomination refers to the fact that, to find a group of people in which (with 50% of possibilities) at least two people have the same date of birth, it is necessary to randomly gather a minimum of 23 people, which is a much lower value than 365 (the possible dates of the year), fact that goes against the intuition of the majority of the people. The underlying mathematics to this phenomenon is the base to the notorious Birthday Attack against cryptographic hash functions, given that the number of N-bits hash values that must be generated before getting a collision is not 2^N, but only $2^{N/2}$ [5].

In the attack plain against the GCP, the central question is to determine the probability of the impostor SCI generating random answers and these answers coincide with values of r in tuples $<q, r>$ relative to entity B that are in the challenge-answers table of the entity A, the attack's target. The knowledge that entity A has about B is stored in A, in the form of a table of k tuples:

$$K_B = \{<q_0, r_0>, <q_1, r_1> \ldots <q_i, r_i> \ldots <q_{k-1}, r_{k-1}>\} \quad (1)$$

In case the impostor SCI has the possibility to repeatedly present answers to one same challenge q, then SCI will have repeated chances to try to guess the corresponding value r, with the correctness possibilities increasing in each attempt, given that to each new attempt SCI decreases the space of values of attempt t, discarding the values already used in previous attempts. Thus, the probability to find a value t equals to r (collision) is defined by the solution of the birthday paradox.

Remembering that the values of the trials t are random integer numbers picked from an uniform discreet distribution among d possible values in the interval [0, d), it is simple to calculate first the probability pdif(t), of all the values of t are different from r. If $t >$ d, this probability is 1 (*pigeonhole principle*). But, for $t \leq$ d, this probability will be:

$$\text{pdif (t)} = 1 \text{ x } (1–1/d) \text{ x } (1–2/d) \ldots \text{ x } (1 – (t – 1) / d) \quad (2)$$

This expression comes from that fact that the second trial cannot have the same value of the first one; the third cannot have the same value of the other two first ones, etc. An equivalent expression for this probability is:

$$\text{pdif (t)} = \text{product (i = 1, i = } t\text{-1, 1 – i / d)} \quad (3)$$

The event of at least two of the t trials result on same value (collision) is the complement to the event of all the values being different in the t trials. Thus, the collision probability is:

$$\text{pcol } (t) = 1 – \text{pdif } (t) \quad (4)$$

Fig. 1 shows, for an interval of 256 possible values, the evolution of this probability as a function of the number of trials. It is worth to notice that in this example in the twentieth trial the collision probability is greater than 50%.

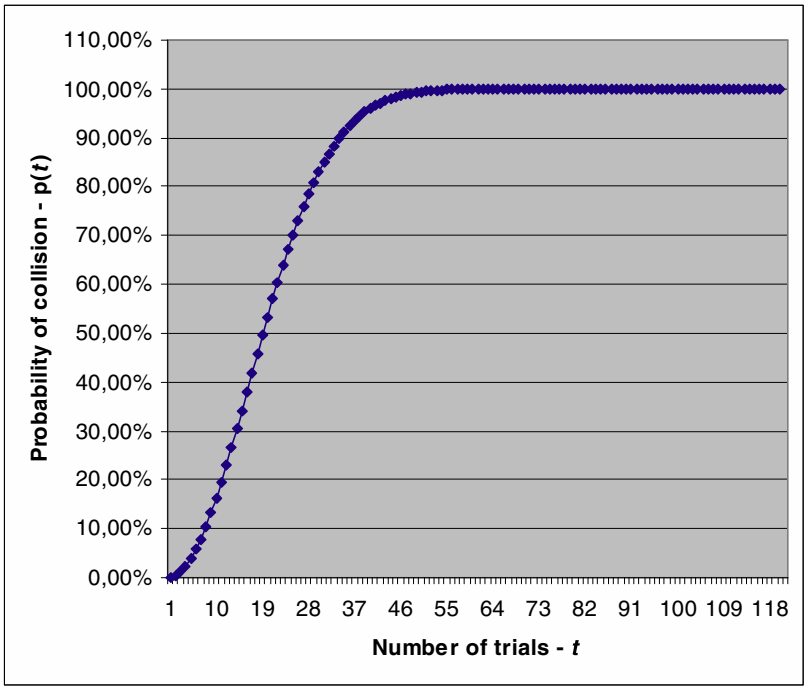


Fig. 1. Probability of collision of 2 values on t trials (d = 256)

Upon these perspectives, the plan of attack against the GCP can be formulated in the following terms:

- The impostor SCI intends to identify itself as entity B to entity A. Without having (yet) the knowledge on the identification of B in A, the impostor SCI keeps in memory a table of attempts TB = {position 0 – value [0, …, n],…, position k-1 – value [0, …, n] }, with k positions and k being a parameter of GCP known by all the entities and representing the amount of challenges that allows an establishment of trust in an identity.
- The impostor SCI introduces itself to A, as B.
- The impostor receives challenge *i* from A and, without having any information on what B would answer, SCI chooses a random reply and sends to A.
- If SCI guesses a correct answer (what may happen, since it is a random phenomenon), SCI writes down the reply in position *i* of table TB and passes to the next iteration, waiting the next challenge from A.
- If SCI guesses wrong, then its (fake) identity will immediately be refused by A, but SCI finds out that the value in position *i* of TB cannot be the attempted reply, what is a valuable and important information of the attacker's point of view.

- The impostor SCI will have then to incarnate another of its multiple personalities (as previewed in sybil attack [1]) and reintroduce itself to A, as B, one more time.
- As A is not interested in repeating the previous challenge, it chooses another random challenge and the process is repeated.
- In each iteration, the impostor SCI either answers correctly and continues or answers wrong and learns a little more, in a way that the table TB evolves with a negative knowledge on the identity of B, TB = {position 0 - (!P_1, !P_2 !P_3)},{…}, {position $_{k-1}$ - (!Q_1, !...,!Q_N). Then, for each position of table TB, the probability of SCI to guess the value that identifies B increases exactly like in the birthday paradox.

Hence an estimation of the effort of the imposter can be calculated, which is the object of the next section.

4 Estimation of the Maximum Effort of the Attack

Another approach to the issue of the proposed attack consists in considering that the impostor has all the possible tables TB. A table TB has k answers, and each answer can have d possible values.

The total number of possible tables is d^k. The impostor SCI could choose one of these tables iteratively and try the validation of identity with entity A, as B. In terms of the effort to succeed, the imposter would be facing a problem of the birthday paradox, in a space of possible d^k birthdays. However, given the attack plan above, a shortcut exists in which the impostor SCI has the possibility to attack each element of table TB separately, thus it will now face k problems of the birthday paradox, each one of these problems in a space of d possible birthdays.

The necessary effort in each case can be estimated by calculating the amount of attempts *t* to get, with more than 50% of possibilities, a randomly chosen integer number (of d possible values) that repeats the first choice (that is, one of the parts of the identification of B that is already in the knowledge base of A).

To calculate the amount of attempts, let's take the probability pesp(t, d) of in t attempts we get a value equal to a particularly determined value already chosen (equivalent to the probability of somebody in a group of t people to have the same anniversary that one determined person):

$$\text{pesp}(t, d) = 1 - ((d - 1) / d)^{t} \tag{5}$$

Inverting this expression, we get:

$$t = \log_{(d-1)/d} (1 - \text{pesp}) \tag{6}$$

To have a probability of collision greater than 50% (pesp > 1/2), the amount of attempts must be:

$$t_{1/2} \geq \log_{(d-1)/d} 1/2 \tag{7}$$

Fig.2 shows, for an interval of d = 256 possible values, the evolution of this probability in function of the number of attempts. Such result applied to the case of the plan of attack against the GCP indicates that "only" 177 attempts would be enough to the impostor generate a valid reply to one determined challenge of the attacked entity.

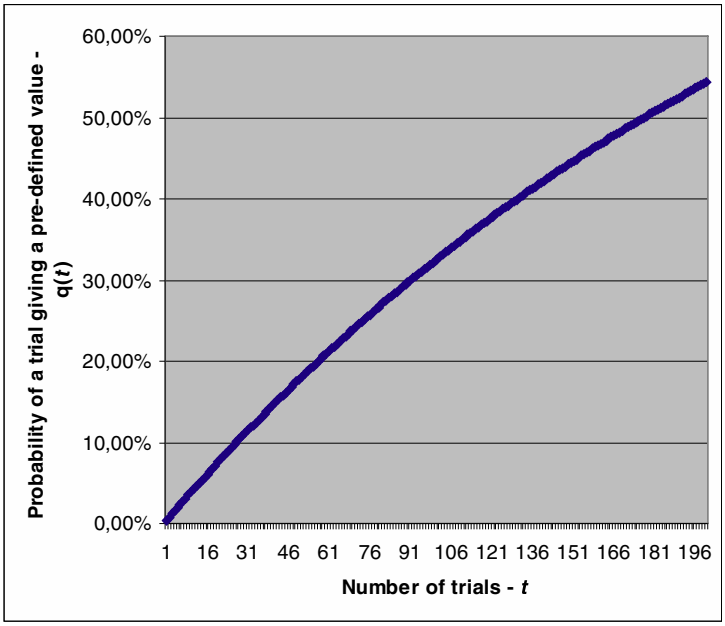


Fig. 2. Probability of a trial giving a pre-defined value (d = 256)

In consequence, considering that there are k answers to be guessed, the total effort of the attack is estimated as:

$$K \bullet t_{1/2} = k \bullet \log_{(d-1)/d} 1/2 \quad (8)$$

Finally, we reiterate that such effort can be divided among multiple personalities of the imposter, what indicates this is the upper bound for a brute force attack of the Sybil class.

5 Numerical Analysis and Simulation

The analysis of the GCP illustrates the problem of identification of entities in spontaneous self-organized distributed environments. The existence of a shared secret between pairs of entities allows the mutual identification of the entities by the exchange of information concerning this secret.

The GCP proposes that such secret, or knowledge, concerning the identity of a peer entity, should be structured in the form of a table of integer numbers, without using cryptographic resources in the protection of these numbers. From an attacker point of view this is a puzzle of integers to be guessed.

Another way to see this knowledge K={<*q*, *r*>} of an identity is to consider that it is a non-cryptographic key with size k*n, formed by k parts with size n. This allows a comparison between the effort estimation for the considered attack against the GCP and the effort for a brute force attack against a cryptographic key of the same size. It is worth to remember that, using N bits for the cryptographic key, there are 2^N possible keys and, assuming that a brute force attack has to seek an average of half of this space of keys, $2^{N/2}$ attempts are necessary to get a successful attack [6].

Using the formula obtained in section 4, Table 1 presents the effort of the attack against the GCP, for answers *r* with size *n*, for chosen values of *n*. It is also shown the amount of attempts to break a knowledge K, and the occupation of memory to store knowledge K, in a system with 500 participant entities, using 256 challenges. At last, for comparison, it is shown the necessary effort for a successful brute force attack against a cryptographic key of size 256**n*.

Table 1. Estimated maximum effort for the birthday attack against GCP

Size (bits) of challenge answers (n)	Number of possible challenge answers (d)	Number of trials for a successful attack against 1 challenge (t)	Number of trials for a successful attack against 256 challenges (256 * t)	Memory occupation (bytes) in each entity (500 entities, 256 challenges)	Effort for breaking a 256*n bits cryptographic key
8	256	177	45.337	128.000	2^1024
16	65.536	45.426	11.628.991	256.000	2^2048
32	4.294.967.296	2.977.044.471	762.123.384.697	512.000	2^4096

Table 2 compares the effort to break knowledge K of the GCP with the effort to break keys of the DES cipher standard [6]. To get a comparison, in the same conditions of memory occupation, the size of 64 bits of DES key (56 bits of the key + 8 bits of parity) is used as limit of the size of knowledge K. Thus, in knowledge K, the answers with greater size imply in the use of a lesser amount of challenges. Being considered with either 4 16-bit or 2 32-bit answers, GCP compares favorably, representing a challenge (integer puzzle to be guessed) for an attacker as hard as, even much more hard than, breaking a DES key.

Table 2. Effort comparison between the birthday attack against GCP the brute force attack against DES keys

Size (bits) of challenge answers (n)	Number of possible challenge answers (d)	Number of trials for a successful attack against 1 challenge (t)	Number of challenges (k=(56+8)/n)	Quantidade de tentativas para sucesso do ataque (k * t)	Memory occupation (bytes) in each entity (500 entities)	Effort for breaking a 56 bits DES key (2^28)
8	256	177	8	1.417	4.000	268.435.456
16	65.536	45.426	4	181.703	4.000	268.435.456
32	4.294.967.296	2.977.044.471	2	5.954.088.943	4.000	268.435.456

The implementation of the protocol allowed us to obtain data from simulation of this protocol usage in a distributed environment. Table 3 shows results from an experiment with distributed software agents using GCP, with an impostor agent executing our attack

Table 3. Data from the simulation of the proposed attack against GCP

Size (bits) of challenge answers	Number of challenges	Estimated number of trials for a successful attack	Number of trials for a successful attack in protocol simulation
8	8	1.417	1.166
8	256	45.337	32.753
16	4	181.703	128.758
16	256	11.628.991	8.199.951

plan against a target agent. Shown results come from a large number of repetitions of the attack and contain medium values from these repetitions. As expected, the impostor effort is in every case under the estimated maximum effort obtained in section 4.

The estimative of the maximum effort of a successful attack against the protocol of identification GCP is not, however, sufficient for a characterization of the protocol, in particular with respect to other possible vulnerabilities and to the robustness of the protocol to attempts of exploration of these vulnerabilities. We comment on these issues in the next section.

6 Comments on GCP Vulnerabilities

One of the points that deserve attention is the initialization of knowledge K in each entity. Besides the issues of tuple indexation and memory occupation, it is necessary to define how the initialization of tables occur between peer entities and to analyze the consequences in terms of important parameters for the application environment, such as the expense of energy in the processing and the communications, as well as the communication channel bandwidth occupation.

Also, there is the issue of parameter k that all entity must know previously to configure knowledge K. The choice of parameter k seems to be determined by the necessity to choose the lesser possible k for a determined robustness against attacks, as greater k is, more storage space is consumed and more complexity in processing the answers is added. And it may be necessary to change the value of k dynamically, what would lead to an integral reset of the system.

The problem of the choice of a global parameter k has an equivalency in the use of the threshold cryptography [7], by coincidence a cryptography system where a global key is composed of p parts (each part is delivered to a present entity in the system), k of this p parts allow an reconstruction of enough information to accomplish operations of certification and authentication in a distributed system. The problem of the adequate choice of k is still without an optimized solution that takes in account the dynamics of entrance and exit of entities in the system, and consequently the dynamic change of the number of parts p.

The automation of the initialization process requires some cryptographic system among entities of the GCP. With a cryptographic system, the use of the GCP is questionable, since symmetrical or asymmetrical cryptography system also would allow, with equivalent effort and robustness, as indicated in Tables 1 and 2, the identification

of peer entities, and other inexistent functionalities in the GCP, such as the confidentiality and the authentication of the messages. But, it is worth to comment that the usage of key exchange protocols and one time session keys, associated to an identification integer puzzle, are promising solutions for strengthening this initialization process.

Concerning the confidentiality necessity, for hypothesis, the GCP must operate in an environment susceptible to attacks, that is, in the presence of impostors. Then, for all practical effect the protocol must be ready to operate in a broadcast medium in the context of the distributed auto-organized system. Thus, the challenge-reply messages are spread out in the system and can easily be captured. Not being ciphered, such messages allow the impostors to acquire knowledge K, without performing active attacks. To avoid this possibility, it is rational to think of the hypothesis of a communication subsystem protected by cryptography, what leads us back to the question of the necessity of the GCP, in case there is already a cryptography system in action.

Therefore, some more study remains to be done regarding the configuration of the protocol and comparing it to cryptographic identification systems, thou it seems that non-cryptographic integer puzzles can be interesting for peer identification in decentralized networks.

7 Enhancing the Protocol

As result of the protocol analysis it is important to say that the estimation effort for a peer is very low in the number of attempts. This assumption takes in consideration the size that a tuple $<q,r>$ may have in bytes.

Assuming the kind of information that can be exchanged among peers in MANET, P2P or GRID systems, and the quantity of messages that goes through these kind of systems, the incorrect size of a challenge-response function and how it is formulated generates security flaws.

Having both assumptions in mind, the following paragraphs show how to enhance the strength of an identification protocol, like GCP.

The first enhancement concerns the initialization of the identification table, which must be built with k values. Each k corresponds to a tuple $t_i <q_i, r_i>$. So if k is of size 256, and r is an integer of size s in bytes, the table t could be built as:

$$t = \{ t_1 = <1, r_1>, \ldots, t_n = <n, r_n>, \ldots, t_{256} = <256, r_{256}> \} \quad (9)$$

Each element of this table is computed and exchanged among peers in the network using a Difie-Helman [9] approach. This will guarantee the confidentiality and the integrity of the identity table.

Another enhancement is defined for the exchange of challenge and responses during identification of peers. Instead of sending just one challenge at a time, one peer must send at least 3 challenges in only one message. The message M is a combination of three different challenges C. With this assumption the message can be expressed as:

$$M = \{(C_1, C_2, C_3) + T_{DH}\} \quad (10)$$

Where each C is a tuple $< q , r >$, q is the position in the table, r is the value of the position and T_{DH} is the timestamp information exchanged with Diffie-Hellman. Extending the discussion, if the challenge-response mechanism is modified in order to send three or more challenges per message, the attacker now has to estimate not only one answer at a time, but the full message. This significantly increases the difficulty of breaking the challenge, because the probability now is much different of $2^{N/2}$.

Following this approach, the entity that sends the challenge may not confirm just one challenge C_1, C_2 or C_3, giving the opportunity to the attacker to built the challenge table by guessing. The sender must confirm the full challenge (not individual C_1, C_2, C_3) with just an answer of OK (M is correct) or ERR (M is incorrect). Thus not informing which tuple is incorrect.

Now, the only manner to initiate the communication process is that the attacker or the other entity really knows three or more answers of the identity table at one step. And if the protocol defines that is necessary at least three steps with correct response to message M to start the communication, the attacker can not efficiently built a challenge-response table as in GCP. In this case the attacker is forced to know at least three different answers at a time, so he can answer efficiently only one challenge (9 positions and the correct answer in total for a full handshake with this modification – three steps).

In the point of view of a legitimate entity (the one that has the correct table) in is only necessary pick up others values in its table. But in the attacker's point of view the probability of guessing three correct challenges at a time is much more difficult. And would take much more computing resources and time to solve the puzzle, thus making the challenge-response mechanism much more difficult to break by brute force, sybil attack or birthday paradox.

The size of *k* implies directly in more calculation to the attacker because is has to compute more high values in terms of probability. The higher the *k* value is the more difficult is to guess it, because the number of values would increase exponentially. The size of *r* in bytes is also important because the higher *r* is the higher is the set that an attacker must guess.

Then the following may be considered, a) the proper size of the challenge implies in more difficulty to the attacker (size of *k* and size of *r*), b) the use of Diffie-Hellman can help the confidentiality and the integrity of the protocol and also SSH [100] can be used to help exchange the tables.

To increase more the strength of the protocol, the answers to the message M should be built in terms of hash functions in conjunction with timestamp information exchanged with Diffie-Hellman T_{DH}. Let the response R of message M be:

$$R_M = \text{hash}\ \{M = [(C_1, A_1), (C_2, A_2), (C_3, A_3)] + T_{DH}\} \tag{11}$$

Where A_1, A_2 and A_3 are the answers to challenges C_1, C_2 and C_3. The entity that sends the challenge knows the answers of each message M because he can also built the hash of the message M. He can also verify if the R_M corresponds to the given challenge using an internal table t_i (M, R_M, T_{DH}). This avoids interception of clear responses, because no value, but the hash is sent through the network.

Than the steps of the new protocol would be:

1. Built the table as: $\{t_1 = <1, r_1>, \ldots, t_n = <n, r_n>, \ldots, t_k = <k, r_{256}>\}$, where k is an integer of the size of the table and *r* an integer with the size of the challenge in bytes.
2. Exchange the table among peers in the network using a Difie-Helman or SSH approach.
3. Built challenges as $M = \{(C_1, C_2, C_3) + T_{DH}\}$, where each C is a tuple $<q,r>$, *q* is the position in the table, *r* is the value of the position and T_{DH} is the timestamp information exchanged with Diffie-Hellman.
4. Response of challenge M as R_M = hash $\{M=[(C_1, A_1), (C_2, A_2), (C_3, A_3)] + T_{DH}\}$.
5. Starts the communication process if in local table (M, R_M, T_{DH}) the received hash corresponds to its pair in the table position after a three round handshake.

8 Conclusion and Future Work

One of the contributions shown in this paper comes from the analysis of an identification protocol, evaluation its strength against a brute force attack and the proposal of a new schema.

To do so, we formulated a novel attack based on the sybill attack and used the birthday paradox probabilistic phenomenon to deduce the maximum effort for this attack to be successful. We then developed the numerical analysis of the attack and implemented the protocol to gather data from a distributed simulation of its usage. Our proposed attack shows how to break the GCP protocol without any previous knowledge on target secret information. We propose enhancements to the identification protocol so that it can represent a hard challenge for an attacker. Then it seems to be interesting to continue research work on the utilization of integer puzzles for peer identification in decentralized networks.

The commented vulnerabilities and the possible solutions can be evaluated in numerical analysis, so a new protocol can be developed with solutions for peer knowledge initialization, confidentiality of challenge answers and resistance to answer replay. To avoid sending information in clear text, the negotiation of the communication process would be efficiently secured due to the Diffie-Hellman and SSH protocol characteristics. This point has already been proved its efficiency in terms of confidentiality and integrity once there is many implementations of both protocols.

Also the use of hash in the answers would avoid man-in-the-middle attacks. It protects the information and does not let an attacker know the correct answer to the message M, because it seems that hash functions, timestamp information and symmetric cryptography offer promising solutions for these issues and should increase strength against Sybil attacks. Besides all modifications proposed to the identification protocol, trust approach [8], [4], [11] can be considered as an approach to future work.

The trust approach can help the identification of peers in network, so it would try to address answers to questions like - how can one decide with which entities it may cooperate to reach its own objectives? To correct answer this question further

research is needed because there is no specific formalism to model trust in groups and also represent the trust consensus.

Trust mechanisms can contribute to enhance the protocol in many different aspects because the trust vision can be observed as a modular scenario [111]. Considering this point of view, trust is built computing the combination of specific information (acquired trust, past situations, reputation consideration, and more). Also more problems may emerge in trust considerations thus making assumptions more difficult to evaluate and implement because of the intransitive characteristic of trust, that must be observed.

It is important to keep with further research because there is no assumption for a group trust model. This assumption, by its own characteristic, is not a trivial problem because an accepted formalism that represents trust consensus and group trust representation has not been achieved.

This problem represents new challenge in the trust research area because most of the known trust models represent the one to one (1:1) approach. But in distributed environments in many cases is necessary the one to many (1:N) or many to many (N:N) approach. In other words, it means that in distributed environments one may assume the communication process with the whole group (maybe without knowing all of its members).

Considering the same situation, the communication process may need the group to group view (N:N), what means that in distributed environments, the necessity to exchange or acquire information about groups may happen.

And without specific research in trust protocols, trust models, there is no short-time solution in distributed networks and identification schemas.

Acknowledgements

The authors thank MEC (Ministerio de Educación y Ciencia, Spain, under Project TEC2007-67129/TCM) and MITyC (Ministerio de Industria, Turismo y Comercio, Spain, under Project FIT 360000-2007-48) for their financial support.

References

1. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
2. Campadello, S.: The Green Card Protocol: an identification protocol for decentralized systems. In: Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 647–651. IEEE Comp. Soc, Los Alamitos (2006)
3. Bloom, D.: A birthday problem. American Mathematical Monthly 80, 1141–1142 (1973)
4. Marsh, S.P.: Formalising Trust as a Computational Concept. PhD thesis, Dept. Comp. Science and Math., University of Stirling (1994)
5. Bellare, M., Kohno, T.: Hash Function Balance and Its Impact on Birthday Attacks. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 401–418. Springer, Heidelberg (2004)
6. Stallings, W.: Cryptography and Networks Security – Principles and Practices, pp. 82–83. Pearson Prentice Hall, Upper Saddle River (2006)

7. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
8. Gambetta, D.: Can We Trust Trust? In: Gambetta, D. (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, ch.13, pp. 213–237. Department of Sociology, University of Oxford (2000)
9. RFC 2631 - Diffie-Hellman Key Agreement Method
10. RFC 4251 - SSH Protocol Architecture
11. Patel, J.: A Trust and Reputation Model for Agent-Based Virtual Organizations. Thesis of Doctor of Philosophy. Faculty of Engineering and Applied Science. School of Electronics and Computer Science. University of Southampton (January 2007)

Virtualization with Automated Services Catalog for Providing Integrated Information Technology Infrastructure

Robson de Oliveira Albuquerque[1,2], Luis Javier García Villalba[1], Osmar Ribeiro Torres[2], and Flavio Elias Gomes de Deus[2]

[1] Group of Analysis, Security and Systems (GASS)
Department of Software Engineering and Artificial Intelligence (DISIA)
School of Computer Science, Office 431
Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases s/n
Ciudad Universitaria, 28040 Madrid, Spain
{robson,javiergv}@fdi.ucm.es
[2] Department of Electrical Engineering (ENE)
Universidade de Brasília (UnB)
Asa Norte – Brasilia, D. F., Brazil
robson@redes.unb.br, osmar@oi.net.br, flavio@nmi.unb.br

Abstract. This paper proposes a service catalog service integrated with virtualized systems aiming at the possibility of raising and automating the availability in an IT (Information Technology) infrastructure. This paper demonstrates that aligning the server virtualization concepts and infrastructure management tools is possible to have gains in time and costs when compared to systems without automated service catalog. The main results presented illustrates that the use of a virtualized environment, with a standard services catalog and specific tools for infrastructure management, provides a time saving, reducing the request interval to a new server from several days to a few hours.

1 Introduction

Organizations considered leaders in its industries are no longer purely focused on costs, but they have also become companies focused in value.The present panorama forces them to aspire, at the same time, in one hand for the gain of productivity and efficiency, and on the other hand, for an increase in the area of capacity of Information Technology (IT) in meeting the new demands of business strategy [1].

The agile, reliable and precise obtaining of technological resources might meet the demands of the two challenges proposed. It is evident that the servers infrastructure (hardware, software and IT services) used need to evolve in order to sustain the technological innovations by leveraging IT resources (for IT resources, it is considered the servers infrastructure, involving hardware and software).

It is taken into account the need of integration amongst the various concepts and technologies involved to support an IT service. For this paper, the definition of "IT service" used is presented by Galup *et al* [1], where it is related to one or more IT

J.M. Alcaraz Calero et al. (Eds.): ATC 2011, LNCS 6906, pp. 75–91, 2011.

systems that enable a business process, taking into account that an IT system is a combination of hardware, software, facilities, processes and people. This article proposes the development of an automated catalog of services, integrating it with an infrastructure management tool and servers virtualization platforms.

The integration of servers virtualization tools with an infrastructure management tooland the services catalog aim at some objectives. First, standardizing the requests for new operational resources, such as servers, basic software and applications. Second, automating the availability of the requested resources as soon as they are approved. Third, reducing the time of availability of a new server, and at last, minimizing operational costs with specialized labor.

Currently, the time required to deliver an IT service depends directly on the stages of request, approval, acquisition, and installation of hardware/software exclusive to serve only one set of applications or systems.

Infrastructure management tools and an adequate control of the entire IT infrastructure can help reduce costs. Companies may losemoney without a services catalog, since its users do not know which IT services are supported by the IT department in terms of virtualization. Besides that, without a well-defined configuration management process, there is also an underutilization of IT resources.

Many companies utilize services from the IT area that cannot be interrupted. Thus, servers are required to remain connected full time and to be responsible for supporting a given service, for instance, a financial transaction server.

Therefore, in the same company there might have various computers with underutilized resources. An alternative solution to this problem is the use of virtualization [2] to group diverse services and other applications that need to be available in parallel.

With virtualization it is possible to consolidate and isolate different virtual machines, multiple operating systems (O.S.), thus uniting various logical servers on a single physical device [3], as illustrated in Figure 1.

The major contribution of this article isthe development of the System of Requests Registration of IT Infrastructure (SRRITI),which was created in order to integrate the concepts of three technologies: servers virtualization tools, infrastructure management tool and the services catalog. As an additional contribution, we can highlight the presentation of some results on simulation tests, proving the gains of this work.

In order to present the System of Requests Registration of IT Infrastructure (SRRITI), its contribution, its functionalities and concepts involved, this paper is organized as follows: in section II the concepts analyzed in the proposed work are presented. In section III the proposal of a service catalog for virtual servers is displayed and the environment is described. In section IV, the tests and results are introduced, and in section V the conclusions and future work are presented.

2 Bibliographic Review

Up to the present moment, proposals for servers virtualization [4], with an infrastructure management tool [5], and a services catalog [1], have been treated as independent concepts and do not serve the purpose of integration and cost reduction sought in this work.

According to [7] the major software is licensed for one single CPU, that is, the user has the right to use the software in one single system. When the discussion comes to large enterprise the situation is much worse because the need is to use software in more CPU at the same time but the demand for hardware varies.

When it is considered the time and cost of hardware and software, virtualization became an alternative for companies interested in providing system infrastructure without having to add more physical devices.

The System of Requests Registration of IT Infrastructure (SRRITI) proposes the integration between the previous concepts to serve its purposes.

A more detailed description of these concepts can be found in the following sections.

2.1 Virtual Machines

A virtual machine (VM) can be defined as an efficient and isolated duplicate of a real machine. In other words, it is an isolated copy of a physical system and this copy is fully protected.The term virtual machine has been described in the 1960s from one term of operating system, or a software abstraction that sees a physical system (real machine) [2].

The heart of the system, known as virtual machine monitor (VMM), runs directly on hardware. It implements the multiprogramming, thus providing not one, but multiple virtual machines to the next layer located above, as it is shown in Figure 1. Indeed, they are exact copies of the hardware [7].

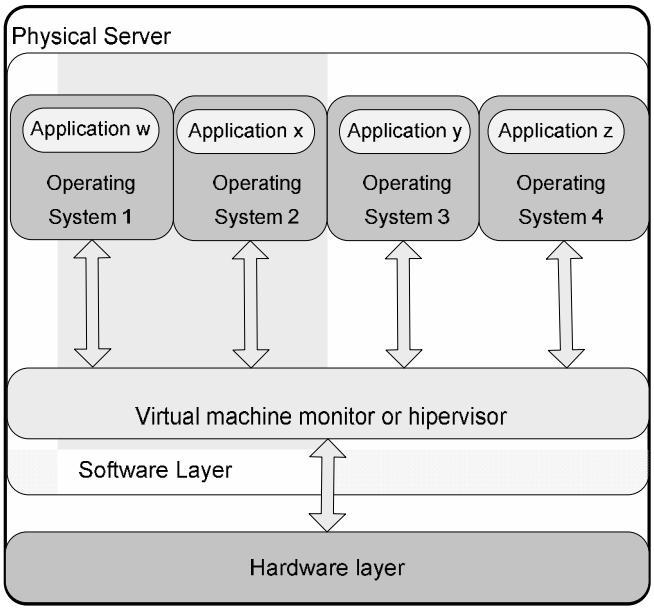


Fig. 1. A Physical Machine with 4 Virtual Machines

By running multiple instances of virtual machines on the same hardware, an efficient use of its processing power is also provided. In data centers, the reduction of physical machines means reduced costs for physical infrastructure such as space, power, cabling, cooling, support and maintenance of various systems [4].

There are four main architectures for virtualization in modern computing that allow the illusion of isolated systems: emulators, full virtualization, paravirtualization and virtualization in operating system level [8].

For this article, the full virtualization was chosen in virtue of the support for implementation given by nearly all virtualization software suppliers which were researched and cited.

2.2 Tools for Virtualization

Any person who currently uses a computer knows that there is something called operating system, which somehow controls the diverse devices composing it. The classical definition for operating system is a software layer inserted between the hardware and the applications that perform tasks for the users, and whose goal is to make the use of computers at the same time, more efficient and convenient [6].

There are commercial solutions, free software, integrated to operating systems, etc. It would be impossible and outside the scope of this article, to comment on all of them, therefore, we chose to present only the ones which are currently market leaders: VMware[9], Xen [8], QEMU[10] and Virtual Box[11]. Besides them, there is Microsoft's answer to the worldwide movement of virtualization [12].

2.3 Data Center Environment Management Tools

If a company wants to make the most of the process of computerization, organizational innovations are needed to sustain the technological innovations [13].

Tools that manage the virtualized environment (virtual machine motors), and also enable the management of the data center environment as a whole, are being developed. These machines can be either physical or virtual, amongst these initiatives, the ones researched were: Cobber[14], Puppet [15] and BladeLogic [5].

The tool BladeLogic was integrated into the service catalog developed in this work in order to enable the automation of availability tasks of a new server. This tool was chosen because of its differentiated amount of resources in relation to the other two competing tools surveyed.

2.4 Services Catalog

The IT services catalog is a menu offered by the information technology department to users of this corporation [16].

The catalog has all the services offered, software and corporate systems that can be installed and supported, avoiding users to request something that is not supported by the IT department.

With the increasing dependency of organizations in relation to Information Technology (IT), the importance of IT Service Management becomes larger every day. It is an excellent opportunity for IT to demonstrate its value and ability to leverage and bring innovation to business processes. But this is not a simple task.It demands clarity of focus and attention of the IT area [1].

2.5 Related Work

In [18] there is a discussion that agrees with the importance of a service catalog. When the infrastructure comes to a cloud environment it becomes important to formally represent knowledge in a services catalog. The focus is to enable automatic answering of user requests and sharing of building blocks across service offerings. In their work is proposed an ontology-driven methodology for formal modeling of the service offerings and associated processes.

[19] presents a model to support decision making for investments in IT services and affirms that it contributes to IT service portfolio management. Their work analyzes business impacts and investment options considering a Service Level Agreement (SLA) policy.

Discussion about the integration of multiple virtualized management tools for enterprise is discussed in [20]. It points that enterprise systems are in direction of the cloud and thus presents a strategy for accomplishing the migration process. It also considers the importance of integrated system management in user environment perspective.

When automation comes to the point of view regarding technology, [21] presents a large discussion of the subject. It presents reviews, benefits, domains and levels of application. One of the main contributions of the work presented is that automation inspires creative work and develops newer solutions. Than concludes the work with several emerging trends in collaborative control and automation, and risks to anticipate and eliminate situations where automation cannot be forget.

3 Proposal of the System of Requests Registration of IT Infrastructure

This section is divided in small parts to describe the main characteristics and functionalities of SRRITI.

3.1 System Persistent Layer

In order to collect users' information in a standardized way, the SRRITI was developed. The System provides its users with the hardware and software settings supported by the IT department. The whole system was developed using HTML and PHP pages and the support of a database as persistence layer.Table 1resumes the main tables and its characteristics.

Table 1. Resume of System Persistence Layer

NAME	DESCRIPTION
1.tb_usuarios	Storage system users' information such as profile and user identification.
2.tb_servidor_fisico	Table for the physical server pool that may support virtualized environment
3.tb_servidor_logico	Its main function is to standardize the names of the systems and main OS available for virtualization.
4.tb_requisicoes	Main system tableandstoresinformationaboutuser requests and status.
5.tb_hardware_processador	Stores information about physical infrastructure total number of available processors.
6.tb_hardware_capacidade_disco	Disk size related to physical available capacity.
7.tb_hardware_memoria	Memory size related to physical available capacity
8.tb_hardware_placa_rede	Stores the network interfaces speed to the virtualized hardware.
9.tb_software_sistema_operacional	This table stores the OS that has been previously prepared for installationusing a data center management tool.
10.tb_software_backup	Maintains the backup software that has been previously prepared to be installed for backing up a virtualized system.
11.tb_software_monitoracao	Maintains the monitoring software that has been previously prepared to be installed for the virtualized system.
12.tb_software_automacao	Stores the automation software for the selected virtualized system.
13.tb_software_servidor_http	This table stores software for HTTP servers that are available for installation in a virtualized environment.
14.tb_software_transferencia_arquivos	It maintains software for file transfer servers that are available for installation in a virtualized environment.
15.tb_software_banco_de_dados	Stores software for Database servers that are available for installation in a virtualized environment.

3.2 System Logics

The project was designed segmenting administration functions, registration of requests, and approval of registered requests. According to the user's credentials, he or she is redirected to the screen functionalities, in agreement with his/her previously registered profile.

Once the request is registered in the system, there must be an approval of the solicitation before it can be provided. The consultation to a request and its subsequent approval is accessed by users who have the approver profile or administrator.

At the moment of approval of a new server request, the files, which will interface with the virtualization tool in order to create the new server, are generated in disk. The files of interface with the infrastructure management tool are also created in order to install the operating system, and the previously registered applications. At this point it is chosen the physical machine where the virtual machine will be created. Figure 2 demonstrates the flow of a new request. Moreover, it is also possible to provision a physical machine without any virtualization feature within the developed system.

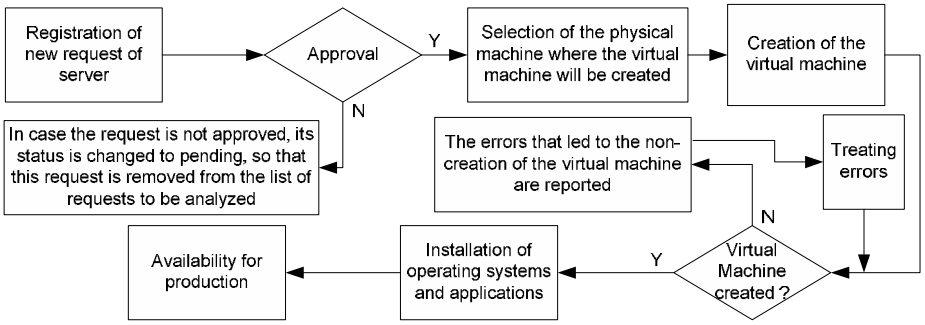


Fig. 2. Flowchart of a New Request

All the files created have the request number to which they are associated; therefore, it is possible to run more than one provisioning at the same time.

3.3 System Basic Caractheristics and Functions

SRRITI was developed using PHP with Apache HTTP Web Server and MySQL as database server. The system divides the user profile based in three main modules: 1) user requests, where users perform its system requests; 2) system approval, where system requests are approved following enterprise process policy by IT department; 3) the SRRITI administration, which is conducted by IT specialists in virtualization and system data center management.

The developed system has a lot of input screens where users can perform its actions based on its profile. As an example, Figure 3 shows one of the screens of SRRITI. There it is presented the registration of a new request from a server. On this screen, certain items are available: processor, disk capacity, memory size, speed of network card, operating system, monitoring software, backup software, automation software, software for transferring files and database software.

The main advantage of SRRITI is that in concentrates the user requests in one point of control and thus reduces de process complexity of requesting a new virtualized system.

It is important to consider that the automation of IT infrastructure is recommended for large enterprises with a heterogeneous and complex computing park. There are no well-known reports regarding the minimum number of servers or database, or even

operating systems that suggest the minimum amount of the related items cited above should be automated.

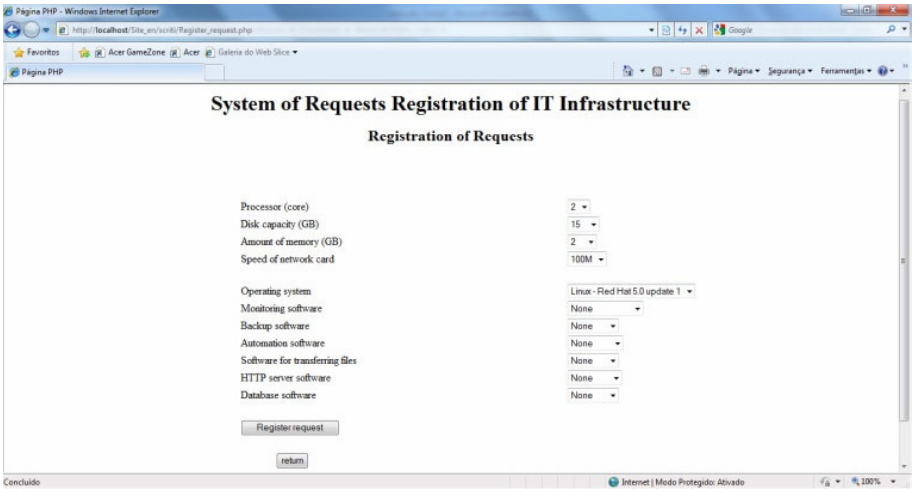


Fig. 3. Screen of Registration of Requests of the System SRRITI

3.4 System Main Outputs

Once the user completes a request for a virtualized service and has the IT department approval, SRRITIautomatically generates scripts to be directly executed in the data center management tool. SRRITIsystem files outputs are based in well-known standards as XML and BAT files, which are easily interpreted and may be imported and integrated in most systems and tools using common programing language. Figure 4 shows a XML output example.

Fig. 4. XML Output of an User System Request

Also there are other XMLs inputs to the system, for instance, system name, system mac-address, system profile, system OS, etc. Each of them depends of the user request and availability of the IT infrastructure. Once the creation of a new virtualized environment is allowed by IT personal, SRRITI reads the XMLs files as data input than connects to the data center management tool and pass the new virtualized system parameters to be created. The whole process is automated using specific commands depending on the data center management tools. Figure 5 shows one type of command that can be executed.

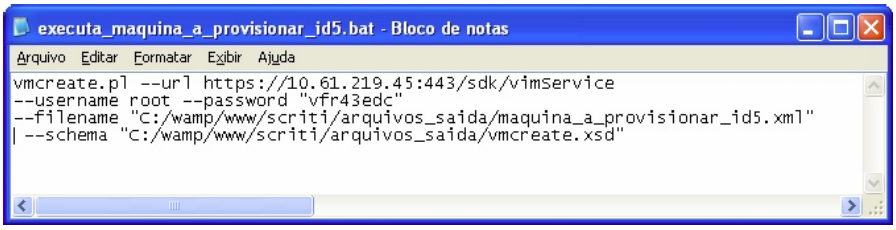


Fig. 5. System Command for Creating a Virtualized System

4 Tests and Results

The testing environment was built seeking to clarify the following questions:

a) Is it viable to automate the availability of a new server from a service request?

b) Is it possible to develop and integrate a services catalog with hypervisors and IT infrastructure management tools, enabling a reduction in the time availability of a new server and also reducing the operating system installation time?

c) How much can the operating cost be reduced by using the integration of concepts and tools presented?

SRRITI was tested as a response to the questions presented.

4.1 Testing Environment

For this article the VMware ESX Server 4, which is the base software for creating virtual data center, was used. The ESX server is a virtual machine monitor that virtualizes hardware resources like processor, memory, storage and networking. Thus, the ESX Server allows a physical server to be partitioned into several isolated and secure virtual machines and, each one is seen as a physical machine in a conventional network infrastructure.

Tests for the creation of virtual machines were performed. Following this, the installation of the operating systems RedHat Enterprise Linux 5.0 update 1 and Windows Server 2003, with various configurations of central processing unit (CPU) and variation of Random Access Memory (RAM), took place.

All the tests were performed by booting only a VM at a time in order to avoid any kind of interference in the tests due to the amount of memory allocated in some other virtual machine.

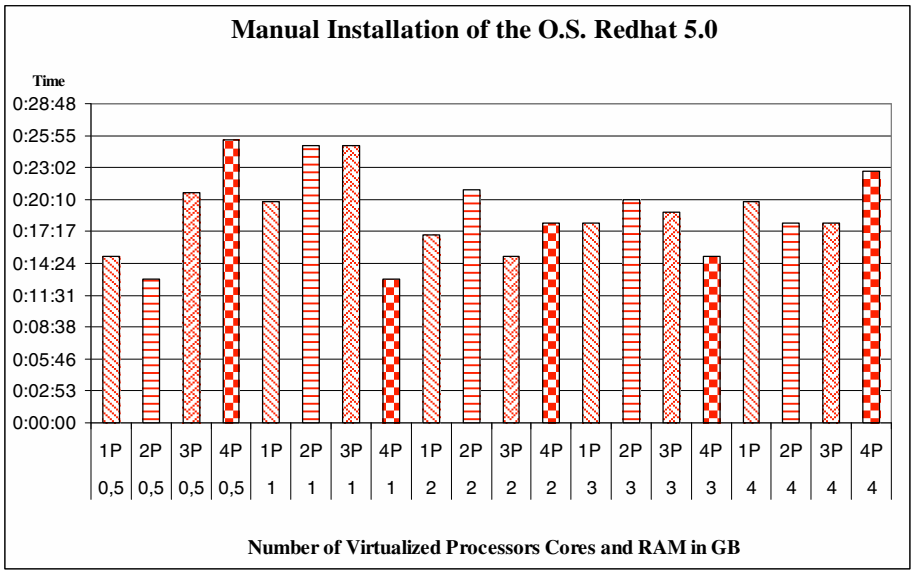


Fig. 6. Manual Installation of RedHat Enterprise Linux 5.0 update 1

The graphs of this paper are organized as follows: on the left side, it is shown the time spent for installation in the format hours: minutes: seconds (h:m:s). At the bottom of each graph, the number of processors (cores: from 1P to 4P) of the virtual machine created, and the amount of RAM in GB (from 0,5 to 4) allocated to each machine, respectively, are presented.

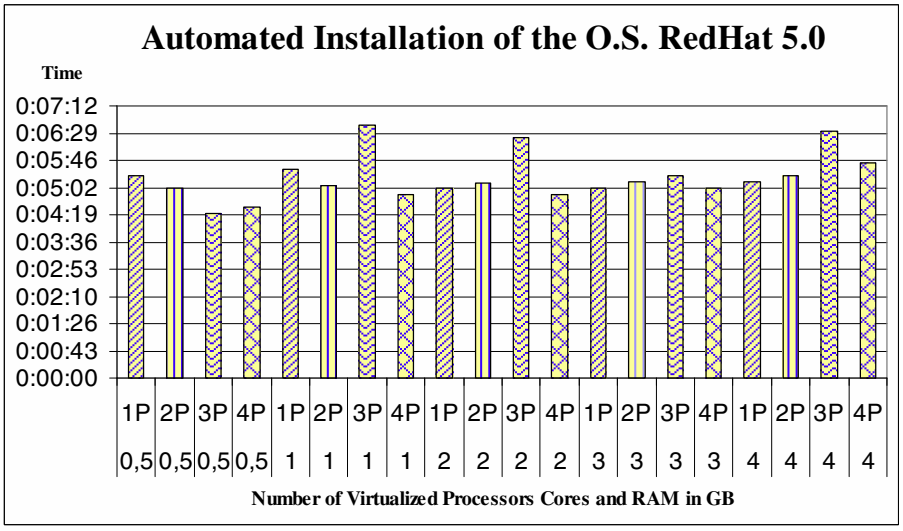


Fig. 7. Automated Installation of RedHat Enterprise Linux 5.0 update 1

Figure 6 illustrates the results obtained in the tests of the manual installation of the operating system, RedHat Enterprise Linux 5.0 update 1. Figure 7 presents the results obtained in the automated installation tests of the operating system RedHat Enterprise Linux 5.0 update 1 utilizing SRRITI.

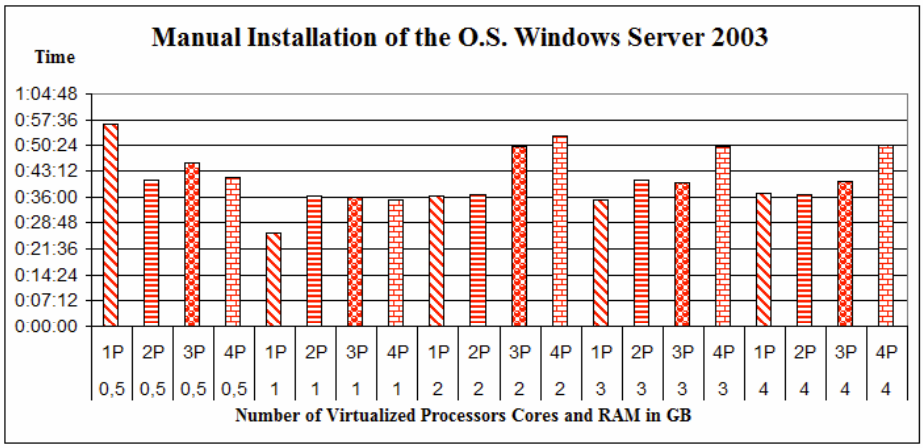


Fig. 8. Manual Installation of the O.S. Windows Server 2003

Figure 8 shows the results obtained in the manual testing of installation of Windows Server 2003 operating system. Figure 9 illustrates the results obtained in the automated test of installation of Windows Server 2003 operating system.

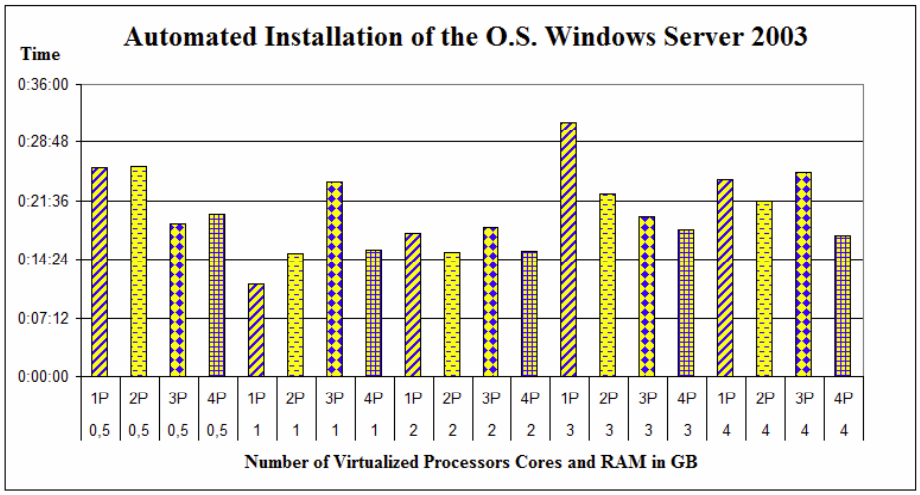


Fig. 9. Automated Installation of the O.S. Windows Server 2003

As shown in Figures 6 through 9 the numbers of processors are not changed because the objective of these tests was to compare the time that the same machine with

the same characteristics would take to perform the process of installation of the virtualized environment in a manual fashion compared to an automated fashion provided by SRRITI.

4.2 Comparison of Results

Technology can be used for automating operations. The objective is to replace the effort and provide the human qualification via technologies that allow the same processes to be executed at a lower cost, under control and continuity.

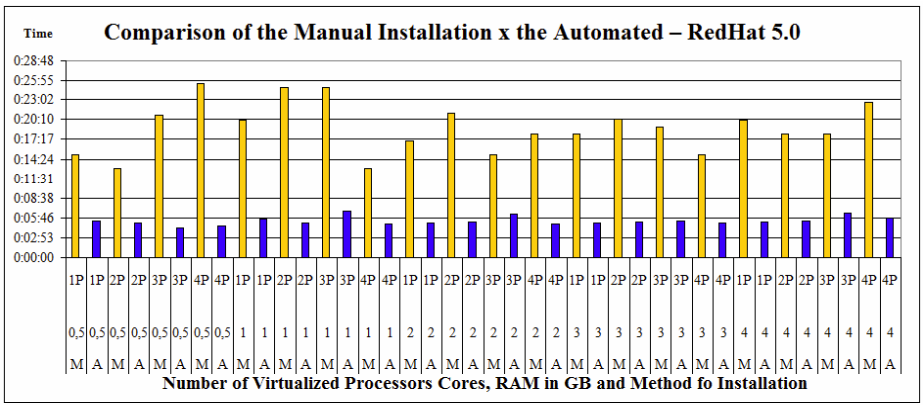


Fig. 10. Comparison of the Manual Installation x the Automated - O.S. Redhat Enterprise Linux 5.0 update 1

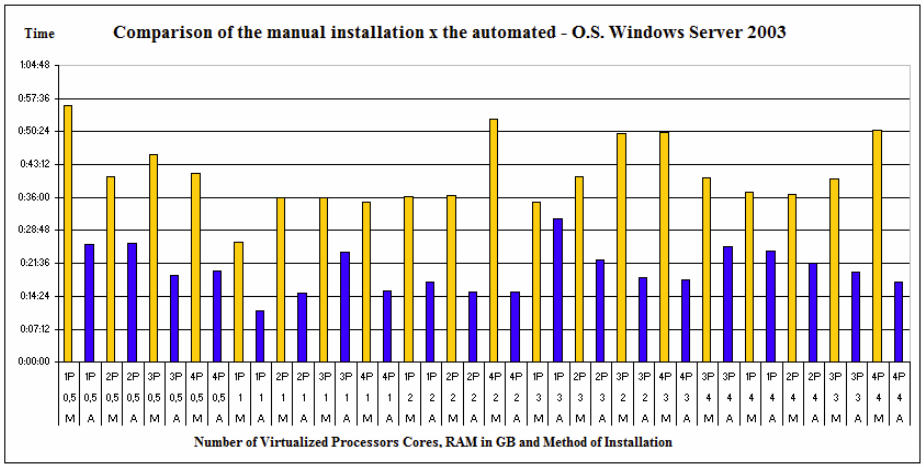


Fig. 11. Comparison of the Manual Installation x the Automated - O.S. Windows Server 2003

Automating a process, which is executed manually, initially, aims at reducing the possibility of human error and, in the background, increasing the productivity by

making available a good or service. This can be verified comparing graphs of results in Figures 10 and 11. They show that an automated process has better time results that a manual installation process.

In addition, information concerning time spent with installation; number of processors (cores) of the virtual machine created; amount of RAM in GB; and the method used for installation manual (M) versus automated (A) are also described.

The system SRRITI proposed in this paper, confirms these concepts.The productivity gain ranged from 57.56% for the scenario of two processors and 2 GB of RAM in the worst case, to 82.30% for the installation of the operating system RedHat Enterprise Linux, including three processors and 0.5 GB RAM, as it is presented in Figure 10.

In relation to the operating system Windows Server 2003, the productivity gain ranged from 10.38% for the scenario of one processor and 3 GB RAM, representing the worst case, to 70.91% in a scenario of four processors and 2 GB RAM, which was the best case as illustrated in Figure 11.

4.3 Comparative Analysis of Costs Estimate

In the case of a large telecommunications enterprise studied in this paper, after analyzing the whole process that is used to make an virtualized system available (Figure 12), it was observed that the process takes 46 days from the user request up to the deploy of the user request. Considering an 8-hour workday, it is possible to conclude that from the request to its proper availability, 368 working hours are required in order to provide a new server. These are reference values, which were obtained through the analysis of the studied process which are also listed in Table 2.

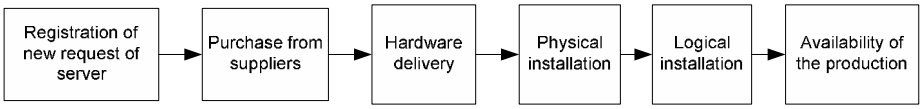


Fig. 12. The Process of a Large Telecommunications Company in Order to Make a Server Available

Based on a salary survey [17], in Table 2, the average salary of a support analyst is presented (expressed in Brazilian currency, reals). The final amount was divided by the total working days within a month, and then this value was divided by the total of working hours. These calculations are needed to reach the analyst's value of one hour of work.

Table 2. Amount of Time Demanded for the Availability of a New Server without Virtualization

Reference Value	Description
46	Workdays in order to make a new server available
8	Daily work hours
368	Total of hours needed for creating a new server

In Table 3 the value of an analyst's hour of work is presented. This value is multiplied by the total of hours needed for creating a new server. The final amount represents how much it costs for a company to create a new server using only specialized labor.

Table 3. Cost of one Hour of Work of a Support Analyst

Reference Value	Description
R$4616,48 22	Average salary of a senior Linux support analyst. Total of workdays in a commercial month.
R$209,84	Valueof a workday.
R$ 26,23	Value of one hour of work of a Linux support analyst.

Table 4 presents a financial analysis utilizing non-automated virtualization infrastructure.

Table 4. Cost for a Company with Specialized Labor to Create a New Server without Virtualization

Reference Value	Description
368	Total of hours needed to make a new server available
R$26,23	Value of a support analyst's hour of work
RS 9.652,64	Total cost of an analyst working for 46 days

In Table 5 a financial analysis using virtualization and the automated process integrated with the services catalog are shown, also the amount of time needed in order to make a new server available through the system SRRITI, which was developed and presented in this paper.

Table 5. Financial Analysis of the Cost utilizing Virtualization

Reference Value	Description
48	Total of hours needed in order to make a new server available with virtualization
RS 26,23	Value of a support analyst's hour of work
RS 1259,04	Total cost in order to make a new server available

The approval of request may be responsible for the time increase in making a new server available, and for this reason, it was considered the worst case: two workdays or 16 hours were necessary for the new server to become available. This analysis is resumed in Table 6.

In order to solve this matter, a set of physical servers with available resources to enable the creation of virtual machines is demanded.

Table 6. Financial Analysis of the Cost of the Automated Process Integrated with the Services Catalog

Reference value	Description
16	Total of hours needed to make a new server available
RS 26,23	Value of a support analyst's hour of work
RS 419,68	Total cost in order to make a new server available

5 Conclusions

Automating a process for a few executions can be more financially costly and onerous in amount of time than executing the proceeding manually. However, after a defined number of repetitions in each specific environment, the time invested, and, consequently, the capital allocated, are to compensate the time spent on automating the task.

The automated process reduced the amount of time demanded for the availability of a new server in virtue of the standardization of requests. With the standardization of the requests, it was possible to automate the delivery of required resources as soon as they were approved. This fact reduced the costs with specialized labor.

Table 7 presents data on the reduction of time and operating costs achieved through the utilization of the System of Requests Registration of IT Infrastructure (SRRITI). Comparing the time to deliver a new server with virtualization to SRRITI, there was a time reduction of 66.67% with the use of the system SRRITI.

Table 7. Comparative analysis of time and cost

Scenario	Time (in work hours)	Cost (in Reais R$)	Percentagegain (*)
WithoutVirtualization	368	9,652.64	--
WithVirtualization	48	1,259.04	66.67
Via the system SRRITI	16	419.68	95.65

(*) The percentage gain considers the time column in relation to the system SRRITI.

With the integration of concepts presented in this paper (virtualization, infrastructure management and services catalog), and through SRRITI it was possible to observe the gains offered and their true contribution to the standardization and automation of IT services. The reduction of time and costs also adds essential value to the System of Requests Registration of IT Infrastructure.

5.1 Future Work

As future work, new studies regarding trust, virtualization, cloud and service catalog may be necessary in order to provide more availability do users in mixed environments.

Trust and security [22], [23] [24] have become crucial to guarantee the healthy development of cloud platforms. Most studies tries to provide solutions for concerns

such as the lack of privacy and protection. These characteristics are important to guarantee security and author rights.

When trust comes to discussion, it is also important to consider that there is no common trust and reputation consensus in distributed environment, for example, to guarantee that a pool of servers are trustworthy in the same service catalog. That makes trust and reputation analysis fully dependent of specific variables and the definitions of the environment that it is attached to.

In the cloud the situation gets more complicated because it is necessary to employ trusts model in cloud environments to guarantee users security and privacy.

Acknowledgements. This work was supported by the Ministerio de Ciencia e Innovación (MICINN, Spain) through Project TEC2010-18894/TCM and the Ministerio de Industria, Turismo y Comercio (MITyC, Spain) through Project AVANZA COMPETITIVIDAD I+D+I TSI-020100-2010-482.

References

1. Galup, S.D., Dattero, R., Quan, J.J., Conger, S.: An overview of IT service management. Communications of the ACM - Security in the Browser CACM Homepage table of contents archive 52(5), 124–127 (2009); ISSN: 0001-0782 EISSN: 1557-7317
2. Poniatowski, M.: Foundations of Green IT: Consolidation, Virtualization, Efficiency, and ROI in the Data Center. Prentice Hall, Englewood Cliffs (2009); ISBN: 0-13-704375-9
3. Benevenuto, F., Fernandes, C., Santos, M., Almeida, V.A.F., Almeida, J.M., Janakiraman, G.J., Santos, J.R.: Performance Models for Virtualized Applications. In: ISPA Workshops Conference Proceedings, pp. 427–439 (2006)
4. Carlsson, N., Arlitt, M.: Towards more effective utilization of computer systems. In: Proceeding of the Second Joint WOSP/SIPEW International Conference on Performance Engineering. ACM, New York (2011); ISBN: 978-1-4503-0519-8
5. BMC. BMC Service Automation The next step in the evolution of Business Service Management. BMC Software (2009), `http://documents.bmc.com/products/documents/10/45/91045/91045.pdf` (accessed on January 05, 2010)
6. Stallings, W.: Operating Systems: Internals and Design Principles, 5th edn. Prentice Hall, Englewood Cliffs (2005); ISBN-10: 0131479547
7. Tanenbaum, A.S.: Modern Operating Systems, 3/E. Prentice Hall, Englewood Cliffs (2008); ISBN-10: 0136006639. ISBN-13: 9780136006633
8. Govindan, S., Choi, J., Nath, A.R., Das, A., Urgaonkar, B., Sivasubramaniam, A.: Xen and Co.: Communication-Aware CPU Management in Consolidated Xen-Based Hosting Platforms. IEEE Transactions on Computers, 1111–1125 (August 2009)
9. VMWARE. VMware ESX e VMware ESXi. VmWare (2010), `http://www.vmware.com/files/br/pdf/products/VMW_09Q1_BRO_ESX_ESXi_BR_A4_P6_R2.pdf` (access January 20, 2010)
10. Becker, M.: Qemu/systemc cosimulation at differet abstraction levels.University of Paderborn/C-LAB. Fuerstenallee 11, 33102 Paderborn, `http://adt.cs.upb.de/quf/quf2011_proceedings.pdf#page=13` (access January 25, 2010)

11. Virtualbox. VirtualBox. VirtualBox (2010), http://www.virtualbox.org/wiki/VirtualBox_architecture (access January 22, 2010)
12. Microsoft. Microsoft Virtual Server. Microsoft (2010), http://www.microsoft.com/windowsserversystem/virtualserver/ (access January 25, 2010)
13. Lundvall, B.-Å.: National Systems of Innovation: Toward a Theory of Innovation and Interactive Learning. The Anthem Other Canon Series. Paperback (January 1, 2010)
14. COBBLER. Cobbler. cobbler (2010), https://fedorahosted.org/cobbler/ (access March 05, 2010)
15. PUPPETLABS. Introducing Puppet. Puppetlabs (2009), http://www.puppetlabs.com/puppet/introduction/ (access September 01, 2010)
16. Curtis, D., Brittain, K:. Document the IT Service Portfolio Before Creating the IT Service Catalog. Gartner Research. ID Number: G00163200 (January 2009), http://confluence.arizona.edu/confluence/download/attachments/2459667/document_the_it_service_port_163200+%282%29.pdf
17. Info Magazine. Brazilian Salary Resarch. RH Info Human Resource Consulting (2010), http://www.rhinfo.com.br/sal-ti.htm (access November 05, 2010)
18. Deng, Y., Head, M., Kochut, A., Munson, J., Sailer, A., Shaikh, H.: An ontology based approach for cloud services catalog management. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 680–681. Springer, Heidelberg (2010)
19. Queiroz, M., Moura, A., Sauvé, J., Bartolini, C., Hickey, M.: A model for decision support in business-driven IT service portfolio management using SLA-dependent criteria and under uncertainty. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems. ACM, New York (2009)
20. Ezaki, Y., Hitoshi, M.: Integrated Management of Virtualized Infrastructure That Supports Cloud Computing: ServerView Resource Orchestrator. Fujitsu Science Technology Journal (2011), http://www.fujitsu.com/downloads/MAG/vol47-2/paper18.pdf
21. Nof, S.Y.: Automation: What It Means to Us Around the World. Springer Handbook of Automation, pp. 13–52. Springer, Heidelberg (2009)
22. Wang, H.-z., Huang, L.-s.: An improved trusted cloud computing platform model based on DAA and Privacy CA scheme. In: IEEE International Conference on Computer Application and System Modeling, ICCASM 2010 (2010); ISBN: 978-1-4244-7235-2
23. Shen, Z., Li, L., Yan, F., Wu, X.: Cloud Computing System Based on Trusted Computing Platform. In: IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA), China, vol. 1, pp. 942–945 (2010)
24. Li, X.-Y., Zhou, L.-T., Shi, Y., Guo, Y.: A Trusted Computing Environment Model in Cloud Architecture. In: Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, China, pp. 11–14 (July 2010); ISBN: 978-1-4244-6526-2

Proceedings of the 2012 International Conference on High Performance Computing & Simulation (HPCS 2012)

HPCS 2012

July 2 - July 6, 2012 • Madrid, Spain

Editor:
Waleed W. Smari

Assistant Editor:
Vesna Zeljkovic

The Proceedings of the 2012 International Conference on High Performance Computing & Simulat ion (HPCS) was produced for IEEE HPCS by The Printing House, Inc.

If you have questions regarding the installation, please contact:

The Printing House

The Printing House, Inc.
Phone: +1-608-873-4500 Fax: +1-608-873-4558
Hours: Monday through Friday, 8 am - 5 pm CST
E-mail: graphics@printinghouseinc.com

Main Menu

Credit: Joan J. Martinez
en.wikipedia.org

Group Trust Model

Robson de Oliveira Albuquerque, Luis Javier García Villalba
Group of Analysis, Security and Systems (GASS)
Department of Software Engineering and Artificial Intelligence (DISIA)
School of Computer Science, Office 431
Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases s/n
Ciudad Universitaria, 28040 Madrid, Spain
E-mail: {robson, javiergv}@fdi.ucm.es

Abstract— **This paper proposes a model for group trust that can be used in distributed systems. Considering that trust has become a research area in distributed environments, this piece of work presents a new approach in trust computing considering groups or subgroups so it can evaluate how trustworthy a group is. In our approach, group is defined as a collection of entities with particular affinities and capabilities. All entities may have a trust and a reputation value of each other in the system. In many cases it may be necessary to trust the whole system instead of one particular entity. In such cases group trust may represent the trust of their particular members. As a result we implemented the proposed model using two groups with a hundred nodes each in different scenarios. We could achieve a trust value for the group.**

Keywords- Group trust; trust; reputation; distributed system

I. INTRODUCTION

Trust is an actual subject in many research fields. In computation, trust represents a manner of creating a more reliable environment, thus allowing users and systems to achieve new scenarios where it was not possible before. In this paper an entity is considered any node that is part of a distributed system. Different authors have studied and defined trust [1] [2] [3] [4] [5] [6] [7]. In their work, trust is considered to be applied to specific systems and situations. This paper also defines trust and extends the discussion into group scenario.

Nowadays trust is used in many different computational contexts. As examples of its application as distributed systems there are P2P networks, Computational Grids [8], MANETs [4] and software agents [5]. Recently trust became a research area in Cloud computing [9] [10] [11].

Most solutions deal with trust depending on a model and its application inside a specific environment. Considering this, it is possible to say that one specific trust model cannot be adopted as the most complete and the most efficient for all scenarios.

In general trust is used in a one-to-one relationship or situation. That means trust may be calculated if an entity, let's say A, establishes a communication process with other entity B inside a specific context, a can calculate the trust it has in B for that particular communication. Extending this view, entity A can create another communication process with as many entities as it may need to do its work. For that, A may use network resources, services identification and any available resource that it has in its connected environment.

Now let's say that A needs to create a trust based communication process with a group of systems in a distributed environment (grid, cloud, cluster, etc.) but it needs trust guarantee that the group itself is trustworthy. In most trust situations this is not easily achieved. That is because for A there is no common group trust representation. If A wants to form an opinion by itself about the whole group it may be necessary to establish communication with every group member. This makes entity A start an exhaustive process of discovering every member of the group. In the most practical situations this is not a good approach because it will make A try to find what could be thousands of nodes in a network.

To address this problem, this piece of work proposes a group trust calculation model and presents a calculation process that permits an entity to find a trust value for the group, thus avoiding discovering the whole group members at once.

This paper is organized as follows. Section II reviews some aspects and the concept of trust and reputation. Section III presents the proposed group model. Section IV shows the implementation results and section V concludes this paper.

II. TRUST AND REPUTATION

A. Trust definition

Marsh [3] is one of the first to study trust in computer science. He provided a clarification of trust concepts applied to computational systems, presented an implementable formalism, and developed a trust model in order to enable agents to make trust-based decisions. He argues that trust involves probability and this permits a representation of trust in values between zero and one.

Patel [1] believes that the use of trust in computational relations includes different considerations. Among them there is the optimized selection of a communication partner, function delegation to individuals and the possibility of establishing agreements between two or more members of the network before the communication starts.

Gambetta [2] defines trust as a particular level of subjective probability. In his approach an agent can evaluate if another agent (or a group of agents) will carry out a particular action, before he can monitor such action and in a context that affects its own action.

In general, if trust exists in a particular environment, there is a high probability of cooperation because a particular entity can execute a useful action. On the other hand, if there is no trust that means the trust level may be too low so cooperation in the environment is discarded or avoided until there is enough trust, or completely abandoned.

Anyway, this approach creates a concept that varies trust in the sense of (0) – which means complete distrust – to (1) – or blind trust and may suffer interference of nodes, betrayal, etc. Figure 1 illustrates this consideration.

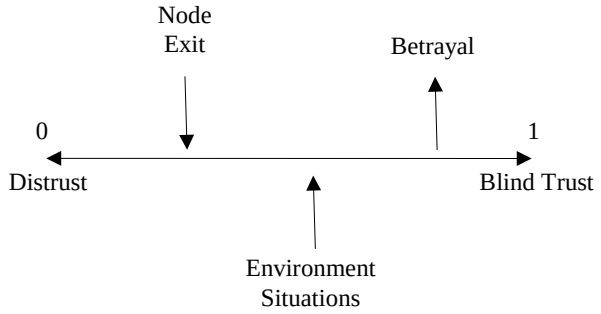


Figure 1. Trust or distrust and influences

In most scenarios blind trust is not desirable because it eliminates any possibilities of the entity inferring to suspicion of any particular situation by its own mechanisms.

Trust is recognized as an important aspect for decision-making in distributed systems [2], [3], [12], but there is no general consensus in the literature about the definition of trust and what trust management involves.

According to Jφsang *et al.* [13], positive and negative feedback about members in a distributed environment is accumulated to help trust calculation. Their model is based on the Bayesian network and uses beta probability density function to calculate members expected future behavior.

According to Aberer and Despotovic [14] entities are able to establish mutual trust. Their work employs the concept of reputation into trust management because whenever an entity requests the reputation of another, this behavior represents a derivative measure of the direct and indirect trust acquired in past interactions.

REGRET [15] proposed by Sabater and Sierra is another distributed trust model. This model uses reputation and social information about agents in trust calculation, besides considering personal experiences. This way the system can operate even if there is a new agent in the system with lack of information.

Trust reviews show that it is a subjective evaluation. To create a complete formalization, plenty of different concepts are needed. Particular contexts are involved and the application depends on the system itself and the model used. Even though there are common arguments that trust must consider. They are resumed in Table 1.

TABLE I. SUMMARY OF TRUST CHARACTERISTICS

Characteristic	Example
Trust is context or situation dependable.	One entity may trust other to download files but do not trust it to perform routing.
Trust can be measured.	A trust more in B than it trusts in C.
Trust changes with time.	The trust value that one entity has in other may increase or decrease as interaction happens.
Trust may suffer influence of a recommendation.	A can trust C that was presented by B that A already knows something about.
Trust is directional.	A may trust B, but B may not trust A.

B. Reputation

Reputation can be defined as an extension of trust. Many authors [1] [5] [15] [16] have discussed it and what the implications of its use may be.

It is important to observe that reputation is an opinion. And as an opinion it has many subjective evaluations. It depends on the situation observation by an individual, its own trust inference and may have the information received by others in a social context.

In distributed environment, reputation suffers influence of time and depends on the behavior of the entity. One very important aspect of reputation is that it should be public and is desirable that it should be available to any entity in a distributed system.

To Dagsputa [16], trust is based on the reputation which is time dependable and based on behavior. He discuss that trust is defined in means of correct expectations of an entity that influences its choices before it can observe other entities.

The definition of reputation depends on the evaluation of the situation by a particular entity [17], besides its own trust value on the information received by others entities in distributed systems, and on its own trust value on the entity that has sent the information.

In general, reputation may represent indirect trust. It involves asking for the opinion of other parties whom the entity have previously interacted with in the past about a third entity. Reputation can also be defined as the common opinion of others regarding an entity [1], which may be used in the absence of trust formed from personal opinions.

Reputation is rarely extreme (all or nothing). It takes time to be acquired, but it can easily be lost in social aspects. Calculation of reputation values may be done using past information that has been obtained and is based in the information that was received from trusted parties. These variables enable an entity to form an idea about an unknown entity. In a larger view, reputation is a social evaluation of an individual or group of individuals in direction of one entity or another group that have impact in trust formation.

III. GROUP TRUST MODEL

A group can be defined as a collection of entities connected together with common goals. These entities are able to perform specific works in a common context like service search or

service offering. They are also able to perform trust and reputation calculation of each other in the system that considers any interaction.

Entities may use any trust and reputation model that attends the system needs. This work considers that there is a trust and reputation algorithm that can perform trust and reputation calculations.

A. Trust consensus

There aren't many works in literature that develop trust consensus in distributed system. Most of them are related to trust itself in a computational manner.

To perform a group trust calculation there should be a leader in the formation of the group. It is not easy to define a leader in a trust manner. That means if we just consider leadership consensus, entities in a group are able to agree to a minimum level of trust (trust threshold) in order to make a common analysis and commonly choose a leader based on trust. The problem is that every entity has not the same trust value about any other entity. This is because trust is calculated individually making use of its own inferences. Even though entities may agree on an ordinary value of trust and also agree that this value is enough to assume that one specific entity can represent the group. This assumption would transform the chosen entity in the leader of the group.

B. Trust leadership

Leadership in trust is dependent on consensus in many aspects. If we consider that it is necessary for a leader in a group to be chosen using trust, a process of checking trust characteristic of every member of the group should be considered. If a leadership election process based on trust is needed, then it is desirable to have a trust consensus process because entities may express their opinions and it is not restricted to a one vote process. If trust leadership is achieved by trust consensus, then trust leadership should consider options, historical contexts and aspects that lead to a better decision making process regarding trust. In distributed systems a leader may be a router, a middleware broker, a dispatcher or any entity that messages can go through.

C. Group trust formalism

In our approach we consider that a leader already exists in the group and entities in the group agreed that it is the representation of the group for new members and for the outside world as well. Trust reviews indicate that any entity may have lots of relationships in many contexts. As trust is calculated considering each interaction, the final trust of one entity over another inside a context in a particular period of time can be calculate as the average result of all interactions already done for that context. It is important that the leader of the group knows every context. In order to calculate the group trust we recommend the use of reputation values that an entity has in another. This is because the leader may not be able to use just his own trust values, but he needs the opinion of every entity in the group about each other. This way the group trust will be the representation of all the opinions of every entity in the group. It means that every entity individually evaluates each other according to its own knowledge.

1) Reputation context representation

In a simple way, reputation that entity A in has in B in a particular context C may be represented as in (1.

$$\delta_{a,b}^{c} = V_{c}^{b} \quad \textbf{(1)}$$

where

$\delta_{a,b}^{c}$ represents the reputation of A in B in context C.

V is the reputation value given by A to B in context C using any available reputation model.

V_{c}^{b} is one record that represents the expectation that B will accomplish what A requests in context C.

In normal situations, trust and reputation values may be stored as many individual records of every interaction with the entity in evidence had. Thus an entity may have a collection of different reputation values about other entities in the system.

This way an entity may calculate the final reputation value of a particular context is represented as in (2).

$$\overline{\delta}_{a,b}^{c} = \frac{\sum_{i=1}^{j} V_{c_i}^{b}}{j}, \text{ with } j > 0. \quad \textbf{(2)}$$

where

$\overline{\delta}_{a,b}^{c}$ is the final reputation that A has in B in context C;

j represents the amount of reputation interactions that A had with B in context C.

As observed in trust and reputation models, one entity may have as many contexts that it is programmed to. Then the final reputation regarding all contexts of one entity about other is given by the expression in (3).

$$\overline{\kappa}_{a,b} = \frac{\sum_{i=1}^{x} \overline{\delta}_{a,b}^{c_i}}{x} \text{ with } x > 0 \quad \textbf{(3)}$$

where

$\overline{\kappa}_{a,b}$ is the final reputation value that A has in B for all contexts;

x is the amount of all contexts that A knows about B.

This way A is able to store all the reputation information that he has in B in a particular period of time. This value then will be used to perform group trust calculation.

2) Group trust representation

It is important to observe that group trust cannot be only calculated using just trust values because it is an individual opinion and do not necessarily represent the entire opinion of all entities about each other. This way we consider what best represents group trust is the image that every entity has about

all entities inside a group. Reputation is what gives this representation.

This means the group trust is based on the behavior of every entity as seen by all members of the group. Once the group leader is decided, it must collect all the reputation values. Or if group members knowing who the leader is, they may also send their reputation information instead of waiting for the leader to directly ask for it. In this case a protocol for trust exchange information may be used.

Once the leader has all information it needs, then it computes the final reputation as the average reputation value of every entity in the system using (4).

$$\overline{\omega}_g^n = \frac{\sum_{i=1}^{j} \overline{\kappa}_{a,b}^{\,i}}{j}, \text{ with } j > 0 \quad \textbf{(4)}$$

where

$\overline{\omega}_g^n$ is the average reputation of entity **n** as seen by entities of group **g** in a particular period of time;

j represents the quantity of members in group **g**;

$\overline{\kappa}_{a,b}$ is the reputation value received by the leader during each entity reputation calculation process.

After performing the final average reputation of every entity in the group, the leader can generate the final group trust value using all reputation values computed before. This computation process is represented in (5.

$$\overline{\lambda}_g = \frac{\sum_{i=1}^{x} \overline{\omega} i_g^n}{x}, \text{ with } x > 0 \quad \textbf{(5)}$$

where

$\overline{\lambda}_g$ represents the final trust of group **g** in a particular period of time

x represents the quantity of members in the group.

3) Group trust calculation algorithm

In order to create a common process to perform group calculation the algorithm represented in Figure 2 can be used.

The idea is that a leader receives or asks for the reputation values of the entities in the group. Once it has the information of the reputation information of the group members the leaders performs group trust calculation in every round that it is necessary.

It than stores the group trust value so it can be delivered to one particular entity in a many-to-one approach or to another group which represents a many-to-many-approach.

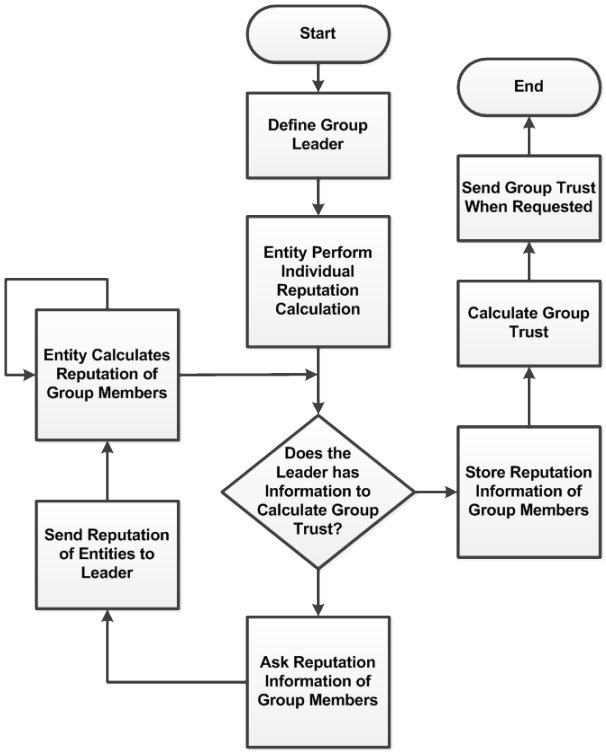


Figure 2. Algorithm for group trust calculation

IV. RESULTS AND ANALYSIS

In this section we present the implementation of group trust model proposed in this work.

A. Main Considerations

The main considerations settings applied during the tests regard the group formation and group amount of nodes. We simulated the creation and maintenance of two groups. Each group has a hundred nodes as members and they all know each other inside the group. Nodes do not communicate with nodes outside its own group. Nodes use one particular context in our simulation.

Nodes are able to perform trust and reputation calculation. We used the model proposed by Patel [1] for individual trust and reputation calculation. In every simulation we performed 20 rounds of nodes interaction, so nodes could create a reputation image of each other. We considered that a threshold ≥ 0.7, than the group is trustworthy.

Each group has a leader that is trustworthy. This means that the leader does not behave unexpectedly, it is honest and it is able to perform individual trust calculation as well.

Information regarding the environment itself, such as network, bandwidth, capacity, delay, etc. is not important for our simulation environment. We simulated three different scenarios. The results and analysis for each scenario is presented in the following items.

B. Scenario 1

In this simulation 100% of the entities behave nicely. They are honest and perform their trust and reputation calculation normally. Sends and replies information when requested. Figure 3 illustrates the behavior of group 1 and group 2 for scenario 1.

As seen in Figure 3, after round 9, group 1 and group 2 are considered trustworthy. Values of group trust are similar because they use beta distribution for generating individual trust value. As the nodes behavior do not change, group trust shows a tendency to increase until it reaches values $0.8 \geq T \geq 0.9$, where T represents the group trust value.

C. Scenario 2

In this simulation 80% of the entities in group 1 behave without change. 20% of the nodes makes a coalition and behave badly which means they do not collaborate in the group, send fakes information, do not perform their normal operation in the group. This simulates bad nodes in distributed environment. Figure 4 illustrates the behavior of group 1 and group 2 for scenario 2.

After round 4 the coalition starts. The nodes realize that some nodes change their normal behavior. This modifies trusts values, thus the group value decrease. But because the threshold ≥ 0.7, group 1 still can be considered trustworthy.

D. Scenario 3

In this scenario 60% of the entities in group 1 behave without change and 40% of the nodes makes a coalition and behave unexpectedly. Figure 5 illustrates the behavior of group 1 and group 2 for scenario 3.

After round 4 the coalition starts and nodes realize that some nodes changed behavior. This modifies trusts values, thus the group trust value decrease. In this case, because the threshold ≤ 07, group 1 is to be considered not trustworthy.

E. Main analysis

In the simulations, it was possible to realize that the individual behavior of each member in the group influences the trust value of the group as a whole. This result was satisfactory because all entities initiate in the network in the same moment and interact with each other the same number of times. As seen in scenario 2 and 3, the behavior change influences in the group trust values.

If a group leader realizes that their group trust value is below the expectation, it may decide to isolate malicious peers in the network because it can identify them through consulting other group members and verifying its trust and reputation values. After that the group trust value may became above expectation, thus getting more interactions. In case interactions represent funds, groups tend to have good behavior most of the time and isolate malicious entities because they represent a threat to the group as a whole.

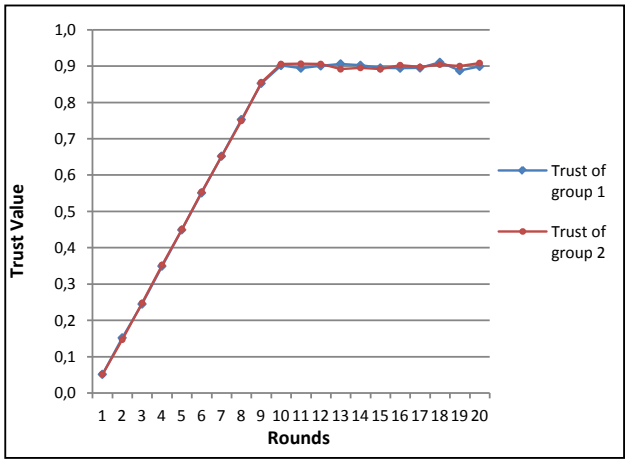


Figure 3. Trust or Distrust and influences

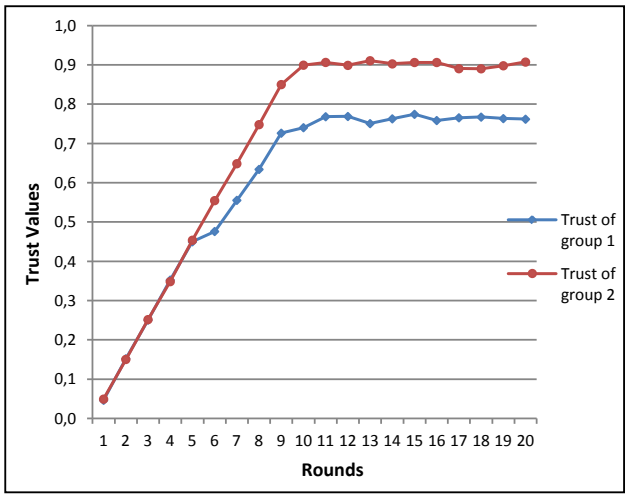


Figure 4. Group Trust with 20% of coaliton with bad behavior

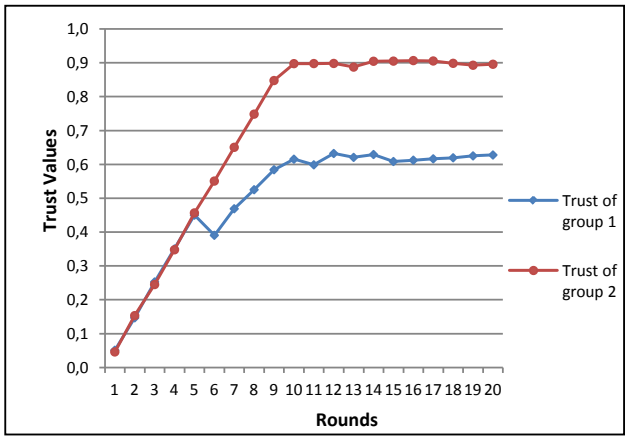


Figure 5. Group Trust with 40% of coaliton with bad behavior

V. CONCLUSIONS

This work reviewed trust and reputation and presented a model for group trust calculation. The proposed model simulated two groups with 100 nodes in each group.

To generate initial trust and reputation value we used TRAVOS [1]. There was no particular reason for using the TRAVOS model. We just used a model that was able to generate and compute individual trust and reputation values. In our implementation TRAVOS could be considered a good choice because it gave entities the ability to learn and perform trust and reputation calculation. With a small number of interactions (about 5) entities were able to identify changes in the environments. Our results show that it is possible to generate and to calculate group trust in distributed system. It is important to remember that trust leadership or trust consensuses were not used in this work. We consider this as an open point regarding trust in distributed systems.

The main objectives of this paper were fulfilled, including generating initial trust values, performing group trust calculations and depending on the behavior of individual entities in the group, we were able to identify the malicious ones in the system.

It was important to keep group trust stability and it was observed that, after some interactions, entities with good behavior established a general agreement about the identification of the malicious entities in the system.

A. Future Work

Using the concept of group trust, we expect to simulate news situations, for example, a group that behaves randomly as a whole. Thus it is important to consider more groups with more nodes. As future work we intend to implement our group trust model in other distributed system such as cloud systems, grids, p2p networks. A trust protocol for trust information exchange is also desirable. Trust consensus and trust leadership is to be considered in next researches as well.

ACKNOWLEDGMENT

This work was supported by the Ministerio de Industria, Turismo y Comercio (MITyC, Spain) through the Project Avanza Competitividad I+D+I TSI-020100-2011-165 and the Agencia Española de Cooperación Internacional para el Desarrollo (AECID, Spain) through Acción Integrada MAEC/AECID MEDITERRÁNEO A1/037528/11.

REFERENCES

[1] J. Patel, “A Trust and Reputation Model for Agent-Based Virtual Organizations,” Thesis of Doctor of Philosophy. Faculty of Engineering and Applied Science. School of Electronics and Computer Science. University of Southampton. January, 2007.

[2] D. Gambetta, “Can We Trust Trust?,” in Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford, chapter 13, pp. 213-237, 2000.

[3] S. P. Marsh, “Formalizing Trust as a Computational Concept,” Department of Computing Science and Mathematics, University of Stirling. Doctorate Thesis, April 1994.

[4] A. Adnane, , R. T. de Sousa Jr.; C. Bidan, L. Mé, “Autonomic trust reasoning enables misbehavior detection in OLSR,” in Proceedings of the 2008 ACM symposium on Applied computing (SAC '08). ACM, New York, NY, USA, pp. 2006-2013, 2008.

[5] T. D. Huynh, N. R. Jennings, N. R. Shadbolt, “An integrated trust and reputation model for open multi-agent systems,” in Autonomous Agents and Multi-Agent Systems, vol. 13(2), pp.119–154, 2006.

[6] P. Lamsal, “Understanding Trust and Security,” Department of Computer Science University of Helsiki, Finland, October 2001.

[7] C. Castelfranchi, R. Falcone, “Social Trust: A Cognitive Approach,” National Research Council - Institute of Psychology. Unit of "AI, Cognitive Modelling and Interaction", Roma, Italy.

[8] G. Fox, M. Pierce, D. Gannon, M. Thomas, “Overview of Grid Computing Environments,” Global Grid Forum, 2002.

[9] W. Han-zhang, H. Liu-sheng, “An improved trusted cloud computing platform model based on DAA and Privacy CA scheme,” IEEE International Conference on Computer Application and System Modeling (ICCASM), ISBN - 9781424472352. IEEE, 2010.

[10] Z. Shen, L. Li, F. Yan, X. Wu, “Cloud Computing System Based on Trusted Computing Platform,” IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA), vol. 1, pp. 942-945. China. 2010.

[11] X.-Y. Li, L.-T. Zhou, Y. Shi, Y. Guo, “A Trusted Computing Environment Model in Cloud Architecture,” Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, pp. 11-14. China, July 2010.

[12] T. Beth, M. Borcherding, B. Klein, “Valuation of trust in open networks,” in ESORICS 94. Brighton, UK, November 1994.

[13] Jφsang, A. Knapskog, S. J. “A metric for trusted systems,” Global IT Security, pp. 541-549, 1998.

[14] K. Aberer, Z. Despotovic, “Managing Trust in a Peer-2-Peer Information System,” In proceedings of CIKM 2001, pp. 310-317, 2001.

[15] J. Sabater, C. Sierra, “Regret: A reputation model for gregarious societies,” in Fourth Workshop on Deception Fraud and Trust in Agent Societies, pp. 61–70. 2001.

[16] P. Dasgupta, “Trust as a Commodity,” In Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford, chapter 4, pp. 49-72.

[17] J. Sabater, C. Sierra, “Review on Computational Trust and Reputation Models,” in Artificial Intelligence Review, pp. 33–60. Springer, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 872842, 9 pages
http://dx.doi.org/10.1155/2014/872842

Research Article

GTrust: Group Extension for Trust Models in Distributed Systems

Robson de Oliveira Albuquerque,[1,2] Luis Javier García Villalba,[1] and Tai-Hoon Kim[3]

[1] *Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), School of Computer Science, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases s/n, Ciudad Universitaria, 28040 Madrid, Spain*
[2] *Electrical Engineering Department, University of Brasilia, Campus Universitário Darcy Ribeiro, Asa Norte, 70910-900 Brasilia, DF, Brazil*
[3] *Department of Convergence Security, Sungshin Women's University, 249-1 Dongseon-dong 3-ga, Seoul 136-742, Republic of Korea*

Correspondence should be addressed to Luis Javier García Villalba; javiergv@fdi.ucm.es

Received 30 October 2013; Accepted 20 December 2013; Published 10 February 2014

Academic Editor: Naveen Chilamkurti

This paper proposes and describes a trust model for distributed systems based on groups of peers. A group is defined as a collection of entities with particular affinities and capabilities. All entities may have a trust and a reputation value of each other in the system. In many cases it may be necessary to trust the whole system instead of one particular entity. In such cases group trust represents the trust of their particular members. To achieve this, this paper presents a group trust calculation model. We implemented the proposed model in a P2P simulation tool and presented main results for group trust calculation.

1. Introduction

Nowadays, distributed systems have become very complex environments in that hundreds of nodes have to collaborate in order to provide large-scale services. Some examples can be P2P networks, multiagent systems, grid systems, cloud systems, and so on. In these scenarios, trust between entities becomes a crucial factor as a way of determining the reliability of the different nodes in the system and to detect and predict misbehaviors and security threats. In these environments, trust can also help in the veracity of the system.

Considering that the Internet nowadays has so many different types of things connected to it, there is no known way of dealing with the amount of information or data that is necessary to verify if systems are fully reliable. This amount of connected elements and its functions are commonly named Internet of things [1]. Once that so many possibilities of information and systems are connected and a lot of them interoperable, the amount of security threats also increases significantly because of the complexity, distribution, channels, data, and networks, which make the traditional security approaches inefficient. This also leads the technology and researches towards new manageability challenges.

It has become clear that new trust management systems are necessary in order to accomplish security related to distributed system [2] considering the amount and variety of systems connected. The management of trust relationships between different peers belonging to a distributed system can be done using different approaches. The trust relationships can be manually established by each node in the network around the rest of the nodes. This manual approach does not scale when the number of nodes becomes bigger and bigger, so other approaches based on a trust model are used to automatically calculate how much a node can trust other nodes. Usually, a given node follows a trust model to determine if a node is trustworthy or not depending on two different aspects: (i) trust values are locally calculated

and (ii) trust values are provided by the rest of the nodes in the network (reputation).

Current trust models are generally used to calculate one-to-one relationships between nodes. This means that an entity A needs to determine the trust value for all the other entities available in the distributed system, which has to communicate with A. In large-scale scenarios, this fact is an important lack of scalability in distributed systems due to the fact that A needs to calculate and maintain as many trust relations as the number of nodes necessary to accomplish A's work. Other scenarios consider several entities as a single node for trust purposes. For example, a distributed file system could be seen as a single entity even though it is composed of several subentities and a security threat in any of them may compromise the complete file system. This kind of simplifications may cause a lack of the accuracy of the current models.

When A needs to communicate with a group of nodes having the guarantee that the group itself is trustworthy, A wants to independently form an opinion about the whole group. If this is the case, it may be necessary to establish communication with every group member. This makes entity A start an exhaustive process of discovering every member of the group. In most practical situations this is not a good idea because it will make entity A try to find maybe thousands of nodes in a network. This consumes time and resources for entity A and may not be the best approach to achieving its objectives.

To address these problems, the main contribution of this paper offers a novel extension to conventional trust models that enables the support for calculating the trust values for groups of nodes. A group is defined as a collection of entities with particular affinities and capabilities. Then, the entities which need to interact with a given group are able to find a trust value for the whole group directly, thus avoiding the necessity of discovering the whole group members and providing more accurate information about the group than approaches which consider the group members as a single entity. The extension provided has been implemented and validated by means of a set of statistics.

In order to describe the proposal, this paper is organized as follows. Section 2 reviews some aspects and definitions about trust and reputation. Section 3 discusses some related work of different trust models for distributed systems. Section 4 presents the proposed group trust model. Section 5 shows implementation results and Section 6 ends this paper with the conclusions and future works.

2. Definitions

The concept of trust and its definitions has been studied by many authors and is part of many research projects. Trust is recognized as an important aspect for decision-making in distributed systems [3–6], but there is no general consensus in the literature on the definition of trust and what trust management comprehends. For example, Patel [7] considers that trust in computational relations may be focused on the optimized selection of a communication partner and on the decision of agreements between two or more members of the network. That is particularly true in distributed environments where there is no certainty about the destination and the communication process can be easily deceived. Gambetta [8] considers trust as a particular level of subjective probability and Marsh [9] says that trust involves probability and this permits a representation of trust in values between zero and one.

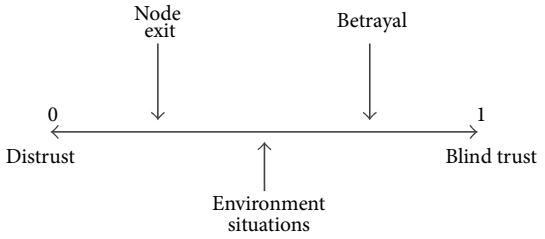


Figure 1: Trust or distrust and influences.

Table 1: Summary of common set of trust characteristics.

Characteristic	Example
Trust is context aware	Entity A may trust B to download files but does not trust B to perform routing
Trust can be measured	Entity A has more trust in entity B than A's trust in entity C
Trust changes with time	The amount A trusts B may increase or decrease as interaction happens
Trust is socially aware	Entity A may trust entity C because C was presented by entity B, and A already trusts B
Trust may be directional	Entity A may trust B, but B may not trust A

This approach creates a concept where a trust value ranges from 0, which means complete distrust, to 1 which means complete trust. This value may suffer interference of nodes, betrayal, and so forth. Moreover, complete trust is not desirable in most scenarios because it eliminates any possibilities to suspect any particular node. Figure 1 illustrates this consideration.

There are common aspects in almost all the references about trust models that make us determine a common set of features related to trust. These features can be summarized in Table 1.

As opinion, it has many subjective evaluations depending on several factors like situation observed (context), own trust value inference, information received by others in a social relationship, and so forth. The reputation value evolves with time and directly depends on the behavior of the entity being observed. Reputation may also represent indirect trust. It includes asking for the opinion of other parties with whom the entity has previously interacted in the past about a third entity. Reputation can also be defined as the common opinion of others regarding an entity [7], which may be used in the absence of trust formed from personal opinions. Reputation values take time to be acquired, but it can be easily lost

in social aspects. Calculation of reputation values is done using past information that has been obtained during time and based in the information that was received from trusted parties. These variables enable an entity to form an idea about an unknown entity. It could be considered as a social evaluation of an individual or group of individuals.

3. Related Work

There is an important number of research works providing trust models for different distributed scenarios such as grid computing, P2P, multiagents, ad hoc networks, wireless sensor networks, and cloud computing. This section provides a review of trust and reputation models in order to motivate our new trust model in context with the other works. Regarding trust models for Zhao and Dong [10] present a trust model multidomain grids scenarios and consider that trust can be used to evaluate the relationship between grid resource providers and grid consumers. Jingshan et al. [3] present a trust model for grids based on the information provided by the last service used. According to the authors, their proposed model achieved less resources occupied, increased flexibility, and prevention of threat of malicious evaluation and cooperative cheating in the grid. GridTrust [11] security framework is able to provide trust management in vertical and horizontal approach in the grid. GSF is based on layers and policies that control and monitor users activities in such a way that security and trust are guaranteed by their model.

Regarding trust models for P2P, the novelty and strong point of the model proposed in DWTrust [12] is that all the factors that have influence on the trust value for a particular node are represented as dynamic weights that adapt themselves depending on the trust policy of each node. AntRep [13] is another trust model where reputation evidence is distributed over a P2P network based on the swarm intelligence paradigm [14]. In AntRep each peer has a reputation table (RT) which is very similar to the distance-vector routing table [15] but differs in the following aspects: (i) each peer in the RT corresponds to one reputation content; (ii) the metric is the probability of choosing each neighbor as the next hop instead of the hop count to destinations. EigenTrust [16] has become one of the most cited trust models for P2P networks. It achieves a decrease in the number of downloads of inauthentic files in a P2P file-sharing network by assigning each peer a unique global trust value, based on the peer's history of uploads.

Regarding trust model for mobile ad hoc networks (MANETs), Chang et al. [17] propose a Markov chain-based trust model to determine the trust value for each one-hop neighbors in multicast MANET and the results indicate that the convergence speed is independent of the trust classes and of the initial values of the proposed model. Liu et al. [4] present a reputation model able to use subjective opinions with familiar values in order to prevent selfish behaviors in MANET. Nodes accumulate reputation information and create a familiarity value used to compute impact of reputation recommendation. RRS for P2P and MANETs [18] present an enhancement of CONFIDANT [19], which is a robust reputation system for P2P and mobile ad hoc networks where everyone maintains a reputation rating and a trust rating about everyone else they care about. PTM [20] is a decentralized trust model which expressed trust relationships with fuzzy logic. These relationships can be established as direct or indirect. In the former, A will trust B without intervention of third parties. In the latter, the indirect trust relationships are given by recommendations from TTPs. A TTP is a peer who has a trust value higher than a certain threshold. Such recommendations are distributed using a pervasive recommendation protocol (PRP) among close entities.

Regarding trust models for multiagents systems, TRAVOS [7] consists of a trust and reputation model for virtual organizations based in agents, in which trust is measured using probability. The evaluation of the amount of trust is based on past interactions and reputation obtained by other nodes. Sporas [21] is another reputation mechanism in agent systems where the reputation is computed recursively and where the more recent a rating is, the more weight it has. MTrust [22] uses a Bayesian network to calculate the trust value among entities in the network. It is focused on a mobile agent system, where the cooperative interactions among these agents and their respective visited hosts are ensured. Regret [23] (one of the most representative trust and reputation models in multiagent systems) manages reputation from three different dimensions: the individual one, given from direct interactions with the agent; the social one, from previous experiences of group members with the agent and its acquaintances; and the ontological one, given by the combination of multiple aspects in order to build a reputation about complex concepts. AFRAS [24] proposes a reputation mechanism in multiagent systems whose main characteristic is the modeling of an agent reputation and the interaction rating as fuzzy sets. In case the reader is more interested in how many of the previous models are being calculated, Gómez Mármol and Martínez Pérez [5] provide a very complete and comprehensible survey of trust models for distributed systems.

Largillier and Vassileva [25] argue that group formation is a difficult task. It has many different contexts and groups can be formed based on users criteria or using methods that matches what users desire. In most cases it does not take into account previous successful or unsuccessful collaborations to forge new ones. Considering this, their work proposes a model of collaborative trust to help select the criteria that is the best fitted group for a task. Al-Oufi et al. [26] explain that in order to protect users it is important to identify trustworthy people. Their work extends the Advogato trust metric [27] so trustworthy users can be identified. The authors [26] claim that their model has advantages over existing representative methods because it is able to discover reliable users and prevent unreliable users. Easa et al. [28] say trust is used in soft security and proposes a group-based trust method to propagate information among peers. They also consider two factors called intermediate group confidence and group confidence used between two groups.

Related to trust and the Internet of things, Saieda et al. [2] propose a new trust management system and design

a context-aware and multiservice trust management system fitting the new requirements that the authors consider important in their model. Schulz and Tjstheim [29] point out that, when there are interactions with objects and services in the Internet of things, usually the users need to trust that their data is safe and that things will fulfill their promise. Wang et al. [30] consider trust management as a way of providing a potential solution for the security issues of distributed networks and propose a new distributed trust management mechanism for the Internet of things. They propose a model using sensor, core, and application layers which provide a general framework for the study of trust management for the Internet of things.

All this preview work shows that trust is used in many different distributed technologies. Very few works consider the group aspect of distributed system as an important characteristic to help provide security and reliability in distributed systems. It is also important to remember that trust and reputation have increased significantly in recent years and nowadays they are being considered as important factors to help extend the functionalities in distributed systems.

4. Group Trust Model

This section describes the extension for supporting groups in trust models. This extension enables the definition of trust values over both groups and single entities. A group is defined as a collection of entities connected together with common goals or even common contexts. Thus the entities are able to perform specific works in a common context like service offering. Moreover, the entities are also able to perform trust and reputation calculation of other entities in the system that considers any interaction. This group extension can be deployed over a system in which entities use any trust and reputation model that attends to the system needs. This work just considers that there is a trust and a reputation algorithm that can perform trust and reputation calculations and is considered as an extension over such algorithm. The trust added value is a consequence of the individual trust values of the group members. The added value represents a point of information for external entities so they can use it to infer the whole group trust value.

To perform a trust calculation over a group, it is a requisite of our model that there should be a leader in the formation of the group. It is not easy to determine a leader for holding trust information and thus a method by consensus is adopted to determine the leadership of the group. Entities in a group may be able to agree to a minimum level of trust (trust threshold) in order to make a common analysis and commonly choose a leader based on trust. The problem is that not every entity has the same trust value about any other entity. This is because trust is calculated by every entity using its own ability and making use of its own inferences.

Entities may agree in an ordinary value of trust and also agree that this value is enough to assume that one specific entity can represent the whole group. This assumption transforms the chosen entity to the leader of the group. In real distributed system scenarios where every node can perform its own decision, such agreement is very difficult because entities must exchange trust and reputation information that should have been defined previously considering, for example, security aspects as availability and integrity. However, if it is assumed that an entity should not trust other entity if its trust value is not inside a specific range, this entity could avoid exchanging information with another in the system thus leading to a complete failure of acquiring enough trust information to create trust consensus. Besides, this should consider that some basic factors are commonly known and used by every entity in the system and that may not be true in most distributed systems. If it is considered just voting schemas for a leadership choosing process, this may not represent consensus because what an entity does is just to vote. In other words, it is to choose one among many options. Voting schemas, in general, do not consider consensus in a distributed manner, which contradicts the trust aspects. It is not the objective of this work to develop a trust consensus algorithm or a trust consensus model in order to choose a leader.

To have a consensus process enables entities to express their opinions about the leadership election process. It is important to observe that any entity in the group may be the potential leader of the group. One prerequisite is that the candidate entity has trust and reputation values expressed by the members of the group. Extending this view, any entity may announce its preferred candidate also depending on the context. That makes the leadership choosing process more complicated than just voting. For example, an entity could be responsible for taking other entities opinion and announcing in the network who has the most elevated trust value among $n + 1$ members, where n is half of the whole group. Well, in large groups that can be a problem because this process may flood the network with messages for just choosing one entity as leader. Another point of view is that entities may infer that a trust value is acceptable or not thus generating a trust threshold. If an entity does not overcome a specific trust value than another entity, it may assume by its own means that the entity in focus may not be the leader for it. However, that may not be true for another entity that believes that it has enough trust to be the leader. Both assumptions lead us to a quantitative trust measure and not a qualitative approach and in some distributed system that is not adequate. It is not the objective of this work to develop a trust leadership algorithm or a trust leadership model.

So, once a leader already exists in the groups, entities in the group have agreed that this leader is the representation of the group for new members and for the outside world. It is normal that entities may not be able to be actuated in every context available in the system. For example, one entity may be able to upload files but may not be able to perform matrix calculation. So it is a requisite to the model that the leader in the group knows every context in order to be able to calculate the trust information of the group for all such contexts available within the group. Thus, let's define how to calculate the trust value of the groups. Firstly, the reputation

that entity B has for entity A in a particular context C is represented in the following equation by notation $\delta_{a,b}^{C}$:

$$\delta_{a,b}^{C} = {}_{i}V_{a,b}^{C}, \tag{1}$$

where ${}_{i}V$ is the reputation value calculated for the interaction i using the available reputation model in the system. In this case ${}_{i}V_{a,b}^{C}$ is one record that represents the expectation that B will fulfill A's requests in context C for the interaction i. Trust and reputation values may be stored as many individual records of every contextualized interaction with the same or different entity. Thus, an entity may have a collection of different reputation values for other entities in the system. Then, an entity B may calculate the final reputation about an entity A for a particular context C using

$$\overline{\delta}_{a,b}^{C} = \frac{\sum_{i=1}^{j} {}_{i}V_{a,b}^{C}}{j} \quad \text{with } j > 0, \tag{2}$$

where $\overline{\delta}_{a,b}^{C}$ is the final reputation that entity B has for A in context C and j represents the amount of interactions that A and B have done in context C. Following, one entity may have as many contexts as it is programmed to. Then the final reputation regarding all contexts of one entity about another is given by the following expression:

$$\overline{K}_{a,b} = \frac{\sum_{i=1}^{x} \overline{\delta}_{a,b}^{C_i}}{x} \quad \text{with } x > 0, \tag{3}$$

where $\overline{K}_{a,b}$ is the final reputation value that entity B has for A for all contexts and x is the amount of all contexts that B knows about A. This way entity B is able to store all the reputation information about A in a given time period as one value. This value is used to perform the calculation of the trust value for the group. It is important to remember that reputation evolves with time, so this value can go up or down during time as B interacts with A.

In our model, we consider that what best represents the trust value of a group is the reputation that every entity within the group has about all entities of the group that it is part of. Then, the trust value can be calculated as an average reputation of all members inside the group. Considering this, the leader of the group receives and organizes the reputation values of the rest of the entities and computes the trust value of the group. For example, let's consider an example group of 5 members $G = \{M1, M2, M3, M4, M5\}$ and $M5$ is the agreed group leader. In this case, $M5$ asks $M1$ about the reputation information about $M2$, $M3$, $M4$, and $M5$. After that, $M5$ asks $M2$ about the reputation information of $M1$, $M3$, $M4$, and $M5$ and so on. Then, $M5$ uses this information to calculate the trust value of the group. It may sound that $M5$ may manipulate the reputation value that it receives, which is true. To avoid that particular case, we assume that the leader role is an important function in the group so the leader must not cheat; otherwise the whole system may fail if it is trust based.

It is important to remember that there is no common trust communication protocol for exchanging information in distributed systems. We assume in our model that the protocol to exchange information with the leader can be proactive, reactive, and hybrid depending on the scenario in which the protocol is deployed. Note that in some scenarios the members of groups already know who is the leader and then can proactively send such information. A new member can always ask for the leader of the group, so we assume that the node is able to find and communicate with the leader. Once this process is defined, the leader computes the final reputation of each entity as the average of the reputation values provided by the rest of entities within the group:

$$\overline{\omega}_{g}^{n} = \frac{\sum_{i=1}^{j} \overline{K}_{a,b}}{j} \quad \text{with } j > 0, \tag{4}$$

where $\overline{\omega}_{g}^{n}$ is the average reputation of entity n as seen by the rest of the entities of group g and j represents the quantity of members in group g. After performing the final average reputation of every entity in the group, the leader can generate the final trust value of the group using all reputation values computed before. This computation process is represented:

$$\overline{\lambda}_{g} = \frac{\sum_{i=1}^{x} {}_{i}\overline{\omega}_{g}^{n}}{x}, \quad \text{with } x > 0, \tag{5}$$

where $\overline{\lambda}_{g}$ represents the final trust value of the group g and x represents the quantity of members in the group. Once the leader has the trust value of the group, the leader can send this information to all members of the group.

In the case where there are many groups (N), every group leader can perform its own trust value calculation and inform its trust value to other group leaders. Every group leader has the responsibility to store group trust information, send it when asked, and distribute this value to new members, new leaders, and group outside requests as well. As seen the group role is very important, so the leader must be chosen carefully.

In order to create a common process to perform group calculation the algorithm represented in Figure 2 can be used.

5. Implementation and Analysis

The proposed model has been implemented and some statistics results have been obtained in order to validate it. In essence, a testbed has been set up by means of a P2P simulation tool [31] to create a basic group and to develop all calculation processes. The simulation tool uses asynchronous interactions between machines and different scenarios were simulated according to specific policies in the network. Some assumptions were defined in the test environment in order to organize the tests. For simplicity, the testbed is accomplished assuming that peers do not lie about trust and reputation values in the network. However this behavior can also be detected using the underline trust model used in the proposal. All participants are doing the same number of interactions in the testbed. The objective is to find standards and verify certain behaviors about trust and reputation values in the system.

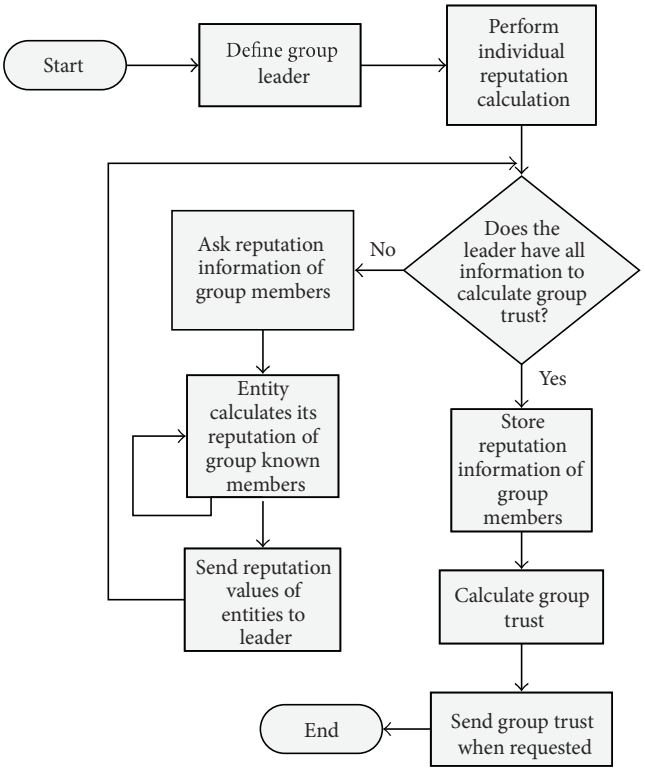


Figure 2: Algorithm for group trust calculation.

Table 2: Resumed interactions for test parameters.

Source peer	Destination peer	Time (s)	Speed (Kb/s)
Peer1	Peer2	0.121	3363
Peer3	Peer4	0.280	723
Peer1	Peer6	0.441	459
Peer2	Peer7	0.510	797
Peer6	Peer3	0.480	1221
Peer9	Peer4	0.701	1161
Peer10	Peer5	0.881	923
Peer6	Peer7	0.160	635
Peer7	Peer8	0.210	1937

Table 3: Probable situations considered.

Description	A-File load	B-integrity check	C
(1) File corrupted and on time	0.00	1.00	0.250
(2) File corrupted and a little delayed	0.00	0.50	0.125
(3) File corrupted and completely delayed	0.00	0.00	0.000
(4) File not corrupted and on time	1.00	1.00	1.000
(5) File not corrupted and a little delayed	1.00	0.50	0.875
(6) File not corrupted and completely delayed	1.00	0.00	0.750

When interactions between entities correspond (or not) to the expected behavior, it can be determined if the peer is trustworthy (or not). Some behavior patterns have been delimited as desirable in the system. Firstly, there are no errors in the communication transmission; secondly, the time for transmitting a file is determined by the quality level of the transmission. These parameters have been chosen in order to simplify the P2P environment, thus permitting the focus of the analysis on trust and reputation values considered in the interactions of peers and performing the calculation of the group trust value.

The testbed is executed in machines with JXTA Shell [31] installed and configured. The simulated environment was composed of 500 nodes. These were defined as 5 different groups with a hundred nodes in each group. Each node only performs interactions within its group. Also, each peer performs at least 20 interactions in the network. The network topology is simple and uses 2 common layer switches. The purpose of this topology is to represent a P2P network connected directly to a LAN. The peers are configured in the same network segment with no additional hops. Each peer uses a different TCP port. This characteristic is to permit the P2P network to establish connections on different ports.

The simulation testbed considered that the transmission delay and the integrity of the file are the parameters used to decide whether a peer is malicious or not. This means that the peer can send a corrupted file, delay its sending to another peer, or perform both. The interval of time values for the transmission is defined after some file transfer tests were executed. Several interactions have been fulfilled for a file with fixed size (100 Kbytes), and a standard time could be defined for a successful interaction.

Table 2 has some summarized definitions of the test interactions to limit the values expected. Based on this information it has been determined that the expected transference time of a file is up to 1 s, a short delay would be between 1 s and 2 s, and a completely delayed is above 2 s. The other parameter defined is the integrity of the received file. A hash calculation is used to verify this condition.

Once this parameter is defined, the file load times are parametrically determined by the variable a, the file integrity check times are determined by the variable b, and the variable c determines the reputation feedback for the trust model associated with the given interaction. Table 3 shows the parameters used in our testbed to define how to infer some reputation for a peer.

The reputation value c is determined by the following equation, where the parameter P represents the weight (importance) that the network administrator allocates to the integrity of the file:

$$C = ((P \times a) + ((1 - P) \times b)). \quad (6)$$

Related to peers behavior, peers only accomplished interactions with appropriate parameters to verify the convergence of trust and reputation values. The underlying trust model used in the testbed is TRAVOS [7], which allows peers to

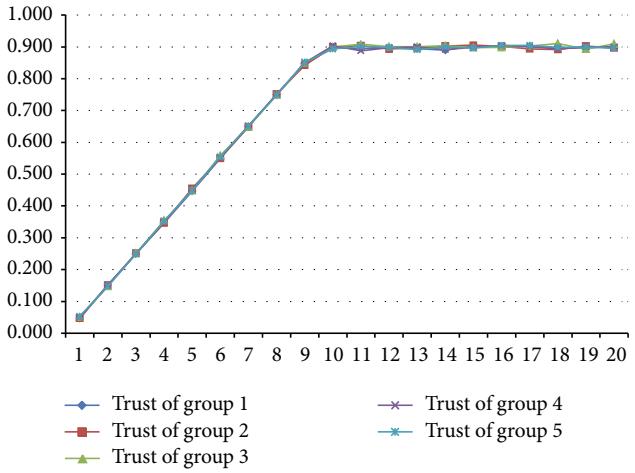


FIGURE 3: Group trust of an ideal environment.

realize that some members of the network changed their behavior. The testbed has been set up in order to have peer1 as leader of each group. Once all these parameters were set up, we defined different scenarios for our tests. Direct trust of the peers and reputation values based on context of the groups are calculated in the simulations in order to calculate group trust. The tested scenarios, its results, and analysis are presented in the following subtopics. In all graphs the x-axis is the number of interactions and the y-axis is the correspondent trust value in each round.

5.1. Scenario 1: All Peers Behave Accordingly. In this test all the peers in all groups behave as expected. This means that they fulfill their requirements and perform their defined context correctly. Note that the entire peer acts following the same behavior, without changing any aspect of its functional context. In this case, the trust value of the group is considered extremely trustworthy and it tends to stabilize in a value near 1, thus avoiding blind trust. When there is no malicious peer in the network, the trust value of the group reflects the individual behavior of the peers in the group. This is considered the ideal world. Figure 3 shows this result.

5.2. Scenario 2: Random Behavior of Peers. In this test all peers in all groups behave randomly after round 4. This means that it is not known for certain by the other group members whether a particular peer behaved accordingly or not. This test was set up in order to verify the results when nodes behave in a proper manner sometimes and then change their behavior with no particular reason. This can be considered the worst environment imaginable because it cannot be possible to predict if a node will or will not behave accordingly. This scenario is represented in Figure 4.

5.3. Scenario 3: Random Behavior of 20 Peers in Group 1. In this test 20% of group 1 behaves randomly. This test simulates a coalition of peers in order to modify the group trust. Such behavior is considered as if the nodes suffer some kind of attack or there are peers acting as black holes in the P2P environment. When 20% of the peers start behaving in a malicious manner, the trust value of the group decreases and tends to stabilize in a value near 0.8. The analysis shows that the increase of the trust coefficients provided by the good peers overcomes the decrease of the coefficient of the malicious peers. In this case the group is still considered trustworthy ($\overline{\lambda}_g > 0.7$) despite having malicious members, as represented in Figure 5.

5.4. Scenario 4: Random Behavior of 40 Peers in Group 1. In this test 40% of group 1 behaves randomly. This test simulates a situation where the P2P network is compromised and there is no guarantee that the peers in this group are trustworthy or not. When 40% of the group members are malicious, the trust coefficient of the group tends to stabilize in a value near 0.6 which represents that the peers in the group are not trustworthy, and thus the group is not considered trustworthy because of the threshold. Figure 6 shows this result.

5.5. Scenario 5: Random Behavior of 60 Peers in Group 1. In this test 60% of group 1 behaves randomly; then the trust coefficient tends to stabilize in a value near 0.5, also making the group untrustworthy because of the threshold, as seen in Figure 7.

5.6. Group 1 Analysis. This test was performed to analyze group 1, compiled in one graph, as seen in Figure 8. When peers change their behavior, the trust value of the group decreases, thus making a group with constant behavior change to be untrustworthy considering a defined threshold of $\overline{\lambda}_g = 0.7$. When nodes behave randomly, the value of the group trust tends to 0.5.

The reader may realize that the individual behavior of each member in the group influences the trust value of the group as a whole. The results can also be considered satisfactory because all the peers are initiated at the same time in the network and interact with each other the same number of times.

These results also show that the trust value of group 1 in the first scenario is originally high (moment in which all the peers have good behavior). After that, it starts to decrease in the moment the peers in the group change their behavior or acts forming a coalition. As a result, the group trust model can be used as a parameter to interact or not with a specific group.

6. Conclusion

This work has reviewed different trust and reputation models in distributed systems. We developed a model as an extension to support the calculation of trust values of groups of entities. The proposed model has been validated in a P2P simulation tool. Our results show that it is possible to generate and to calculate group trust behavior in distributed systems.

We consider that it is important that a trust leadership based algorithm or a trust consensus algorithm should be better studied in order to create leaders in groups in a distributed manner. It is also important to define a trust protocol

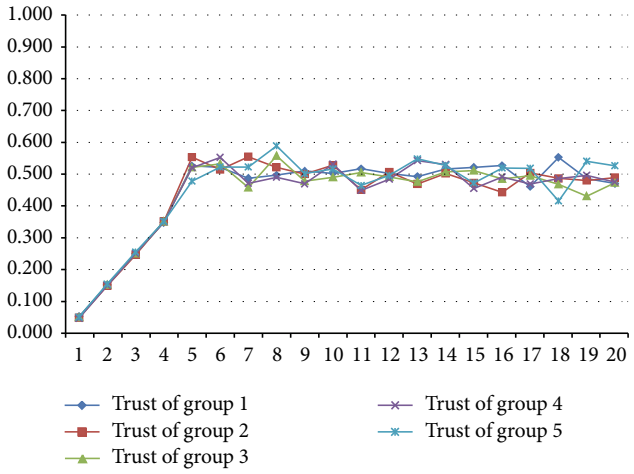


Figure 4: Group trust of the worst environment.

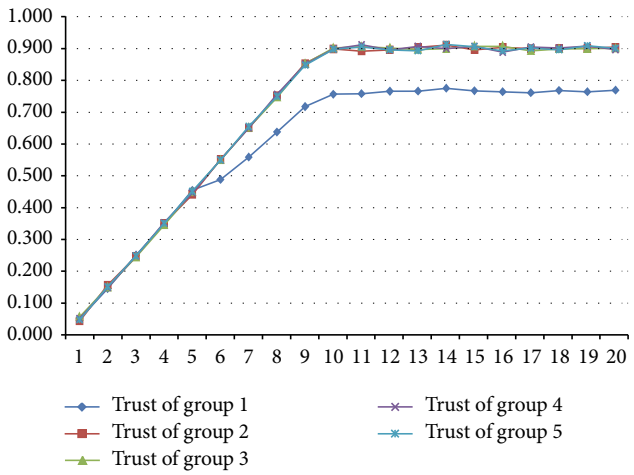


Figure 5: Group trust of G1 when 20 nodes change their behavior.

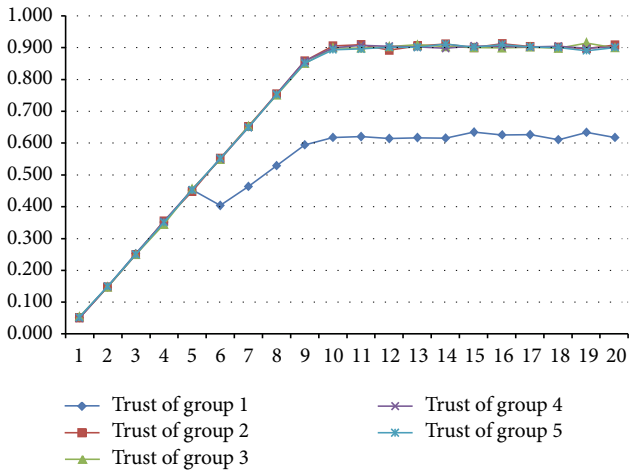


Figure 6: Group trust of G1 when 40 nodes change their behavior.

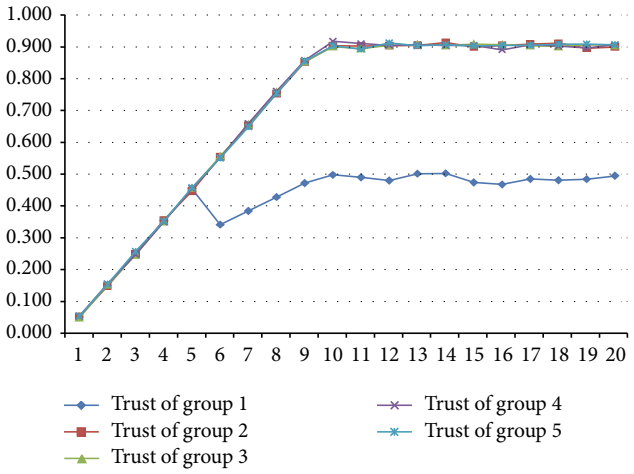


Figure 7: Group trust of G1 when 60 nodes change their behavior.

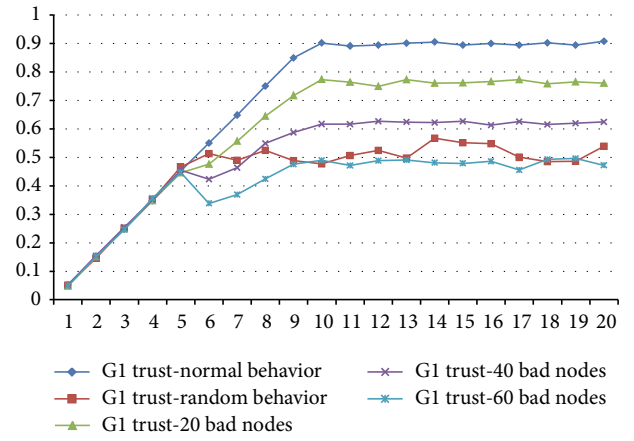


Figure 8: Group 1 synthesis.

as a platform to support trust based communications. We consider that as research areas that can be deeply studied.

Using the concept of group trust, the proposed model in this paper can be used in bigger and more complex distributed systems architectures. As future work we will implement our group trust model in software agents, grid platforms, or cloud environments in order to evaluate its behavior in bigger systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Part of the computations of this work was performed in EOLO, the HPC of Climate Change of the International Campus of Excellence of Moncloa, funded by MECD and MICINN. The first author acknowledges the Laboratory for

Decision Technologies at the University of Brasilia (LATITUDE/UnB) for its support to this work.

References

[1] M. J. Covington and R. Carskadden, "Threat implications of the internet of things," in *Proceedings of the 5th International Conference on Cyber Conflict (CyCon '13)*, pp. 1–12, Tallinn, Estonia, June 2013.

[2] Y. B. Saieda, A. Olivereaua, D. Zeghlache, and M. Laurent, "Trust management system design for the internet of things: a context-aware and multi-service approach," *Computers & Security B*, vol. 39, pp. 351–365, 2013.

[3] S. Jingshan, Y. Jiabin, and Y. Fengshou, "Grid trust model based on last service and hierarchy," *Journal of Nanjing University of Aeronautics and Astronautics*, vol. 43, no. 2, pp. 273–278, 2011.

[4] Y. Liu, K. Li, Y. Jin, Y. Zhang, and W. Qu, "A novel reputation computation model based on subjective logic for mobile ad hoc networks," *Future Generation Computer Systems*, vol. 27, no. 5, pp. 547–554, 2011.

[5] F. Gómez Mármol and G. Martínez Pérez, "Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems," *Computer Standards and Interfaces*, vol. 32, no. 4, pp. 185–196, 2010.

[6] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks," in *Proceedings of the 3rd European Symposium on Research in Computer Security*, pp. 3–18, Springer, London, UK, 1994.

[7] J. Patel, *A trust and reputation model for agent-based virtual organisations [Ph.D. thesis]*, Electronics and Computer Science, Faculty of Physical Sciences and Engineering, University of Southampton, Southampton, UK, 2007.

[8] D. Gambetta, *Can We Trust Trust?* chapter 13, Department of Sociology, University of Oxford, 2000.

[9] S. P. Marsh, *Formalising trust as a computational concept [Ph.D. thesis]*, Department of Computing Science and Mathematics, University of Stirling, Stirling, UK, 1994.

[10] T. Zhao and S. Dong, "Trust-GSM: a trust aware security model for multi-domain grid," in *Proceedings of the 5th Annual ChinaGrid Conference (ChinaGrid '10)*, pp. 43–47, Guangzhou, China, July 2010.

[11] S. Naqvi and P. Mori, "Security and trust management for virtual organisations: GridTrust approach," in *IFIP Advances in Information and Communication Technology*, pp. 306–309, 2009.

[12] C. Huang, H. Hu, and Z. Wang, "A dynamic trust model based on feedback control mechanism for P2P applications," in *Autonomic and Trusted Computing*, L. T. Yang, H. Jin, J. Ma, and T. Ungerer, Eds., vol. 4158 of *Lecture Notes in Computer Science*, pp. 312–321, Springer, Berlin, Germany.

[13] W. Wang, G. Zeng, and L. Yuan, "Ant-based reputation evidence distribution in P2P networks," in *Proceedings of the 5th International Conference on Grid and Cooperative Computing (GCC '06)*, pp. 129–132, Hunan, China, October 2006.

[14] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, Boston, Mass, USA, 1st edition, 2001.

[15] W. Stallings, *Data and Computer Communications*, Prentice Hall, New York, NY, USA, 7th edition, 2004.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 640–651, ACM, New York, NY, USA, May 2003.

[17] B.-J. Chang, S.-L. Kuo, Y.-H. Liang, and D.-Y. Wang, "Markov chain-based trust model for analyzing trust value in distributed multicasting mobile Ad Hoc networks," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference*, pp. 156–161, December 2008.

[18] S. Buchegger and J. Y. le Boudec, "A robust reputation system for P2P and mobile Ad-hoc networks," in *Proceedings of the 2nd Workshop on the Economics of P2P Systems*, pp. 156–161, June 2004.

[19] S. Buchegger and J. Y. le Boudec, "Performance analysis of the CONFIDANT protocol," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '02)*, pp. 226–236, ACM, New York, NY, USA, June 2002.

[20] F. Almenrez, A. Marn, C. Campo, and C. Garca, "PTM: a pervasive trust management model for dynamic open environments," in *Proceedings of the 1st Workshop on Pervasive Security and Trust*, 2004.

[21] G. Zacharia, "Trust management through reputation mechanisms," *Applied Artificial Intelligence*, vol. 14, no. 9, pp. 881–907, 2000.

[22] S. Songsiri, "MTrust: a reputation-based trust model for a mobile agent system," *Proceedings of the 3rd International Conference on Autonomic and Trusted Computing*, Springer, Berlin, Germany, vol. 4158, pp. 374–385, 2006.

[23] J. Sabater and C. Sierra, "Regret: reputation in gregarious societies," in *Proceedings of the 5h International Conference on Autonomous Agents (AGENTS '01)*, pp. 194–195, ACM, New York, NY, USA, June 2001.

[24] J. Carbo, J. M. Molina, and J. Davila, "Trust management through fuzzy reputation," *International Journal of Cooperative Information Systems*, vol. 12, no. 1, pp. 135–155, 2003.

[25] T. Largillier and J. Vassileva, "Using collective trust for group formation," in *Proceedings of the 18th International Conference on Collaboration and Technology*, pp. 137–144, Springer, Berlin, Germany, 2012.

[26] S. Al-Oufi, H. N. Kim, and A. El Saddik, "A group trust metric for identifying people of trust in online social networks," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13173–13181, 2012.

[27] R. Levien, "Advogato's trust metric," 2000, http://www.advogato.org/trust-metric.html.

[28] F. R. Easa, A. G. Bafghi, and H. Shakeri, "A group-based trust propagation method," in *Proceedings of the 2nd International Conference on Computer and Knowledge Engineering*, pp. 313–317, October 2012.

[29] T. Schulz and I. Tjstheim, "Increasing trust perceptions in the internet of things," in *Aspects of Information Security, Privacy, and Trust*, L. Marinos and I. Askoxylakis, Eds., vol. 8030 of *Lecture Notes in Computer Science*, pp. 167–175, Springer, Berlin, Germany, 2013.

[30] J. P. Wang, S. Bin, Y. Yu, and X. X. Niu, "Distributed trust management mechanism for the internet of things," *Applied Mechanics and Materials*, vol. 347, no. 350, pp. 2463–2467, 2013.

[31] "JXTA—get connected," 2013, http://jxta.free.fr/.

Arquitectura de Seguridad Multinivel: Una Guía para las Organizaciones Modernas

Robson de Oliveira Albuquerque[1,2], Fábio Buiati[1,2], Luis Javier García Villalba[1]
[1]Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases s/n, Ciudad Universitaria, 28040 Madrid, España
Email: {robson, fabio, javiergv}@fdi.ucm.es

[2]Faculdade de Tecnologia, Universidade de Brasília (UnB)
Curso de Engenharia de Redes de Comunicação, Departamento de Engenharia Elétrica
CEP: 70910-900 - Brasília -DF - Brasil
Email: {robson, fabio.buiati}@redes.unb.br

***Resumen*—La información puede considerarse como el activo más importante de cualquier organización moderna. Garantizar la seguridad de esta información implica preservar la confidencialidad, la integridad y la disponibilidad de la misma, tríada conocida como CIA en inglés. Este trabajo presenta una arquitectura de seguridad multinivel motivado por la necesidad de considerar la información desde diferentes puntos de vista con el fin de protegerla. Además, se sugiere una nueva clasificación de los elementos de información, operaciones, entidades y componentes que se pueden integrar para mostrar las distintas fuentes de riesgos al tratar con información sensible. Se muestra también una visión general de cómo se trata y se representa actualmente la información y por qué es tan difícil garantizar la seguridad en todos los aspectos del tratamiento de la información.**

Palabras clave*—Arquitectura, confianza, seguridad de la información. *(Architecture, trust, information security)

I. Introducción

La gestión de seguridad de la información es fundamental en cualquier organización. Aun así, son muy pocos los modelos formales que ayudan a proteger eficazmente la información. Una manera de tratar el problema de la seguridad de la información es gestionar los riesgos desde diferentes puntos de vista. Estos riesgos están asociados a fenómenos naturales, riesgos tecnológicos y riesgos humanos [4]. Teniendo en cuenta estos aspectos, este trabajo propone una arquitectura multinivel para la gestión de riesgos de seguridad en las organizaciones modernas. Este trabajo está organizado en 7 secciones, siendo la primera la presente introducción. La Sección II recoge los trabajos relacionados más representativos. La Sección III propone una arquitectura de seguridad multinivel. La Sección IV presenta un modelo de confianza para la arquitectura multinivel. Por último, la Sección V muestra las principales conclusiones que se extraen de este trabajo.

II. Trabajo Relacionado

Mucho se ha dicho sobre normativas y estándares en seguridad de la información y sobre la importancia de su uso. Las normas de seguridad sirven como una guía para el desarrollo de un sistema de gestión de seguridad de la información.

Normas como la BS7799 e ISO 27000 [5] son guías ampliamente reconocidas en el área de la seguridad de la información. Plataformas como ITIL y COBIT [6] son utilizadas también en la administración de las tecnologías de la información con el fin de guiar a las organizaciones a aumentar su productividad y, en algunos aspectos, ayudan a mantener la seguridad de la información en términos de organización y metodología [7].

Sin embargo, el cumplimiento de las normas no garantiza en absoluto la seguridad. Para hacer frente a la seguridad de la información se requiere ir más allá del cumplimiento de normas o de mejores prácticas.

Respecto a las arquitecturas de seguridad de la información, el Zero Trust Model for Cybersecurity [8] sostiene un mensaje muy claro: dejar de confiar en los paquetes de datos como si fuesen personas. La idea subyacente es que el concepto de redes internas y externas debe cambiarse porque uno asume que todo el tráfico no es de confianza. Zero Trust viene a decir que los datos internos deben ser protegidos contra abusos procedentes de la red interna y que los datos externos deben ser protegidos en las redes públicas.

[9] señala que existe una necesidad de mejorar la seguridad de la información a nivel administrativo y organizacional. Por su parte, [11] [10] advierten de un cambio en la manera de cómo las personan se relacionan con la seguridad de la información, convirtiéndose además en el centro del problema.

Con el fin de proteger la información, es muy importante entender la forma en que se trata en el mundo digital. Desde la perspectiva del usuario, la información puede ser un texto, una imagen o una combinación de ambos. Internet redefinió la forma de representarla y de recuperarla [12]. La representación de la información requiere de complementos estructurales o semánticas adicionales, que transforman los datos en algo significativo para los seres humanos.

Considerando todo lo expuesto anteriormente, las arquitecturas de seguridad actuales no logran gestionar los riesgos, las políticas, las personas y los activos de forma correcta. Para intentar paliar esta carencia, este trabajo propone una arquitectura de seguridad de información multinivel que trata

ISBN: 978-84-9717-323-0

de conectar todas las piezas entre sí respecto a la seguridad de la información. La especificación del modelo en niveles es importante para ver cómo todos los elementos de la arquitectura de seguridad interactúan.

III. Arquitectura de Seguridad de la Información

La forma en la que vemos la seguridad está basada en una arquitectura multinivel. En este enfoque cada elemento es una pieza del rompecabezas que debe estar bien conectada, de forma que la seguridad de información pueda ser vista como un todo indivisible. La Figura 1 ilustra la arquitectura de seguridad de la información propuesta con sus niveles.

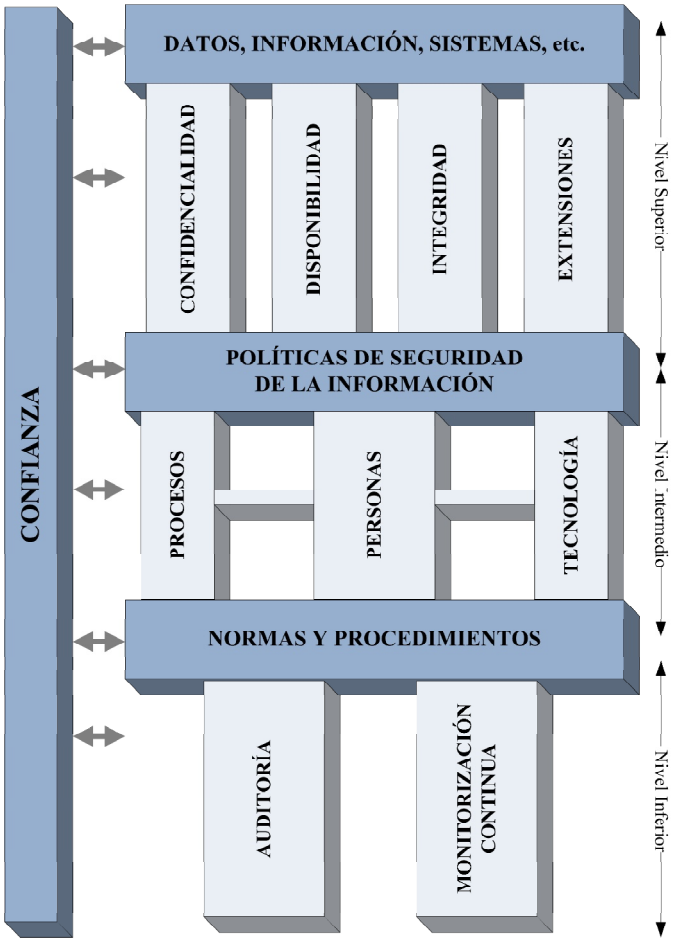


Figura 1. Arquitectura de Seguridad Multinivel

III-A. Nivel Superior

El nivel superior es la base para empezar a pensar en la seguridad de información de cualquier organización. Sin la adecuada comprensión de lo que son los datos, la información, los activos de información, etc., no hay cómo hablar de seguridad de la información, simplemente porque uno no sabe qué hay que proteger. Es importante señalar que el enfoque de "proteger todo" no es eficaz y es, además, bastante costoso.

En general, en este nivel es donde se localizan los datos importantes o con valor para las organizaciones o las personas. Considerando la importancia que tienen los datos actualmente, una gran cantidad de información se puede recuperar a partir de los datos y los sistemas de información. Utilizando herramientas y técnicas adecuadas, es posible crear además nuevos conocimientos a partir de los datos que, a simple vista, no parecen tener ningún sentido.

Cuando se trata de activos de información es muy importante que estos sean identificados y etiquetados, y la relación con la información debe ser claramente entendida por la organización.

Las redes de comunicación conectan los datos, la información y sus activos para que cualquier persona con acceso autorizado pueda explorarlas. Quién controla (personas) o cómo se controla (proceso, hardware o software) la red es lo que la hace peligrosa o no. Así que la creación de perímetros de redes, políticas y otros mecanismos de defensa sigue siendo una forma de controlar lo que entra y sale de la red. El uso de estos mecanismos es clave para entender lo que sucede en la transmisión dentro de los sistemas de información.

También en este nivel la seguridad tiene como foco salvaguardar la confidencialidad, la integridad y la disponibilidad de la información, debiendo aplicarse de forma efectiva en toda la cadena. La confidencialidad se refiere a la limitación de acceso a la información y a la divulgación a los usuarios autorizados. La integridad se refiere a la fiabilidad de los recursos de información, es decir, que los datos no han sido modificados inapropiadamente, ya sea por accidente o deliberadamente. Por último, la disponibilidad se refiere a la disponibilidad de los recursos de información.

Las extensiones de seguridad de la información son nuevos atributos o propiedades que protegen la información y los sistemas, pero no se limitan a ellos. La autenticación, el control de acceso, el no repudio, la privacidad, el anonimato y la autorización son servicios que se caracterizan como extensiones de seguridad.

III-B. Nivel Intermedio

Siguiendo un recorrido descendente nos encontramos con este nivel que es la parte de la arquitectura que nos ayudará a definir cuestiones tales quién, cómo, por qué y qué tecnologías pueden utilizarse para garantizar la seguridad de la información en el nivel superior. Los siguientes ítems son analizados: políticas de seguridad, procesos, personas y tecnología.

Una política de seguridad de la información es un documento de alto nivel que describe los requisitos o reglas que se deben cumplir para garantizar la seguridad de la información en una organización. En general, esta política es muy específica y cubre una única organización. La política de seguridad también está relacionada con los problemas de gestión y de control de la información, una vez que la protección de la misma está directamente relacionada con la cultura de la organización.

La política de seguridad debe explicar la necesidad de la seguridad de la información para todos los usuarios dentro de la organización y complementar los objetivos de la organización,

siendo necesario que esté alineada con el plan estratégico de la organización [13].

En la seguridad de la información los procesos son una manera formal de identificar, medir, gestionar y controlar los riesgos relacionados con la información o su valor para la organización. Los procesos incluyen mecanismos formales e informales (grandes o pequeños, simples o complejos, ...) para hacer las cosas y proporcionar un vínculo vital para todas las interconexiones dinámicas.

Las personas son el principal bloque del rompecabezas y representan el recurso humano. En general, una persona diseña e implementa cada parte de la política de seguridad, crea y mantiene los procesos, los activos de información, la tecnología utilizada, etc. Los problemas de seguridad afectan a las personas, sus relaciones, sus valores y sus comportamientos. Cuando se trabaja con seguridad de la información es importante hacer frente a puntos como las estrategias relacionadas con la contratación, el acceso, las responsabilidades, la formación, el despido, las sanciones y todo lo que sea importante abordar para ayudar a mantener la estrategia de seguridad de la información de la organización.

La tecnología es el elemento del rompecabezas constituido por un conjunto de sistemas de información, aplicaciones, herramientas, infraestructura y mecanismos de defensa que la organización utiliza para llevar a cabo su misión de proteger la información. Los elementos tecnológicos son susceptibles a frecuentes cambios y actualizaciones y pueden hacerse obsoletos rápidamente. La tecnología puede ser la parte fundamental de una infraestructura de la organización. La tecnología se usa también para resolver las amenazas de seguridad y los riesgos.

Es muy importante tener en cuenta que la tecnología por sí misma no hace nada. Debe ser vista como una parte de un sistema complejo que tiene necesidades específicas para proteger lo que es valioso en la organización. Además, la tecnología debe trabajar conjuntamente con personas y procesos completando un ciclo, todos ellos guiados por la política de seguridad de la información de la organización.

III-C. Nivel Inferior

Este nivel trata de las actividades diarias y las medidas que se deben adoptar en caso de un problema específico. Las prácticas de seguridad son guías para mantener la información segura. Sin embargo, las normas, procedimientos de monitorización y auditoría dan a los administradores las herramientas necesarias para ayudarles a mantener la información, los activos, las redes, los sistemas, etc., más seguros. Los siguientes ítems son analizados: normativas de seguridad, auditoría y monitorización continua.

Básicamente, una normativa define cómo deberían ser las cosas y cómo hay que valorarlas. También tiene que ver con la forma de clasificar las acciones en correctas o equivocadas. Las normativas son primordiales para la priorización de los objetivos y para definir cómo se deben hacer las cosas.

La auditoría de la seguridad de la información es un proceso que determina la valoración cualitativa y cuantitativa del estado actual del sistema analizado según criterios específicos de seguridad de la información. El proceso de auditoría es clave para encontrar riesgos, fallos técnicos, políticas, procedimientos y problemas normativos en una organización. Hay que tener en cuenta que la auditoría es un proceso que nunca termina. Cuando se realiza la auditoría, uno debe estar preparado para abarcar temas desde seguridad física de los centros de datos hasta la seguridad lógica, incluyendo los perímetros de red, la configuración del sistema y los sistemas de información.

Otra de las tareas realizadas en este nivel es la monitorización continua. Se trata de una actividad de mantenimiento de los conocimientos de seguridad de la información, vulnerabilidades, amenazas y riesgos asociados [14]. Es un punto clave de apoyo a la toma de decisiones relativas a la gestión de riesgos de una organización.

La monitorización continua se inicia definiendo qué, cómo, por qué y cuándo monitorizar los activos de información o cualquier parte de la arquitectura. Se apoya en tecnología, procesos, procedimientos, entornos operativos y personas. También ayuda en el establecimiento de prioridades y gestiona el riesgo de forma coherente en toda la organización.

IV. Confianza

Desde el punto de vista de la seguridad de la información, la confianza puede tener un valor de cero o de uno. Uno confía o no en sus sistemas de información, redes, activos, etc. El “tal vez” debe evitarse a toda costa. Por lo general, la confianza se adquiere mediante la observación empírica, por prueba formal de los sistemas, etc. [15].

La confianza y la seguridad están estrechamente relacionadas [15]. Si se consideran los objetivos de seguridad, está claro que los aspectos de confianza están conectados con la seguridad ya que mantener la información segura depende de las personas, las extensiones de seguridad (autenticación, autorización, control de acceso, no repudio, etc.).

Considerando lo anteriormente expuesto, no se puede proteger la información sin ser capaz de comprender todo el ciclo de vida que tiene la información. Hay que tener en cuenta una visión detallada si se desea más seguridad en el sistema; uno debe ser capaz de representar, procesar y utilizar la información en un entorno donde las personas, la tecnología, los activos de información, el hardware, el software, etc., están conectados entre sí. Y, paralelamente, hay que tomar medidas de seguridad para garantizar su protección. Ahí es donde la arquitectura de seguridad de la información multinivel con confianza entra en escena porque sólo proteger una parte de la información se ha demostrado ineficaz, como se ha visto recientemente [1][2].

La confianza en general es parte del rompecabezas cuando hay un conocimiento suficiente de la información, los sistemas, la tecnología y los demás componentes que ayudan hacer afirmaciones como “totalmente seguro” o la información es segura porque se cumple alguna condición en particular. Esta arquitectura en niveles le permite a uno hacer frente a determinados componentes y aislar problemas relacionados con cada uno de ellos.

V. Conclusiones

La tarea de garantizar la seguridad de la información no es un fin en sí mismo; es un medio para lograr un fin [16]. Se trata también de un tema en constante evolución, debido a la creciente magnitud y complejidad de las amenazas de seguridad de la era digital. Como se observa en la actualidad, el campo de investigación de la seguridad de información es cada vez más importante porque el mundo está interconectado con redes de comunicación que se utilizan para la transmisión de información crítica y sensible.

En este trabajo se ha introducido una arquitectura de seguridad multinivel donde los elementos de seguridad de la información están interconectados siendo útiles para la gestión de riesgos en los diferentes niveles de la organización. De esta forma, la seguridad de la información puede ser vista como un todo.

Gobierno, organizaciones y empresas que consideran la gestión de seguridad de la información necesitan un enfoque sistemático para abordar de manera coherente la seguridad en cada nivel, disminuyendo así los riesgos de administración y mejorando la eficiencia de la gestión de la seguridad. Bajo esta perspectiva, la arquitectura de seguridad de la información en niveles puede ser utilizada como una guía para obtener mejores resultados en la protección de la información.

Agradecimientos

Los autores también agradecen el apoyo proporcionado por el Laboratorio de Tecnologías de Decisión de la Universidad de Brasilia (LATITUDE / UnB). Asimismo, Fábio quiere agradecer la financiación que le brinda el Programa Nacional de Post-Doctorado de Brasil (PNPD/CAPES). El Grupo de Investigación GASS agradece la infraestructura proporcionada por el Campus de Excelencia Internacional (CEI) Campus Moncloa - Clúster de Cambio Global y Nuevas Energías (y, más concretamente, el sistema EOLO como recurso de computación de alto rendimiento HPC - High Performance Computing), infraestructura financiada por el Ministerio de Educación, Cultura y Deporte (MECD) y por el Ministerio de Economía y Competitividad (MINECO).

Referencias

[1] G. Greenwald, E. MacAskill, L. Poitras, "Edward Snowden: the whistleblower behind the NSA surveillance revelations," *The Guardian*, Vol. 9, 2013.

[2] J.-T. Richelson, "The Snowden Affair. Web Resource Documents the Latest Firestorm over the National Security Agency," *National Security Archive Electronic Briefing Book*, No. 436, 2013.

[3] Gartner Press Release. "Gartner Says Cloud-Based Security Services Market to Reach 2.1 Billion dollars in 2013", Stamford, Conn., 2013. Disponible en ttp://www.gartner.com/newsroom/id/2616115.

[4] B. Blakley, E. McDermott, D. Geer, "Information security is information risk management," *ACM Proceedings of the Workshop on New security Paradigms*, pp. 97–104, 2001.

[5] M. Whitman, H. Mattord, "Management of Information Security," *Cengage Learning, Fourth Edition*, 2013.

[6] R. Parvizi, F. Oghbaei, S. R. Khayami, "Using COBIT and ITIL frameworks to establish the alignment of business and IT organizations as one of the critical success factors in ERP implementation," *5th Conference on Information and Knowledge Technology (IKT)*, 2013 pp. 274–278, 2013.

[7] Department of Communications, Information Technology and the Arts and the Trusted Information Sharing Network. "Secure Your Information: Information Security Principles for Enterprise Architecture," *Report, Australia*, 2007.

[8] The National Institute of Science and Technology (NIST), "Developing a Framework to Improve Critical Infrastructure Cybersecurity. Submitted by Forrester Research. In Response to RFI# 130208119-3119-01", 2013. Disponible en http://csrc.nist.gov/cyberframework/rfi_comments/040813_forrester_research.pdf.

[9] R.-M. Ahlfeldt, P. Spagnoletti, G. Sindre. "Improving the Information Security Model by using TFI", *In the New Approaches for Security, Privacy and Trust in Complex Environments. IFIP International Federation for Information Processing*, Vol. 232, pp. 73–84, 2007.

[10] R. Blakley, A. Johnston, P. Lowry, Q. Hu, M. Warkentin, R. Baskerville, "Future directions for behavioral information security research," *Computers & Security*, Volume 32, February 2013, pp. 90–101.

[11] S. Aurigemma, R. Panko, "A Composite Framework for Behavioral Compliance with Information Security Policies", *In proceedings of the 45th Hawaii International Conference on System Sciences. IEEE Computer Society*, 2012.

[12] H. Chu, "Information Representation and Retrieval in the Digital Age", *Information Today, Inc*, Second Edition, 2010.

[13] ISACA. "An Introduction to the Business Model for Information Security", 2009, Disponible en http://www.isaca.org/Knowledge-Center/BMIS/Documents/IntrotoBMIS.pdf.

[14] The National Institute of Science and Technology (NIST), "Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations", *NIST Special Publication*, pp. 800–137, 2011.

[15] P. Lamsal, "Understanding Trust and Security", *Department of Computer Science. University of Helsinki, Finland*, 2001.

[16] T. Peltier, "Information security fundamentals," *CRC Press*, 2013.

Sensors **2014**, *14*, 22754-22772; doi:10.3390/s141222754

OPEN ACCESS
sensors
ISSN 1424-8220
www.mdpi.com/journal/sensors

Article

A Layered Trust Information Security Architecture

Robson de Oliveira Albuquerque [1,2], Luis Javier García Villalba [1,*], Ana Lucila Sandoval Orozco [1], Fábio Buiati [2] and Tai-Hoon Kim [3]

[1] Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Information Technology and Computer Science, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain; E-Mails: robson@fdi.ucm.es (R.O.A.); asandoval@fdi.ucm.es (A.L.S.O.)

[2] Electrical Engineering Department, University of Brasilia, Campus Universitário Darcy Ribeiro, Asa Norte, 70910-900 Brasilia, DF, Brazil; E-Mail: fabio@fdi.ucm.es

[3] School of Engineering and ICT, University of Tasmania, Private Bag 87, Hobart, TAS 7001, Australia; E-Mail: taihoonn@daum.net

* Author to whom correspondence should be addressed; E-Mail: javiergv@fdi.ucm.es; Tel.: +349-1394-7638; Fax: +349-1394-7547.

External Editor: Neal N. Xiong

Received: 5 October 2014; in revised form: 14 November 2014 / Accepted: 25 November 2014 / Published: 1 December 2014

Abstract: Information can be considered the most important asset of any modern organization. Securing this information involves preserving confidentially, integrity and availability, the well-known CIA triad. In addition, information security is a risk management job; the task is to manage the inherent risks of information disclosure. Current information security platforms do not deal with the different facets of information technology. This paper presents a layered trust information security architecture (TISA) and its creation was motivated by the need to consider information and security from different points of view in order to protect it. This paper also extends and discusses security information extensions as a way of helping the CIA triad. Furthermore, this paper suggests information representation and treatment elements, operations and support components that can be integrated to show the various risk sources when dealing with both information and security. An overview of how information is represented and treated nowadays in the technological environment is shown, and the reason why it is so difficult to guarantee security in all aspects of the information pathway is discussed.

Keywords: information security; information treatment; risk management; security architecture; trust

1. Introduction

Managing information security is critical. Even so, very few formal models are able to secure information. After what happened regarding the information disclosed by Mr. Snowden [1,2], what already seemed very difficult in terms of security, controls, policies, and so on, became much more problematic. Regarding disclosure and technologies that could be used, research on information security has already acknowledged many possibilities, but its level was beyond expectations. The fact is that it is no longer possible to approach information security naively.

The power behind security agencies and information security systems have proven to the world to be something that only security experts were able to understand before. Now, the truth is everywhere, so people can perceive it by themselves. There is a clear shift in the world of security that is taking place after Snowden's information disclosure. As a consequence, information security compliance has been proven to be ineffective.

Having such concerns in mind, up to the present day, discussions about this subject seem endless. Consequently, the information security research field faces new challenges in terms of confidentiality, privacy, anonymity, plausible deniability, and so on.

The world of security is a billion dollar industry [3], only considering cloud solutions. Nevertheless, the problem is that no matter how much money you pay for security, it may never be enough to avoid data theft, information disclosure, system exploitation and other risks that are part of the information technology security context. There is also an understanding that, usually, information security is not properly understood by organizations, because most projects start with no security approach to the problem itself, nor information security related to the whole project.

One way to deal with the information security problem is to manage those risks from different points of view. Risks that can compromise enterprises or governments are associated with natural phenomena, technological risks and human-related risks [4]. Therefore, we propose a suitable and flexible trust information security architecture (TISA) in order to manage risks and security. In summary, the main contributions of this paper are the following: (1) it reviews current information security aspects; (2) it proposes a new-layered trust information security architecture considering several interconnected elements; (3) it introduces a trust layer encompassing all architectural levels; and (4) it proposes information security elements, operations and components for managing risks at different levels considering information treatment.

This paper is organized into the following sections. Section 2 presents relevant reviews and related works in this field. Section 3 presents and discusses the proposed trust information security architecture. Section 4 provides a comparison of our proposal, TISA, with others regarding information security models. Section 5 mentions the information treatment and representation and its connection to the trust information security architecture proposed. Section 6 concludes this paper and points out future works.

2. Related Work

A lot has been said about norms, compliance, standards and frameworks regarding information security and the importance of following some of those. Commonly, security standards are guidelines to develop and maintain the information security management system [4–8].

Information security must support business objectives by minimizing risks and developing trust. In addition, it is important to understand that information security requires continuous improvement in a cyclic approach [9]. Standards, such as British Standard 7799 (BS 7799) and ISO 27000 Family [10], are well-known guides in the information security business. Frameworks, such as Information Technology Infrastructure Library (ITIL) and Control Objectives for Information and related Technology (COBIT) [11] are also used in information technology governance in order to help organizations to reduce costs and increase productivity and, in some aspects, aiding security in terms of organization and methodologies [9].

However, ensuring compliance with standards does not guarantee security at all. To deal with information security, it is required to go beyond compliance or the best practices. Up until now and to the best of our knowledge, there is no known proven technology or framework for developing application and information systems without security issues, such as exploitable flaws, configuration errors or system misuse.

Discussions and references precede 1980, considering the information security triad (confidentiality, integrity and availability). The main discussion about confidentiality dates back to 1974, maybe earlier, with Bell and LaPadula [12]. Regarding integrity, the references quote [13] and his model that describes a set of access control rules designed to ensure data integrity. Regarding availability, some references are found in the National Institute of Standards and Technology (NIST) [14]. The exact origins of the 'CIA triad' expression appear to be unknown, but apparently, they date back to the NIST publications of the 1990s.

Regarding information security architectures, the zero trust model for cybersecurity [15] conveys a very simple message: Stop trusting packets as if they were people. The idea behind this model is that the concept of internal and external networks should be changed, because one should assume that all network traffic is untrustworthy. In practice, zero trust claims that one should protect internal data from insider abuse as one protects external data in public networks.

In [16], the authors discuss the general effect of risk management, which is limited to security activities regarding an information security architecture on information assets for an organization.

The work proposed in [17] describes an information security architecture with three levels: Domain, component and control. Their target is to align security and business and customer service and manage security, and to create traceability between them, proactively and predictively.

According to [18], it is necessary to develop and improve information security in both administrative and organizational levels. Then, a holistic view of information security based on a survey of the literature, along with a specific Swedish model, is proposed. According to the authors, they have extended the model based on the concepts of semiotic theory and on the perspective of an information system, which included technical, formal and informal sectors.

Behavioral information security shifts the way people are related to information security [19,20]. People have become the center when talking about dealing with information security, a standpoint with which we agree.

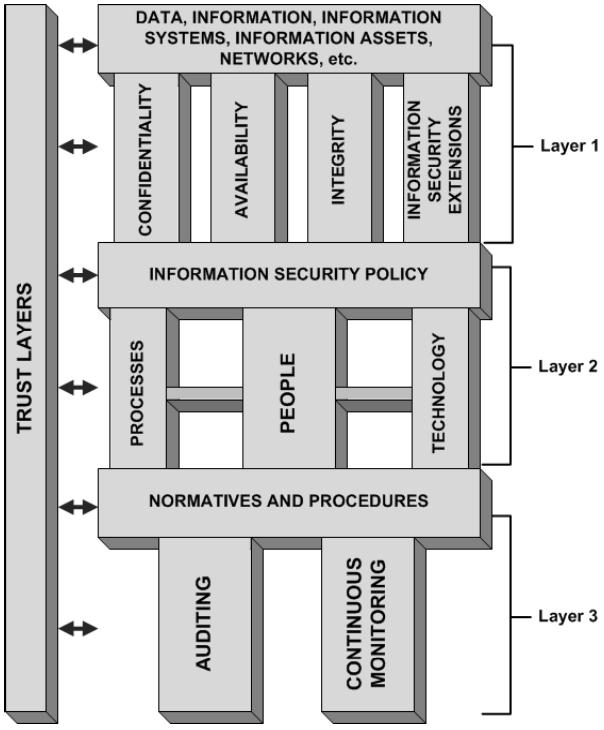
Much more has been discussed when it comes to information representation ability. In technology, if information can be retrieved, it is because it has somehow been digitally represented before [21]. Despite the amount of effort, it is still rather difficult to represent and retrieve the correct information for the correct user at the correct moment considering the amount of variables that may be included or related.

The web provides information from a variety of data sources that have great potential for knowledge discovery [22]. One of the long-standing problems is that search engines retrieve a plethora of data; however, most of it is useless, depending on what the user is interested in or searching for. Yet, when it comes to protecting such information during transferences, mechanisms, such as Secure Socket Layer [23] or Transport Layer Security [24], are used very often.

Considering these aspects, in order to protect information, it is very important to understand how it is digitally represented and treated. The users' view may be text, images or both formats. The Internet has redefined the way information is represented and how it is retrieved [25], and as a consequence, the way it is treated has significantly changed. Furthermore, information representation needs extra structural or semantic complements, which transform raw data into something intelligible to people.

Now that we have presented these issues, it is concluded that existing security architectures fail to manage risks, policies, people and assets efficiently. As a contribution, we propose a layered trust information security architecture (TISA) as a way to see how the components are connected regarding information security. In addition, this layered view is important to depict how all the security architecture elements interact with one another.

Figure 1. TISA: Layered trust information security architecture.



3. TISA: Trust Information Security Architecture

The way we assess security is based on a layered architecture with components connected in such a way that everything is part of a puzzle that must be well connected and understood, so information security can be seen as a whole. Figure 1 illustrates the proposed information security architecture and its layers.

The following topics describe each layer in a top-down description and its corresponding subtopics, in addition to a brief description of its importance to the puzzle.

3.1. Layer 1

This covers the basics to start thinking about information security inside an organization. If one does not understand what data, information, information assets, and so on, are to an organizational environment, there is no point in discussing information security, simply because one does not know what should be protected. It is important to point out that the "protect everything" approach is not effective, and it is very expensive. After that, there is the CIA triad and what we decided to call "information security extensions". The following items summarize our approach.

3.1.1. Data, Information, Information Systems, Information Assets, Networks, *etc*.

Generally, this is the layer in which data is important to organizations or in which individuals are found or mapped. Considering the importance of data for organizations nowadays, information can be retrieved from data and information systems. Using the correct tools and techniques, it is possible to create new knowledge from data, which, at first, made no sense at all. In the vast data world, crawling, retrieving, analyzing, discovering relations and finding hidden patterns are the areas in which efforts have been made in order to create knowledge from raw data.

When it comes to information assets, it is also very important that they are identified and labeled, and the relationship to information should be clearly understood by the organization and the ones responsible for its protection.

Networks connect every piece of data, information and information assets, so that anyone with granted access is able to browse them. When it comes to information or data in transit, it is important to remember that, in networks, bits are not treacherous by themselves: People who control them or how they do so (process, hardware or software) are who/what makes them dangerous or not. Therefore, marking network perimeters, having network policies, providing defense mechanisms *etc*., still are ways of managing what comes in and out of a network.

The proper use of such mechanisms is the key to understanding what happens in the network and information systems. These components of the puzzle create inputs for supporting decisions related to information security.

3.1.2. Confidentiality

Confidentiality is the information security property responsible for preventing unauthorized disclosure of information. In other words, it is a mechanism to give access to authorized individuals or systems in

the organization only. It applies principles, such as the "need-to-know", and in order to be effective, confidentiality must ensure that access to vital information is limited only to those who specifically need to have access to or use that particular information. This piece of the puzzle deals with the organization's ability to keep its data, information and knowledge protected from unauthorized disclosure.

3.1.3. Availability

Every piece of information has a specific value or use depending on a specific end. In order for information systems to serve their purpose, information must be available when necessary. To acquire such a property, all information systems, networks, databases, information assets, storage mechanisms, and so on, must be accessible by the ones who have granted access to manage them when required. Information is unavailable, not only when it is lost or destroyed, but also when its access is denied or delayed for those who are authorized to use it.

Therefore, to guarantee availability, communication channels used to access information, wherever it is, must be operating correctly and according to information security policies. This piece of the puzzle deals with the required organizational skills to guarantee that the ones who have the necessary permissions regarding its use, when necessary, can access information.

3.1.4. Integrity

Integrity is the ability to guarantee the accuracy and consistency of data and information during its entire life cycle. Considering such a property, data or information should not be unauthorizedly destroyed or modified, which would hinder its detection. Integrity has to guarantee that the information is accurate, reliable and, most importantly, that it has not been changed or tampered with by an unauthorized party all of the sudden. Integrity may include the ability to verify whether information content has been unauthorizedly altered, to determine the origin of any action in the system and to associate it with a specific entity. This piece of the puzzle deals with the way information is securely managed in organizations, with no loss of its basic properties.

3.1.5. Information Security Extensions

There is discussion among the information security experts regarding whether the information security triad is enough to stand as the basis of information security by itself; we believe it is not. However, it is also very difficult to extend all types of information security properties or attributes, thus ensuring that your information is secure for you to have access to the so-called extensions.

We picture information security extensions as a group of new attributes or properties to protect information and systems, but that are not limited to it. We summarizes some of them from this work's perspective.

- Authentication: This is the measure that verifies identities. In a communication process, the party must provide evidence that it is the person or entity to whom the credentials belong. At least one identification element is included in the authentication process. Usually, this process requires personal or individual knowledge, ownership information or personal information, despite the way

these are linked. They are related to something you know, something you have or possess and some information about yourself.

- Access control: This is what restricts access to a certain datum, information or resource. According to this principle, once access control is guaranteed, the entity should be able to extract, enter or use a particular piece of information or an information system. Generally, access control mechanisms begin with the identification and authentication processes.
- Non-repudiation: In short, a party within a communication process cannot deny having received specific information, nor can the other party deny having provided specific information. Generally, cryptographic systems can support non-repudiation efforts using digital signatures, but the discussion goes beyond the technological field, because one cannot guarantee that such a signature proves authenticity and integrity, thus preventing repudiation; for instance, cases of data theft. The opposite is so-called plausible deniability, in which one's culpability might be denied or mitigated or it is even possible to deny that one was responsible for it in the first place.
- Authenticity: This is a way to ensure that data, systems, communication processes or information are genuine. Authenticity is also responsible for validating that both parties involved in a communication process are who they claim to be.
- Privacy: This regards the right to control information about an individual and the right to limit access to that information. It is also related to domains in which individuals have the right to keep confidential information and data and to share them in private conversation [25]. It is also the right to protect your personal information and to prevent invasions of privacy. Privacy is preserved for one's good [26].
- Anonymity: This is simply a result of not having identifying characteristics disclosed or made available to the public, which would allow the identification of an entity. In a communication system, anonymity is the party's state of being anonymous and, yet, being able to respond to or interact with another party (without revealing its identity). It is also related to terms, such as untraceability and unlinkability [27].
- Authorization: This is the process of providing access to particular information or a system to a party based on their identity. After going through the authorization process, one is allowed to have access to some or all of the data in a specific environment or system. Authorization allows an entity to access and perform determined actions regarding data. In order to be effective, authorization should be based on the roles that an entity may have.

Accordingly, these properties are able to support themselves as elements that depend on many other complex characteristics, such as context, technologies that support them, usage objectives, and so on. In short, information security extensions are a bit of the puzzle that completes the information security basis, and in addition to that, some of the terms are sufficient by themselves.

3.2. Layer 2

According to the top-down description, this is the layer where the architecture will help to understand how, why and which technology may be used and who may use it in order to provide information security for higher levels. The following items summarize these.

3.2.1. Information Security Policy

The information security policy is a high-level document that outlines specific requirements or rules that must be met regarding information security. Generally, this policy is very specific and covers only one organization. The information security policy also links security management to security issues and security controls. Once the information security policy is straightforwardly attached to the organization, it is important that the policy is formulated taking the organization's features into consideration.

The information security policy should explain the fact that all users need information security within the organization and should complement business objectives. Thus, the policy should be aligned with the organization's strategic information plan [28].

The information security policy is the piece of the puzzle that guides people during the creation of the processes and the definition of technological components in order to help to protect information. Daily practices show that without information security policies, things are done based on individual efforts that, usually, are non-effective.

3.2.2. Processes

Inside information security, processes are formal mechanisms to identify, measure, manage and control risks related to information or its value to the organization. Processes are very important when information security is in evidence, thus it should not be seen as a black box or something that is meaningless to the organization. To be more specific, processes include formal and informal mechanisms (large and small, simple and complex) and provide a fundamental link to all dynamic interconnections.

Processes derive from strategies and implement the operational side of what should be done within the organization. To be useful and provide advantages to the enterprise, processes must be directly linked to business requirements and be aligned with information security policy in the puzzle. They also should consider emergency situations and be adaptable to changes in requirements. It is important that information security processes are well documented and communicated to human resources, which should know about them. It is fundamental that processes should be reviewed periodically to ensure efficiency and effectiveness [28].

3.2.3. People

"People" is the number one subtopic of the puzzle and represents human resources. In general, people develop and implement each part of an information security policy, create and maintain processes, information assets, define which technology should be used, design networks, *etc*. Security issues affect people and their relationships, values and behavior. When working with information security, it is important to address points, such as strategies related to hiring, access, responsibilities, training, dismissal, damage and whatever is considered important to be addressed, to help to maintain the organization's information security strategy.

When dealing with people, it is very important to understand that what sounds obvious to security experts definitely does not sound obvious to someone without the same experience level. People are the ones whose actions within the organization influence the information security triad and security extensions related to data, information, information systems, information assets, network usage,

resources and whatever is valuable to the organization. In short, individuals' actions and motivations have direct positive or negative influence on information security. All of this is straightforwardly related to behavioral information security [19,20].

3.2.4. Technology

This the piece of the puzzle that is the set of all informatics systems, applications, tools, infrastructure and defense mechanisms that the organization applies to achieve its goals and to help information security. Technological elements may frequently change and update, become obsolete very fast or be the core of an organization's infrastructure. Technology may also solve security threats and risks. Users and the organization's environment also have a strong impact, once technology can be perceived as a way to avoid security controls being transpassed [28].

It is very important to remember that technology, by itself, does nothing. It should be seen as part of a complex system that has specific needs to protect what is valuable within the organization and should work along with people and processes in a cyclic pattern, all these guided by information security policies.

3.3. Layer 3

This represents the piece of the puzzle that deals with daily activities and how they should be done. In addition to that, it is related to what actions should be taken in case a specific problem arises. Bear in mind that complex systems and their security practices are guides to keep information secure; but norms, procedures, monitoring and auditing will give systems administrators the tools to help them keep information, information assets, networks, systems, *etc*., more protected.

3.3.1. Normatives and Procedures

Basically, a normative is related to how things should be and how they should be rated. It is also related to how to classify things as good or bad or which actions are right or wrong. Normatives are fundamental for prioritizing goals, organizing and planning actions in order to define how things should be done. Considering the user's perspective, it is much more common than it may seem, but users generally try to follow social expectations rather than following security procedures [29]. This is not desirable regarding security perspectives, because such conflicts lead to security policy inconsistency.

In general, normatives can be implicit or explicit, while security policies are explicitly mandatory normatives. Procedures are step-by-step definitions of how things must be done considering specific contexts. It is also a guide to what one should do if a particular condition is met.

A procedure is way to provide minimum safeguards (administrative, technical, physical or all of these) that are employed to protect sensitive information. It is expected that procedures are technical and intended for information custodians, systems administrators and information technology personnel within the organization.

3.3.2. Auditing

Consider that auditing information security is a process that takes measures in terms of qualitative and quantitative assessing of the current state of what is being audited regarding specific criteria of information security.

Auditing is the key to discover risks, technical flaws, policies, procedures and normative problems. Bearing in mind that auditing is an everlasting process, when auditing information security, one should be prepared to cover topics from the physical security of data centers to logical security, including network perimeters, system configuration and information systems. Each one has its peculiarity, thus different methods for auditing should be taken into consideration and should also be applied to guarantee the auditing results.

Always assume that information security auditing is for information security professionals and the entire auditing process should be part of an overall plan. Auditing will provide at least an independent review of the adequacy and effectiveness of the internal controls regarding business process, people behavior, infrastructure current state, policies, normatives and procedure compliance.

Auditing is also an operating measure to verify the state of what is being audited in a specific period of time, so it can verify previous behavior, but will not guide the organization for all future possibilities.

3.3.3. Continuous Monitoring

Continuous monitoring keeps track of ongoing knowledge of information security, vulnerabilities, threats [30] and, thus, their associated risks. It is the key to support decisions regarding risk management. It also guides decisions regarding protective and proactive measures.

This piece of the puzzle is important, because compliance with international normatives and standards does not guarantee that the organization's security objectives will be achieved. In practice, using normatives and standards will help to achieve a higher level of maturity regarding information security.

Monitoring systems begin with the definition of what, how, when and why information assets or any part of the architecture should be monitored. It relies on technology, processes, procedures, operating environments and people. It is deeply connected to the understanding of organizational risk tolerance. It also helps to set priorities and to consistently manage risk inside the organization. It should be standardized, so that one is able to give visibility and identify security status at all monitored information assets.

The task of monitoring can be perceived as an active security component [31]. It can sense the infrastructure's current state. It also considers a wide variety of data and information in order to react according to specific defined policies as a way to help protect the information within the organization. Continuous monitoring should be based on measures, such as protection, sense, adjustment, collection, alarms and others related to information security.

3.4. Trust Layer

When it comes to security, trust is zero or one. You trust your information systems, network, *etc*., or you do not; "maybe" should be avoided. Usually, trust can be acquired by empiric observation, by formal

proof of the systems and the mechanism involved and other techniques [32]. Once all expectations are fulfilled, one may establish trust. The problem is that trust by its own means is an expectation. It is a probability that things will work and keep working as they are supposed to.

Considering a security perspective, for something to be trusted, it must be clearly identified and operate exactly as planned and expected. It also must not do anything it was not supposed to do and must be able to operate nonstop. If a trust approach is taken, and it is acknowledged that a system may be or has been compromised, it is enough to make it suspicious or untrustworthy.

Trust and security are closely related [32]. If security objectives are considered, it is clear that trust is connected to security, because information security depends on people and security extensions, such as authentication, authorization, access control, non-repudiation, *etc*. However, in practice, information security approaches should be "trust, but verify." [15]. Additionally, verify always, even if apparently there is nothing suspicious.

In general, trust is the piece of the puzzle in which there is enough knowledge about information, systems, technology and other components to help one to make assertions, such as "fully secured" or "information is secure because a particular condition was met". Such cases are very common nowadays, because most people do not build information systems. They use them and trust them in terms of functionality and security. However, the scope of trust goes beyond that. It is more related to the ability that one may have to check all of the particularities of the software or hardware he uses and, also, being able to careful choose what he trusts or not. Additionally, he should define what would be the security acceptance regarding a particular hardware, software or information system.

Having such particularities in mind, the layered approach herein presented allows one to see that trust is connected as part of the security architecture and may be addressed when needed in all layers and in all components that are part of the architecture.

The IEEE Cybersecurity Initiative published a list of what they felt were the top security design flaws [33], where trust is considered as an important subject when security is considered. They state, for example, that data sent to an information system by untrusted clients or channels should be assumed to be compromised until proven otherwise. In such cases, if the integrity cannot be verified, the data inherently are not trustable.

In short, the main message is to make sure all data received from an untrusted client or system are properly validated before processing, because one may be exposed unnecessarily to vulnerabilities by trusting components that did not earn that trust [33].

3.5. Measuring Trust in a Distributed Environment

In [34], the authors present a model able to measure trust regarding groups in distributed environments. As a particular result, using the same group trust model as in [34], we extended the measure capabilities with more nodes in a simulation.

In these results, the test bed is composed of five different groups with 200 nodes in each group, having a total amount of 1000 nodes. In all of the result graphs, the x-axis represents the number of rounds that the nodes perform for their particular job, having a total of 20 rounds used to calculate group trust. The

y-axis represents the amount of trust calculated in each round using the model proposed in [34]. We simulated three different scenarios.

In the first scenario, we simulate the ideal environment, where all nodes behave accordingly and perform their job as expected, so that they fulfill the other nodes' outlooks. In this case, this represents a scenario where there are no malicious nodes inside the network and all groups are considered trustworthy. Figure 2 illustrates this perspective.

Figure 2. Ideal scenario where all groups are trustworthy.

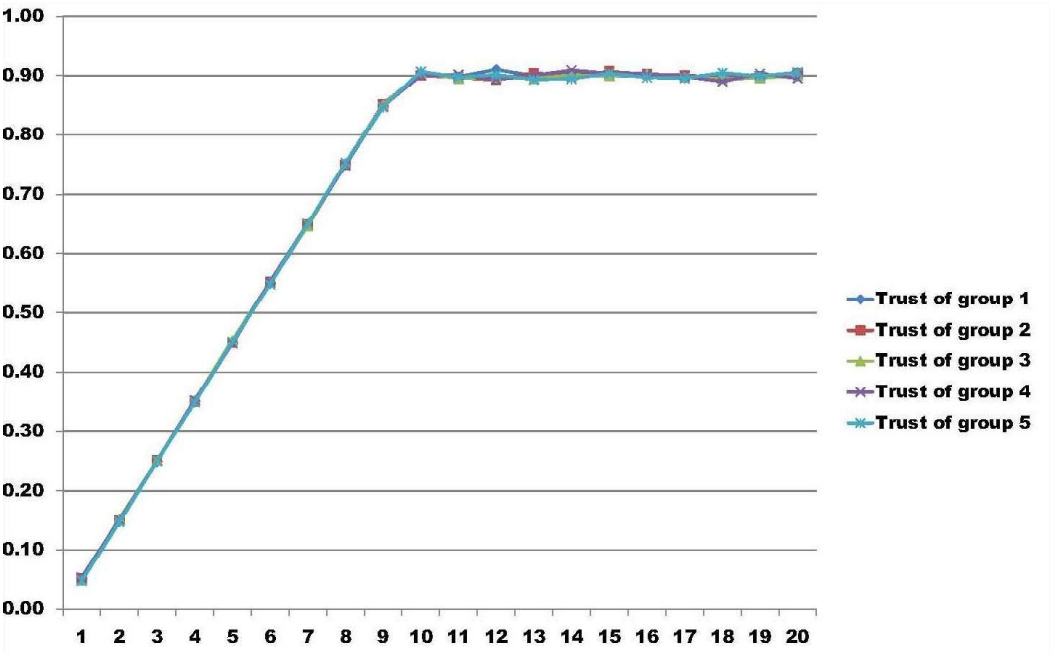


Figure 3. Worst scenario where all nodes in all groups behave randomly.

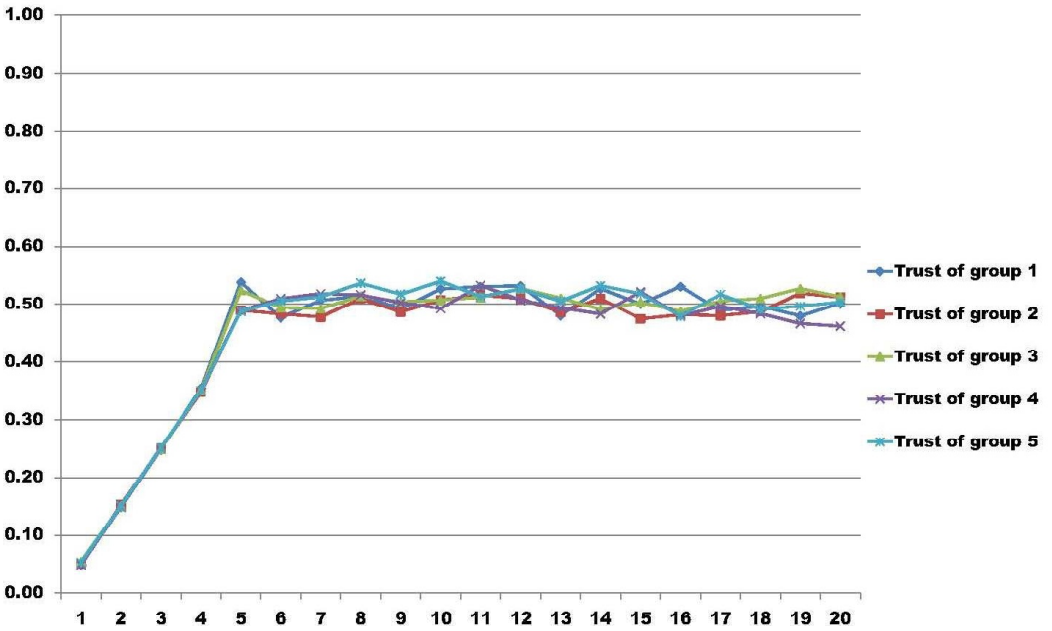
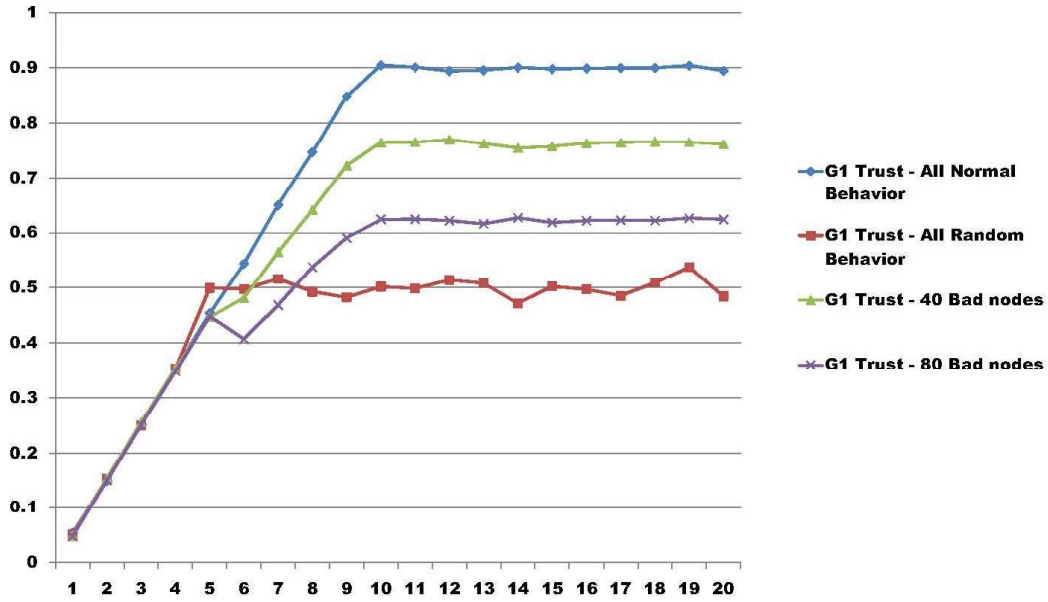


Figure 4. Group trust view for the first group.



In the second scenario, we have all nodes moving in a random way. This scenario, illustrated in Figure 3, represents the worst environment possible regarding group trust. In a condition like this, it is not possible to say if the node is bad or good, because it changes its behavior unexpectedly. In such cases, one particular node may be trustworthy to one entity, because it may fulfill one's expectation in a particular condition, but next, it may completely change its state.

The last scenario, illustrated in Figure 4, summarizes the view of Group 1 in four different approaches. It compares Group 1 trust with all nodes behaving accordingly and all nodes behaving randomly. Then, it shows a particular case where 40 nodes (20% of the members of Group 1) change their behavior after Round 5. After, it shows 80 nodes (40%) of the members with changes in their behavior. In both cases, with 40 and 80 bad nodes, it is possible to see that when a coalition of nodes are made, the group trust model is able to address the change in the trust of the group.

4. Comparing Information Security Architectures

It is very hard and also complex to compare information security architecture in general. There is no common approach or benchmark able to measure every peculiarity of each known architecture. Additionally, doing so, this still may be incomplete for addressing every aspect related to information security.

The work in [35] addresses the problem of how hard it is to compare information security models. In general, it recommends to first perform a high-level comparison based on the criteria definition. Then, it addresses that it is also important to create an appropriate selection of candidates and then perform a more time-consuming extensive evaluation.

The authors in [36] review various information security standards and compare major information security standards ISO 27001, BS 7799, Payment Card Industry Data Security Standard (PCIDSS), ITIL and COBIT in terms of information security policy, communications and operations management, access control, information system acquisition, development and maintenance, organization of information security, asset management, information security incident management, business continuity

management, human resources security, physical environment security and compliance. The authors conclude that each standard plays its own role and position regarding information security management systems.

The work in [37] also reviews some information security standards and architectures in terms of security policy theory, risk management theory, control and auditing theory, management system theory and contingency theory.

Having the assumptions of previous works in mind and taking a trust approach, we compare TISA with two major widely known information security guides. Table 1 summarizes the comparison approach.

Table 1. Comparison criteria.

Architecture\Criteria	**ISO 27001**	**BS 7999**	**TISA**
Trust as an important role	No	No	Yes
Address security extensions	Yes	Yes	Yes
People as an important role in security	Yes	Yes	Yes
Continuous monitoring activities	Yes	Yes	Yes
Auditing activities	Yes	Yes	Yes
Address security compliance	Yes	Yes	No

As seen, basically, TISA considers trust as an important role regarding information security, while ISO 27001 and BS 7799 have little or no consideration of trust. Furthermore, TISA considers that compliance guarantees no security. Compliance in this case helps increase the level of maturity regarding information security.

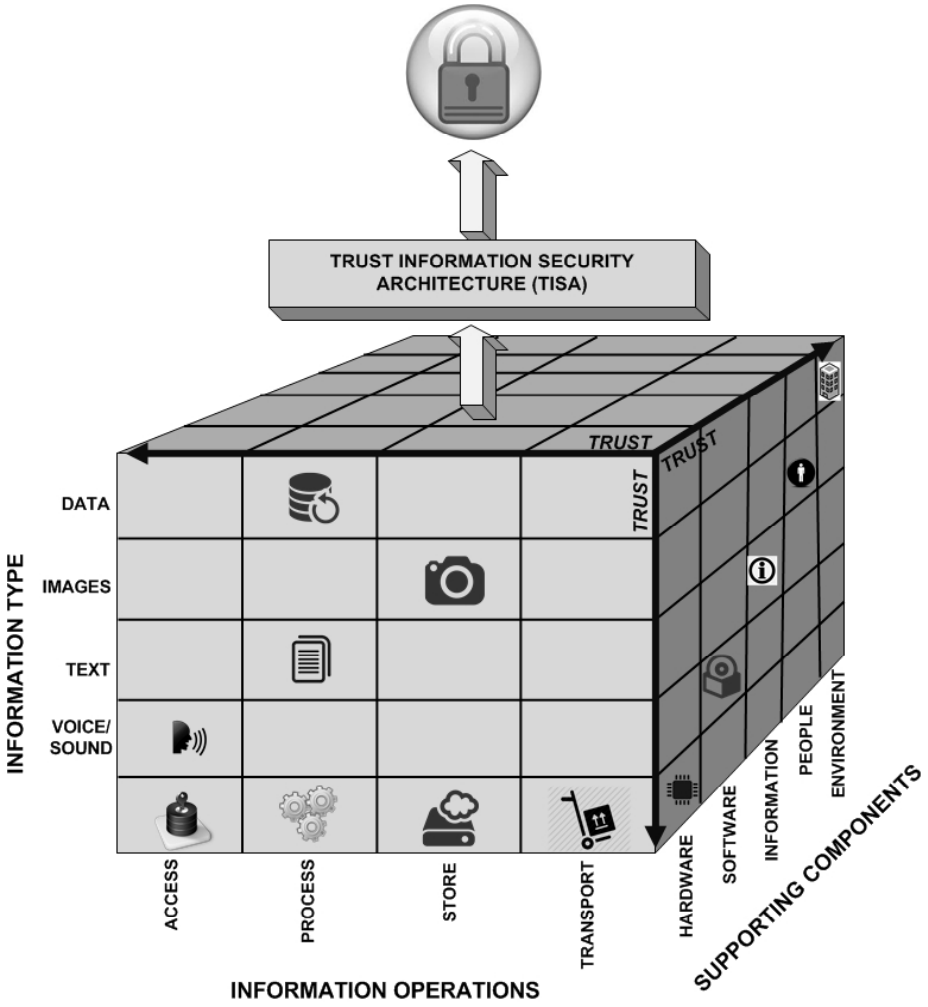
5. Information Representation and Treatment and Their Relation to TISA

When it comes to security, the type of information, that operations that can be performed regarding it and the supporting components must be considered if you want to protect them. In fact, you cannot protect information if you do not understand where it is stored, how it is represented or who or what manages it. In other words, how information is digitally treated is key to trying to protect it.

Figure 5 depicts how information is treated regarding its basic representation, the basic operations to process it and where these activities may digitally happen from a cyber perspective. If the whole complexity of Figure 5 is considered, it sounds very difficult to guarantee or to achieve information security. From our perspective, a trust layer is used as a means to explain when technology or a regular approach by itself is not enough to guarantee security during the entire process. For instance, from the users' perspective, things should work just as when such things are supported by an information security architecture, as described in previous sections.

Considering the digital representation of information, the types of information can be images, text, voice or sound and videos (a combination of images and sound). When it has no particular characteristic, it can be perceived as data (which may be a set of digital representations, such as database files or proprietary data types).

Figure 5. Types of information, operations and components.



Once the information has its representation regarding a particular information type or a combination of them, it may undergo specific operations in the digital environment. Operations, such as access data, process text, store voice records or transport any of those from one place to another using any kind of media (disks, cell phones, tablets, networks, *etc*.), are very common in modern organizations. Any combination of such information operations is also possible; for example, one particular user may wish to transform a text file into a voice file, or *vice versa*, or a particular user wishes to describe an image giving it more sense then the previous one. All of these processes are possible and very common in the digital environment, and there are a lot of possibilities when it comes to the combination of information type and information operations. Thus, to protect it in all existing scenarios is rather difficult.

The supporting components are where or how the information is used. For example, a video is stored in hardware and uses proprietary software to play. This is also where people directly act regarding information. For example, if a user copies a file from the database to a thumb drive disk through the network, different supporting mechanisms are applied so that the user is able to achieve his or her goal.

The entire cube depicts how information is digitally represented and treated. Therefore, it should be considered that information security should be guaranteed inside environments where the following activities take place: copying files, printing images, writing data to database systems, using distributed

network environments, network connections of all types are supported, users making common mistakes, such following unknown hyperlinks, and so on.

According to a very simple approach, it is a complex matrix of at least three dimensions to keep one piece of information secure inside such environments. That happens because one should consider the type of information, which operations it may undergo and what information support is used, all that just as a small example of to keeping information secure.

In our perspective, we believe it to be very difficult to achieve information security regarding the cube perspective depicted in Figure 5 without using an information security architecture as proposed in Section 3. According to [1,2,15], using technology or being compliant with standards is not enough to keep information protected. More needs to be accomplished, and in such cases, trust should be used even without knowing it in first place.

6. Conclusion and Future Work

Information security is not an end by itself; it is a means to achieve an end [5]. Information security is also a constantly developing subject, due to the ever-increasing scale and complexity of security threats in recent days.

Nowadays, the information security research field is becoming more and more important, significantly because the world is highly interconnected, with the use of networks to carry critical and sensitive information. Considering a cyberspace perspective, information may be in use by an entity, may be stored inside media or may be in transit through some communication channel. Additionally, if you consider your information important and care about information security, you should protect it in whatever state it is.

All things considered, at the present time, one cannot protect his/her information without understanding its whole life cycle. Bearing in mind a deeper perspective, if information security is a goal, one should be able to represent it, process it and use it in a cyberspace environment where people, technology, information assets, hardware, software, and so on, are amply used and connected. Consequently, security measures should be taken to guarantee protection, and this is where the information security architecture comes in, because the "protect one piece of information" scenario by itself has been proven to be ineffective [1,2].

In this paper, we introduced a layered trust information security architecture (TISA) and its connected elements, which are useful to manage risks at different levels regarding information in an organization. In the layered architecture, every sub-item is a piece of the puzzle, so information security can be analyzed as a whole.

Government, organizations and enterprises consider that information security management needs a systematic approach to consistently address security in every layer, reducing unmanaged risks and improving operational security efficiency. From such a perspective, the proposed layered information security architecture can be used as a guide to helps achieve better results regarding information protection.

The proposed architecture provides several opportunities for further security research. Regarding future works, first of all, we intend to extend the trust functionalities and better explain the relations

between the sections and all layers of the architecture. Secondly, we intend to carry out more research to address all of the details regarding trust in the architecture. Thirdly, we intend to detail how the pieces of the puzzle, such as people, technology and processes, should be connected and guided using information security polices, as in [28]. Finally, the information representation and treatment cube can be improved regarding an information security perspective that would connect it to an information security life cycle.

Acknowledgments

This work was supported by "Programa de Financiación de la Universidad Complutense de Madrid - Banco Santander para Grupos de Investigación UCM (Referencia: GR3/14)". Robson and Fábio acknowledge the Laboratory for Decision Technologies at the University of Brasilia (LATITUDE/UnB) for its support of this work. Fábio would like to thank PNPD/CAPES (Programa Nacional de Pós-Doutorado/CAPES) in Brazil for the support.

Author Contributions

R. de Oliveira Albuquerque, L. J. García Villalba and A. L. Sandoval Orozco are the authors who mainly contributed to this research, performing experiments, analysis of the data and writing the manuscript. F. Buiati and T.-H. Kim analyzed the data and interpreted the results. All authors read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Greenwald, G.; MacAskill, E.; Poitras, L. Edward Snowden: The whistleblower behind the NSA surveillance revelations. *The Guardian*, 10 June 2013; Volume 9.
2. Richelson, J.T. The Snowden Affair. *Washington Post*, 4 September 2013.
3. Release, G.P. *Gartner Says Cloud-Based Security Services Market to Reach 2.1 Billion in 2013*; Gartner Inc.: Stamford, CT, USA, 2013.
4. Blakley, B.; McDermott, E.; Geer, D. Information security is information risk management. In Proceedings of the 2001 Workshop on New Security Paradigms, New York, NY, USA, 11–13 September 2001; pp. 97–104.
5. Peltier, T.R. *Information Security Fundamentals*; CRC Press: London, UK, 2013.
6. Disterer, G. ISO/IEC 27000, 27001 and 27002 for Information Security Management. *J. Inf. Secur.* **2013**, *4*, 92–100.
7. Stamp, M. *Information Security: Principles and Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
8. Posthumus, S.; von Solms, R. A framework for the governance of information security. *Comput. Secur.* **2004**, *23*, 638–646.

9. Secure Your Information: Information Security Principles for Enterprise Architecture. Available online: http://www.tisn.gov.au/Documents/Secure_Your+Information+-+Information+Security+Principles+for+Enterprise+Architecture+-+Report.pdf (accessed on 27 November 2014).
10. Whitman, M.; Mattord, H. *Management of Information Security*; Cengage Learning: Boston, MA, USA, 2013.
11. Parvizi, R.; Oghbaei, F.; Khayami, S.R. Using COBIT and ITIL frameworks to establish the alignment of business and IT organizations as one of the critical success factors in ERP implementation. In Proceedings of the 2013 5th Conference on Information and Knowledge Technology (IKT), Shiraz, Iran, 28–30 May 2013; pp. 274–278.
12. Bell, D.E.; LaPadula, L.J. *Secure Computer Systems: Mathematical Foundations*; Technical Report; DTIC Document: Springfield, MA, USA, 1973.
13. Biba, K.J. *Integrity Considerations for Secure Computer Systems*; Technical Report; DTIC Document: Bedford, MA, USA, 1977.
14. Burrows, J.H. *Guideline for Computer Security Certification and Accreditation*; PN: Springfield, VA, USA, 1983.
15. Contreras, J.L. *Developing a Framework to Improve Critical Infrastructure Cybersecurity*; University of Utah: Salt Lake City, UT, USA, 2013.
16. Jeong, G.H.; Yi, D.W.; Jeong, S.R. The Effect of Composition and Security Activities for Information Security Architecture on Information Asset Protection and Organizational Performance. *KIPS Trans. Part D* **2010**, *17*, 223–232.
17. Tang, C.L. Establish a Dynamic Business Driven Integrative Information Security Architecture. *Appl. Mech. Mater.* **2014**, *513*, 1309–1315.
18. Åhlfeldt, R.M.; Spagnoletti, P.; Sindre, G. Improving the Information Security Model by Using TFI. In *New Approaches for Security, Privacy and Trust in Complex Environments*; Springer: Sandton, South Africa, 2007; pp. 73–84.
19. Crossler, R.E.; Johnston, A.C.; Lowry, P.B.; Hu, Q.; Warkentin, M.; Baskerville, R. Future directions for behavioral information security research. *Comput. Secur.* **2013**, *32*, 90–101.
20. Aurigemma, S.; Panko, R. A composite framework for behavioral compliance with information security policies. In Proceedings of the 2012 45th Hawaii International Conference on System Science (HICSS), Maui, HI, USA, 4–7 January 2012; pp. 3248–3257.
21. Chu, H. *Information Representation and Retrieval in the Digital Age*; Information Today, Inc.: Medford, NJ, USA, 2003.
22. Nayak, R.; Senellart, P.; Suchanek, F.M.; Varde, A.S. Discovering interesting information with advances in web technology. *ACM SIGKDD Explor. Newsl.* **2013**, *14*, 63–81.
23. Freier, A.; Karlton, P.; Kocher, P. *The Secure Sockets layer (SSL) Protocol Version 3.0*; Internet Engineering Task Force (IETF): Dallas, TX, USA, 2011.
24. Dierks, T. *The Transport Layer Security (TLS) Protocol Version 1.2*; Internet Engineering Task Force (IETF): Dallas, TX, USA, 2008.
25. Diffie, W.; Landau, S.E. *Privacy on the Line: The Politics of Wiretapping and Encryption*; MIT Press: Cambridge, MA, USA, 2007.
26. Anderson, R. *Security Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2008.

27. Hughes, D.; Shmatikov, V. Information hiding, anonymity and privacy: A modular approach. *J. Comput. Secur.* **2004**, *12*, 3–36.
28. An Introduction to the Business Model for Information Security. Available online: http://www.isaca.org/Knowledge-Center/BMIS/Documents/IntrotoBMIS.pdf (accessed on 27 November 2014).
29. Pieters, W.; Coles-Kemp, L. Reducing normative conflicts in information security. In Proceedings of the 2011 Workshop on New Security Paradigms Workshop, Marin County, CA, USA, 12–15 September 2011; pp. 11–24.
30. Dempsey, K.; Chawla, N.S.; Johnson, A.; Johnston, R.; Jones, A.C.; Orebaugh, A.; Scholl, M.; Stine, K. *Information Security Continuous Monitoring (ISCM) for Federal Systems and Organisations*; NIST Special Publication: Gaithersburg, MD, USA, 2011; pp. 800–137.
31. Hand, R.; Ton, M.; Keller, E. Active security. In Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks, College Park, MD, USA, 21–22 November 2013; p. 17.
32. Lamsal, P. *Understanding Trust and Security*; Department of Computer Science, University of Helsinki: Helsinki, Finland, 2001.
33. Avoiding the Top 10 Security Flaws, 2014. Available online: http://cybersecurity.ieee.org/center-for-secure-design/avoiding-the-top-10-security-flaws.html (accessed on 27 November 2014).
34. De Oliveira Albuquerque, R.; García-Villalba, L.J.; Kim, T.H. GTrust: Group Extension for Trust Models in Distributed Systems. *IJDSN* **2014**, *2014*, 872842.
35. Van Os, R. Comparing Security Architectures: Defining and Testing a Model for Evaluating and Categorising Security Architecture Frameworks. Master's Thesis, Lulea University of Technology, Lulea, Sweden, 2014.
36. Susanto, H.; Almunawar, M.N.; Tuan, Y.C. Information security management system standards: A comparative study of the big five. *Int. J. Electr. Comput. Sci. IJECS-IJENS* **2011**, *11*, 23–29.
37. Hong, K.S.; Chi, Y.P.; Chao, L.R.; Tang, J.H. An integrated system theory of information security management. *Inf. Manag. Comput. Secur.* **2003**, *11*, 243–248.

The Journal of Supercomputing manuscript No.
(will be inserted by the editor)

Relations in Cyber Security and Information Security and Trust

Robson de Oliveira Albuquerque · Luis Javier García Villalba · Ana Lucila Sandoval Orozco · Rafael Timóteo de Sousa Júnior · Tai-Hoon Kim*

Received: date / Accepted: date

Abstract This paper provides a survey about three security connected fields: trust, cyber security and information security. In cyber security there are hazards such as exploits and Advanced Persistent Threats (APTs), which are discussed in particular sections of this paper. Information may be considered the most important asset of any modern organization, thus dealing with information security is very important because from processed and organized information comes knowledge, and therefore it needs protection. Information security is also a risk management activity; consequently it is tied to trust,

R. de Oliveira Albuquerque
Group of Analysis, Security and Systems (GASS)
Department of Software Engineering and Artificial Intelligence (DISIA)
Faculty of Information Technology and Computer Science, Office 431
Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain.
E-mail: robson@fdi.ucm.es
Electrical Engineering Department, University of Brasilia
Campus Universitrio Darcy Ribeiro, Asa Norte, 70910-900 Brasilia, DF, Brazil.
E-mail: robson@redes.unb.br

L. J. García Villalba and A. L. Sandoval Orozco
Group of Analysis, Security and Systems (GASS)
Department of Software Engineering and Artificial Intelligence (DISIA)
Faculty of Information Technology and Computer Science, Office 431
Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain.
E-mail: {javiergv, asandoval}@fdi.ucm.es

Rafael Timóteo de Sousa Júnior
Electrical Engineering Department, University of Brasilia
Campus Universitrio Darcy Ribeiro, Asa Norte, 70910-900 Brasilia, DF, Brazil.
E-mail: desousa@redes.unb.br

Tai-Hoon Kim
Dept. Convergence Security, Sungshin W. University,
Dongseon-dong-3-ga, Seongbuk-gu, Seoul, Korea.
E-mail: taihoonn@daum.net

privacy, anonymity, confidentiality, integrity, availability and other characteristics. When emphasis is given to trust, one realizes that it has different lines of understanding because of its subjective evaluation. Trust and reputation are both subject to social evaluation, suffer influence of time, and are context and behavior dependent. Considering points as trust and cyber security, both fields have enormous impact in modern society. That is why almost everything in our day-to-day activities deals with some information and communication technology that are connected to the Internet. In general, this paper provides detailed information about trust, cyber security and information security. It discusses aspects regarding APTs and exploits and how they are being used in cyber space. This paper also provides a view of information treatment related to trust and information security.

Keywords APT · cyber security · exploit · information security · trust

1 Introduction

Protecting cyberspace is vital to maintain trust on the Internet and most of the information systems the way it is known. Most of that is because cyberspace is based on the uninterrupted availability of the Internet and because information systems are expected to work efficiently. Cyberspace is considered as critical infrastructure to many countries these days. Thus, dealing with threats in this environment is important and affects areas in society such as economics, health and energy among others. However when one realizes things such as exploits, advanced persistent threats, cyber campaigns, etc. that are part of the day-to-day activities on the Internet, one may start doubting cyber security and wonder if it is really as reliable as previously thought. Cyber security is inserted in an environment where daily threats are real and they have destructive potential. From this perspective cyber security is connected to management, policies, physical infrastructures, virtual environments and collaboration among different parties.

Cyber security nowadays is a fundamental challenge to any country or organization. It has to deal with threats that are mostly unknown to ordinary people. It is connected with strategic interests and information security. It also has to consider communication technology as Internet, cellular networks, satellite communications, and critical infrastructure systems and so on. In order to manage such a huge field it is important to have security guidance, policies, collaborative work and to continually review processes that are subject to such activities.

In a wider perspective, cyberspace should be protected from abuse, misuse and malicious activity. Also governments play a major role in guaranteeing online rights in such environments. However, at the same time, the Internet should maintain reliability and interoperability where most of it is controlled by private sectors of society.

When the attention comes to trust in the cyber security, as noted by Geer [1], trust will simply end as surveillance activities takes place and keep growing

in scales that only Government actors are able to tell. Still, after Snowdens revelations [2], much work has to be done and one cannot deal with information in cyberspace naively.

According to Menn [3], the cyber security industry is a US$ 71 billion market in 2015. It is fragmented along geopolitical lines because security companies chase government contracts, but they also share information with security agencies. If only the technical aspects regarding information security are considered, the power behind security agencies and information security systems was previously only understood by experts. Now, there is a clear shift in the information security field. There is technology being developed and deployed rapidly. There are new companies concentrated in protecting user data from surveillance. There are frameworks being revised and new ones being developed with intentions to guarantee that user data is safe and cannot be understand by third parties.

Regarding compliance standards, it is important to note that being information security compliant has been proven ineffective. It means that being in compliance with a particular security standard, does not necessarily mean your information is safe. Also, it is important to mention that the perception of how data analysis is being held by big data industry is changing because of worries in security information area. It allows generations of knowledge from meaningless bulk data, changing the way decisions are made.

The world has been dealing with cyber security strategy for a long time. ENISA [4] has information that some countries are considering cyber strategy planning since the year 2000. Thus it shows that the cyber security field is no place for isolated actions. It requires strategic planning and specific goals. It considers that the study of technology is mandatory and it is in constant evolution. In such cases, past plans may not be appropriate anymore and should be constantly revised. The cyber security field is constantly moving, be it by the development of new technologies or by way of protecting them or simply by way of deceiving them. Anyway, the cyber security research field needs focus, analysis, constant timepiece, advanced training and skilled people. This is all part of a strategic view.

As an example of how the cyber security is being held with deep interest of nations, news [5] show that the Central Intelligence Agency is planning to change its structure to put emphasis on cyber security and cyber espionage. It is no secret that countries such as China, Israel, France, Germany, Spain, England, Russia, North and South Korea, etc. also pay close attention to cyber security. Another important point to mention is that much of this attention has relevance not only in protecting themselves, but also dealing with offensive measures in cyber space. Cases such as BelgaTelecom, Sony and American health companies are examples of that. Offensive security is more about using hackers to pentest boundaries and defense perimeters to test your own systems. It is also about using skilled people to gain control over your adversaries systems. Its a double edge sword.

Trust has a complex relationship in information security and cyber security. To this day, one of the main problems of trust has been how to build

trustworthy systems from untrustworthy components. From the perspective of this review, the absence of protocols to exchange trust and reputation information is a problem that needs attention to help to build trust in network systems, which would lead to the improvement of computational trust.

In summary, the main contributions of this paper are the following: a) it presents a schema where all these areas are connected to information assurance; b) it discusses current cyber security, trust and information security aspects; c) it presents an information security architecture and connected elements considering information treatment in cyberspace; d) it evaluates trust and its relations to information security; e) it discusses and presents some flow schemas about APTs and exploits being used).

This paper is organized into the following sections. Section 2 reviews some related works in this field. Section 3 presents discussions, definitions and reviews about information assurance and cyber security. Section 4 evaluates information security and trust. Section 5 reviews definitions and information about exploits and APTs. Section 6 concludes this paper.

2 Related Work

According to Shah and Mehtre [6], cyber security is a bigger problem now than it used to be in the past because of the growth of connectivity through the Internet, the extension of information systems and its complexity. In addition to this the importance that is given to critical infrastructure systems and the impact that it may have if such systems are misused or abused by unauthorized parties.

If one just considers that hackers will keep hacking and breaches in such systems will keep coming up, the analysis is quite simple. However things are far beyond that. We face real threats in systems that deal with peoples lives in such manners that are beyond control or measure by simple analysis. There is no doubt that efficiently dealing with threats and attacks to critical systems can prevent bigger damages to society. Still, things are happening in a cyclic view. Users keeps being targeted by phishing attacks and falling on them, not properly locking down sensitive information internally, allowing their users too much access without control measures, unnecessary network services up, and so on. All these things make it easy for targeted attacks to take place with high rates of success. Indeed, taking required measures to safeguard information from possible cyberattacks might not be enough because no matter how much effort you spend in security measures, it may not avoid data theft, systems exploitation, and other offensive measures. It has to be considered that it also depends on the attackers capabilities and resources.

The point is not just about investment either. It is about dealing with security from different points of view, it is about people, it is about technology, it is about continuous monitoring and so on. There must be a balance between perimeter protection, internal controls, auditing and constantly monitoring the technological infrastructure. Most of the time, attackers are already inside and

there isnt much that can be done on the perimeter to defend yourself. Without humans the most sophisticated technology applied to information security and cyber security may render useless.

If critical infrastructure is on notice, much of Supervisory Control And Data Acquisition (SCADA) systems relies on proprietary networks and hardware [7] and thus has been considered immune to cyberattacks for a long time. As researches keep going, this affirmation is no longer supported and it is absolutely unthinkable to avoid taking cyber security measures to protect such systems. SCADA systems are highly customized so it is unlikely that there are two exactly the same in production at different sites. So, what is done to protect one system may be completely different to protect another. It also leads to the point that hacking such systems required specific knowledge and a lot of investment, i.e duplicating the infrastructure, having specific controllers available for testing, and so on.

Safe and reliable operation of SCADA in critical infrastructure systems is a major concern as stated in [8]. Studies of the work show that schemes are based on high measurement redundancy fail in the presence of intelligent and skilled attackers, where they can, for example, send false information to the control centre. Their research shows that the more accurate model the attacker has access to, the larger deception attack they can perform undetected. The developed tools can be used to further strengthen and protect the critical state-estimation component in SCADA systems [8].

In [9] there is a conclusion that states that hacking activities, active defense measures and everything in between have lawful and unlawful impact and all associated risks with deceptive practices, misattribution, and escalation. Tools, technologies, partnerships, and information sharing between corporations, governments, vendors, and trade associations are promising, effective and improving, all that considering a cyber security perspective.

The work discussed in [10] shows that virtualization is a major area when cyber infrastructure takes place. With virtualized systems it is possible to raise and automate the availability in an information technology infrastructure. Aligning server virtualization concepts and infrastructure management tools provide gains in time saving, costs and management when compared to systems without such automation steps.

Thinking about attacks against cyber infrastructures, virtualization is the path used as a way of gaining control over the hypervisor throughout the guest host machine, thus making it possible to compromise a whole infrastructure. According to Wojtczukif [11], if a hypervisor isolates untrusted code that runs in a virtual machine from the rest of the system, an attacker that exploits vulnerability successfully in the hypervisor breaks this isolation, thus gains access to all the resources available to the hypervisor. Evidence of that can be seen in [12] and [13]. Even so, virtualization helps reducing costs, permits scalability [10] and is widely used to build cyber infrastructures.

Trust in information system is subject to many discussions. For example, the Zero Trust Model for Cyber security [14] tells it to stop trusting packets as if they were people. The model of internal and external networks should be

changed because one should assume that all network traffic is untrustworthy because you simply do not know for sure what is in transit in the network. The Zero Trust model claims that internal data should be protected from insider abuse as one protects external data in public networks.

Trust is important to ensure secure and reliable communications. The accuracy of trust evaluation depends whether a trust system can function properly under all situations and be capable of handling unfair rating attacks in a satisfactory manner [15]. It is clear that trust management systems are necessary in order to accomplish security related tasks in distributed systems [16]. The management of trust relationships between different peers can be done using different approaches. Trust relationship can be manually established by each node in a network but it does not scale when the number of nodes increases. In such cases trust models are used to automatically calculate how much a node can trust other nodes. Normally trust values are locally calculated or may be provided by other nodes in the network (reputation).

In general, information security must support business objectives by minimizing risks and developing trust. Information security requires continuous improvement in a cyclic approach [17]. ISO 27000 Family and BS7799 [18] are well known guides in the information security business but fundamentally they do not address trust. To the best of our knowledge, there is no known proven technology or framework in information technology without security issues, such as exploitable flaws, configuration errors or system misuse, all of which leads such systems to be considered unreliable.

The work presented in [19] reviews various information security standards and compares them in terms of information security policy, communications and operations management, access control, information system acquisition, development and maintenance, organization of information security, asset management, information security incident management, business continuity management, human resources security, physical environment security, compliance.

According to [20], an APT is a deliberately slow-moving cyberattack. It is applied to compromise interconnected information systems without revealing itself. APTs use a variety of attacks to get unauthorized access and then spread itself throughout the network. Also, APTs cannot be detected by protection systems that rely on signature-based methods [21]. Therefore, users and cyber infrastructure need proactive defense systems, which are capable of making intelligent decisions.

When dealing with exploits, sometimes it is not possible to perform comparative approach to rate exploits authors because normally they will not show the full scope of their knowledge. Basically, an exploit will carry out the minimal necessary actions to accomplish the task of infecting the target with malware [22]. To work with exploits, technical knowledge is required. Once the target is infected, reverse engineering, malware analysis and network behavior are important as all of them are tools used to try to mitigate and understand what a malware does.

The way this works sees the connections between cyber security, trust, information security and information assurance is shown in Figure 1. Infor-

mation assurance is considered the bigger part of the schema. It has strong relations with trust, information security and cyber security.

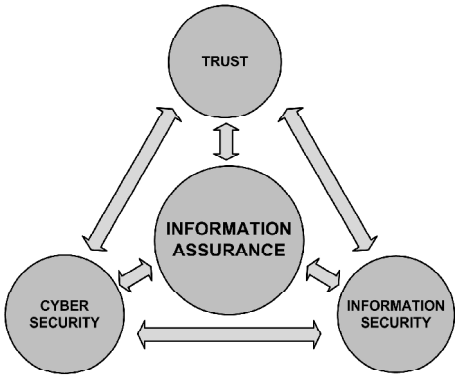


Fig. 1 Relations in information assurance, information security, trust and cyber security.

In general, if one wants to understand and apply information assurance in a particular organization, it is important to remember that the connections among information security, trust and cyber security should also be understood and detailed. This schema will help in understanding the difficulties that exist in protecting information in a global context.

3 Cyber Security and Information Assurance

This section reviews and presents some definitions about cyber security and information assurance.

3.1 Cyber Security

From the perspective of this work, cyber security is considered the set of technologies, processes and practices designed to protect computer networks, computers systems, hardware and software, information and data from attacks, damage or unauthorized access. In information technology, the term security implies cyber security. Cyber security is a field where information security and computer network security are close together. It deals with application security, end-user systems and information.

Due to its characteristics, cyber security constantly changes and requires coordinated efforts throughout an information system. In such scenarios, threats advance quicker than a particular organization can keep up with it. Generally in cyber security, threats change faster than the main idea of the risk itself to

a particular system. Thus, it is very difficult to address risks in cyber security because once it is addressed the threat may already be another. In such perspective, according to [23], ensuring cyber security is not a simple task. It requires domain knowledge and cognitive abilities to determine possible threats from large amounts of network data.

3.2 Information Assurance

Information assurance has a strong link to information security and business continuity. It is an interdisciplinary field that requires expertise in accounting, user experience, fraud examination, business, computer and network forensic science, management, systems engineering, security engineering, and depending on the case, even in criminology, in addition to computer science. It may be seen and understood as a superset of information security.

This work considers that information assurance is an area that seeks to protect and defend information and information systems supported by computer systems and networks. It uses means such as tools, processes and frameworks to try to ensure confidentiality, integrity, authentication, availability, and non-repudiation. The practice of assuring information and managing risks are related to the use, processing, storage, and transmission of information or data and the systems and processes used for those purposes in order to guarantee that authorized users have access to authorized information at the authorized time.

Also this review considers that information assurance includes the use of physical, technical and administrative controls to accomplish the task of data protection. This encompasses not only digital but also analog or physical forms of data. It can be applied to data in transit, as well as data at rest. Information assurance is also a method of adding benefit to business using information risk management activities. It is expected to increase the utility of information to authorized users and, of course, to reduce the utility for those unauthorized.

Considering a wider picture, using critical infrastructure as an example, the cyber security context is embraced by information security itself as part of efforts towards information assurance. As a means of creating a representation of the steps and areas it implicates, Figure 2 resumes this view in a top-down approach.

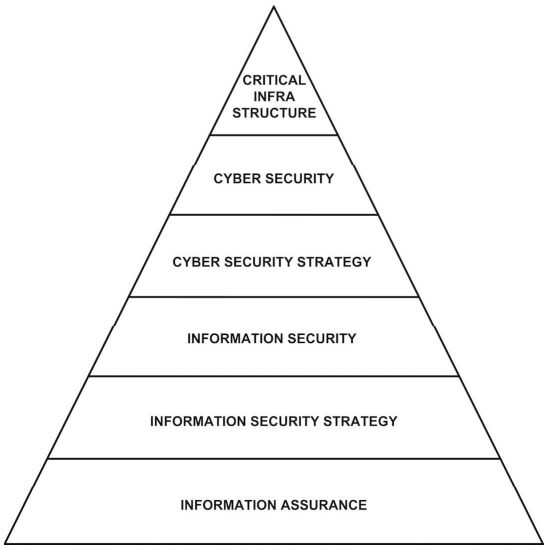


Fig. 2 Implications of cyber security towards information assurance.

3.3 Cyber Security Strategy

Cyber security strategy is a high level plan to achieve conditions of security in the cyber environment, for example, where critical infrastructure systems are placed. Cyber security is seen as a strategic area by many countries. For example, Table 1 details the countries and years when cyber security strategy became part of overall international planning, according to the European Union Agency for Network and Information Security (ENISA) and North Atlantic Treaty Organization (NATO) Cooperative Cyber Defense Centre of Excellence.

3.4 Cyber Strategy Areas of Interest

In order to better discuss and understand what cyber security includes, we divided this section into areas that are important to consider when preparing a strategy to safeguard cyberspace.

Table 1 Summary of since when some countries have overall planning of cyber security strategy.

Country	Year
Qatar, Nigeria, Costa Rica, Jamaica	2015
Czech Republic, Estonia, Denmark, Latvia, Kenya, Namibia, Rwanda,Azerbaijan, Bangladesh, Pakistan, United Arab Emirates, Ghana, Mauritius, Grenada	2014
Austria, Finland, Italy, Hungary, Poland, India, Japan, Montenegro, Singapore, Turkey, Zimbabwe, Romania, Spain, Cyprus, Saudi Arabia, Egypt, Morocco, Panama	2013
Belgium, Norway, Switzerland, Netherlands, Georgia, Jordan, Mauritania, South Africa, Trinidad and Tobago	2012
France, Germany, Lithuania, Luxembourg, United Kingdom, New Zealand, South Korea, Uganda	2011
Canada, Mongolia	2010
Australia	2009
Slovak Republic	2008
Malaysia	2006
United States of America	2003
Russia	2000

3.4.1 Cyber Threats

Cyber threats go beyond aspects that deal with national security or military defense. First have in mind that there is no silver bullet. Cyber threats are related to critical infrastructure systems, computer networks, financial systems, research facilities, industry, etc. Thus, to address cyber threats, integrate efforts, multidisciplinary and specialized knowledge and skilled people are all necessary. The knowledge required to deal with cyber threats requires contextualization, means and indicators, direct and indirect implications, warnings, monitoring new threats to assets and critical infrastructure systems. All this is used together to support strategic decisions and measures what may be applied in each situation.

Such analysis is impossible without humans, because they require deep knowledge in information security and communications, information assets, auditing, engineering, and other necessary components. Thus, to be able to monitor and follow cyber threats, some process may be automated but it is no guarantee of efficiency because, to the best of our knowledge, there is no such automated system that may be able to protect, adapt and learn efficiently from everything that is required to address such problems. On the other hand, the amount of data is such that it is impossible for humans to process it without information systems, security systems, correlational systems, big data analytics, etc., but all these systems cannot be fully automated with current technology.

To deal with cyber threats, the focus has to do with processed information, ordered information, information treatment, cyber intelligence, reliable

sources, information correlation, and decision without complete knowledge, tempested answers, global players, nation-state efforts, organized crime, hacktivism activities, and other subjects, which depends of each case.

3.4.2 Cyber Attacks

This piece of work believes that cyberattacks are offensive maneuvers conducted by individuals or organizations that use technological resources with the intention of deactivation, disruption, data theft, vulnerability exploitation, implant malicious code or cause damage to critical infrastructure systems, computer networks, information systems, hardware, software or any technical resource of the attackers interest. A cyberattack is a planned exploitation of a computational system related to particular technology. It is important to remember that a cyberattack may be just an ordinary attack or a step that is part of a particular cyber campaign, cyber espionage, cyber terrorism or even non-declared cyber war.

Considering observations over time, such attacks will become more sophisticated and disruptive in the future (most of them already are). Normally a cyberattack uses malicious codes, bad assembled packets in networks, disguised packets, hidden codes in computer systems, furtive techniques in order to accomplish its objectives. Cyberattacks also deal largely with exploits and advanced persistent threats, which are the subject of section 5 in this paper.

When a cyberattacks take place, the attention is related to denial of service attacks, distributed deny of service attacks, suspicious code in computational systems, botnets, network misbehavior, reverse engineering, penetration tests, breaking computational codes, among other activities.

3.4.3 Network Flow Analysis

There is no point in arguing how computer networks are widely used nowadays. They have its bases in communication infrastructure and protocols to send and receive data. Cyberattacks make use of computer networks to accomplish data theft, systems exploitation, implant malware and so on. It is important in a cyber security strategy to understand and be able to perform network flow analysis. By using such capability one may be able to understand standard network behavior, detect anomalous traffic, the network flow in order to detect compromise indicators of the computer network.

The focus of such activities relies on behavior network analysis, protocol analysis and understanding communications protocols, collecting and executing network flow analysis, processing network flow reconstruction, temporal analysis, and other activities.

3.4.4 Malware Analysis

In the perspective of this review, malware analysis is the study of suspicious or malicious code using reverse engineering and behavior analysis in order

to understand and discover the code functionalities and the techniques that were used to build such codes. Malware analysis allows one understanding of what the main purpose of malicious code is so it can be used do better deploy security measures and deal with incident response. It is also used to learn how offensive techniques are applied in computational environments.

The effort towards malware analysis is basically divided into static malware analysis and dynamic malware analysis. In depth knowledge is requires as well as skills in reverse engineering, behavior analysis, coding and decoding techniques, among other areas.

3.4.5 Big Data Analytics

The amount of data generated, for example, by the traffic inside an ordinary communication network, is such that it is impossible to be processed or analyzed by simple tools. Much of the data is not structured, or semi structured to be intelligible, but it is not restricted to it. In most cases, even if the data is structured, the amount of raw information is such that one cannot analyze it without algorithms and expert systems.

Big data analytics have the ability to exam large data sets with a variety of data types. Once data is processed it can uncover hidden patterns, unknown correlations, find trends, preferences and other useful information that is important for security analysis. With such tools, security and analysis experts can analyze large volumes of data that may be unused by conventional business intelligence solutions.

This kind of data includes server logs and Internet data, social media content and social network activity, open source content, mobile data, records and data captured by sensors disposed in network points of interest. Big data analytics uses software tools as part of advanced analytics disciplines such as predictive analytics, data mining, text analytics and statistical analysis. Software and data visualization tools are all part of the analysis process.

Big data focus from a cyber security perspective embraces massive data visualization, open source and distributed system analysis, data processing infrastructure, heterogeneous data source, analytics algorithms, parallel processing, anomaly detection, and other research areas of interest.

3.4.6 Underground Cyberspace

Much of the Internet is not indexed in search engines like Google or Yahoo. Projects as Memex [24] aims a break through paradigm in Internet search capabilities. Web environments such as closed forums, unindexed web pages, unavailable public web servers and anonymous networks are sources of information that may help mitigate or even anticipate and prevent cyberattacks.

P2P, BitTorrent, TOR, I2P, closed forums, anonymity networks, deep web and darknet are subjects of interest in cyber security studies.

3.4.7 Cyber Alliances

To be able to go further in cyber security, it is important to have in mind that creating cyber alliances with like-minded actors based on common sets of practices and principles is key to advance in security. With cyber alliances one may be able to share information about threats and responses, train both civil and military defense authorities and conduct joint cyber exercises.

These alliances also permit discussions in cyber terrorism, and best practices to protect critical infrastructure systems. Intelligence and law enforcement agencies may discuss and share practices in searching and blocking threat actors, for example people that are part of organized crime and use black markets to buy and sell illegal goods. Cyber alliances also include private sector, international law enforcement and nongovernmental organizations. In short, cyber alliances should be expansive and go further than state-to-state diplomacy, if a wide view is considered.

The focus of a cyber alliance are related to establishing partnerships with centers of recognized capacity in information security, research and development institutions, founding of solutions of cyber security, joint information security projects, transfer of technology among participants, and others interests.

4 Information Security and Trust

This topic discusses aspects dealing with information security and trust. Then, a layered concept called information security architecture is presented with strong relations to trust.

4.1 Trust Definitions

People believe that most of their online data is secure because the company holding it says so. The work in [25] published in 1999 already stated that experts knew that networked information systems lack trust, but critical infrastructure previously relied on them.

From this perspective, to fully trust something that you did not build may not be the correct approach, even though, people believe their systems, networks, databases, and so on. To trust an information system is basically to claim that the system does and operates as required, aside environment problems, human or operators errors and even attacks. However the most important thing about this approach is that the system wont do things it shouldnt. So to trust information system or a network information system requires more than just assembling parts or components. Due to its characteristics, trust depends on subjective evaluation and requires both context and analysis.

Definitions by Gambetta [26] supported by Lamsal [27] and Dagsputa [28] state that trust is a subjective probability in which an entity believes that

another entity will fulfill the necessary actions to complete a particular job. Such actions are subject to verification and may suffer influence by other entities and may even consider historical evaluations, depending on the entity capabilities. Marsh [29] is among the first to study trust in a computational perspective. Usually, trust can be acquired by empiric observation, by formal proof of the systems properties and other techniques [27]. Once all expectations are fulfilled one may establish trust.

Even though humans use trust every day in social aspects, work, home, etc., trust is not yet widely used as a computational resource in network and information systems. It still lacks research and development to be used as an intrinsic characteristic. The process of trusting depends on the knowledge the trusteer gathers about the trustee. It is not a good approach to trust something without being able to measure it or to fully understand it.

Anyway, trust is indeed an expectation. It is a probability that things will work and keep working as they are supposed to. However when it comes to security, trust should be zero or one, i.e., one trusts in an information system, network, etc. or does not. If security is important "maybe" considerations should be avoided. If something is to be trusted, it must be clearly identified and operate exactly as planned and expected. It also must not do anything it was not supposed to do and must be able to operate continuously. If it is acknowledged that a system has been or may be compromised, this is enough to make it suspicious or untrustworthy.

Trust is also an important aspect in the Internet nowadays. If someone connects to the Internet, somehow this person trusts in hardware, software and networks in daily activities [30]. But these elements are prone to a lot of failures and even when they seem trustworthy they could still be used for purposes other than they were designed for.

There are common aspects in almost all the references about trust models that make us determine a common set of features related to trust. These features are summarised in Table 2, as presented in [16].

Table 2 Summary of trust characteristics

Characteristic	Example
Trust can be measured	Entity A trusts more in entity B than A trusts in entity C.
Trust is context-aware	Entity A may trust B to perform URL filtering but do not trust B to perform authentication tasks.
Trust changes with time	The amount A trust B may grow or reduce as interactions take place.
Trust may be directional	Entity A may trust B, but B may not trust A.
Trust is social-aware	Entity A may trust entity C, because C was introduced by B to A, and A already trusts B.

4.1.1 Group Trust

Group trust is the ability to perform trust calculations in distributed systems seen as a collection of entities connected together with common goals or common contexts. The concept of group is related to a set of entities working together to accomplish a particular task. For example, entities are able to perform specific works in a common context like service offering [16].

Group trust permits those involved in trust context to be able to perform trust and reputation calculation of each other entities in the system that considers any interaction, using any trust or reputation model. In such cases, group trust value is consequence of the individual trust values of the entities participating the group. The added value represents a point of information for external entities so they can use it to infer if the group is trustworthy. Entities in a particular group may be able to agree to a minimum trust threshold in order to make a common analysis and make decisions with trust support.

4.1.2 Trusted Computing

According to the Trusted Computing Group [31], a trusted platform module (TPM) is a secure cryptographic integrated circuit that provides hardware level capabilities to users so they can manage authentication, network access, data protection and other tasks to increase security to a higher level than software-based approach.

Most computer vendors have a chip in its hardware and it takes just a few steps to enable it. TPM may be used do generate keys, fetch keys from a certificate authority, encrypt files, folders, email or even full disk encryption. TPM can also be used to work together with smart card readers so it meets multi-factor authentication requirement.

4.2 Information Security Review

It is no secret that information security is a billion dollar industry and it is still common that information security is not properly understood by organizations. Bear in mind that there is no "holy grail" to deal with information security. If most projects of organizations could be widely reviewed, it would show that most of them start without any security approach, even knowing that risks can compromise the project and the image of the organization. It seems that the big problem is still that people think that because you invested money in your security, your systems and information are safe because of that investment. This is a huge mistake. The fact is no matter how much money is spent in information security, it seems never to be enough to avoid data theft, information disclosure, system exploitation and other risks that are intrinsic to information supported by networks and information systems. It is a cyclical task. It never ends. You have to keep pushing.

Also, if information disclosure is related to security events, one will realize that current information security platforms are not able to properly handle many different facets of information security. Anyway, what is remarkable in information security is that there is a huge amount of architectures, policies, metrics, technologies, etc.; and organizations still fail to address basic information security right.

Research about information security has been published for a long time. A milestone in the discussion on confidentiality is Bell and LaPadula [32]. Regarding integrity, most references quote Biba [33] and his model that describes a set of access control rules designed to ensure data integrity. Regarding availability, some references are found in the National Institute of Standards and Technology (NIST) [34]. Indeed, the exact origins of the Confidentiality, Integrity and Availability (CIA) triad expression appear to be unknown. Apparently they date back to NIST publications in the nineties.

It is important to mention that being compliant with security standards does not guarantee that security will be accomplished either. In other words, compliance with security standards and best practices does not guarantee that an organization is shielded against treats such as data theft or cyber-intelligence campaigns. Information security management requires more than compliance or best practices.

Unfortunately, up to now and to the best of our knowledge, there is no known proven framework or technology for developing application and information systems immune to security issues, such as exploitable flaws, configuration errors or system misuse.

4.2.1 Information Security Basics

The information security basics are widely known as Confidentially, Integrity and Availability, the well-known CIA triad. Availability states that all information systems, networks, databases, information assets, storage mechanisms and so on must be accessible by those who have been granted access to manage them when required. Also, communication channels used to access information, wherever it is, must be operating correctly and accordingly to information security policies.

Integrity is the ability to guarantee accuracy and consistency of data and information during its entire life cycle. Integrity has to guarantee that the information is accurate, reliable and it has not been changed or damaged by an unauthorized party all of the sudden.

Confidentiality is the information security property responsible for preventing unauthorized disclosure of information. It uses principles such as the "need-to-know" and, in order to be effective, confidentiality must ensure that access to vital information is limited only to those who specifically need to have access to or use that particular information.

There is an ongoing debate among information security experts regarding whether the CIA triad is enough to stand as the basis of information security by itself; this piece of work does not believe it is. However, it is also very

difficult to extend all types of information security properties or attributes, thus, ensuring that your information is secure. We picture information security extensions as a group of attributes or properties to protect information and systems, but that are not limited to it. Table 3 summarizes some of them.

Table 3 Information Security Extensions

Principle	Description
Non-repudiation	In short, a party within a communication process cannot deny to have received specific information nor the other party can deny to have provided specific information. The discussion goes beyond the technological field of non-repudiation because one cannot guarantee that digital signature proves authenticity and integrity, thus preventing repudiation; for instance: cases of data theft. The opposite is so called plausible deniability.
Authentication	It is what verifies identities. In a communication process, the party must provide evidence that it is the party the credentials belong to. It can be something you have, some information about yourself, or something you know about.
Privacy	It regards the right to control information about an individual and the right to limit access to that information. It is also related to domains in which individuals have the right to keep confidential information and data, and to share them in private conversation to whom they desire to do so.
Resilience	It is the ability of an information system to keep its minimal service levels guaranteed, even under challenges to its normal operation, attacks or failure of some of its components. It is also a perspective of dependability when facing changes in its operations.
Access control	It is what restricts access to a certain data, information or resource. According to this principle, once access control is guaranteed, the entity or should be able to extract, enter or use a particular piece of information an information system.
Anonymity	It is simply a result of not having identifying characteristics revealed or made available to the public, which would allow the identification of an entity.
Authenticity	It is a way to ensure that communication processes data, systems, or information are genuine.
Authorization	It is the process of providing access to particular information or system to a party based on their identity.

As one may see, these properties are able to support themselves as elements that depend on many other complex characteristics, such as context, technologies that support them, usage objectives, and so on. In short, information security extensions complete the information security basis, and, in addition to that, some of the terms are able to stand by themselves.

4.3 Information Security Architecture

The review provided in [35] describes in short an information security architecture, which integrates security elements and their interfaces as represented

in Figure 3. In general, the trust layer is the piece of the architecture that is used to help in providing enough knowledge about information, systems, technology and other components so as to allow decisions such as "secured" or "information is secure because a particular condition was met". This is precisely the role of a trust layer module in the given architecture. This part of the architecture can be seen as a module that gathers knowledge from all the other component levels and may be used to make decisions which impact the security.

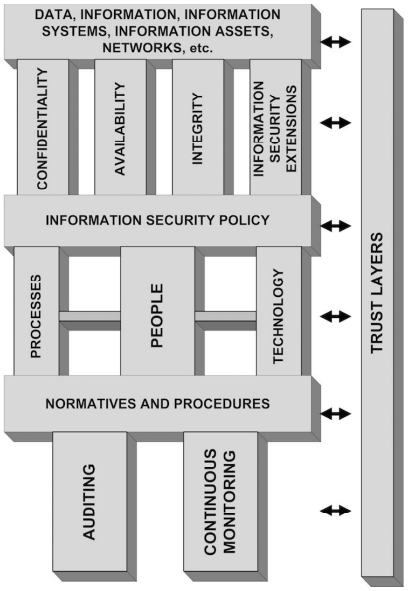


Fig. 3 Information Security Architecture.

It is important to understand the implications that data, information, information assets and so on have on information security. When discussing information security most of the things that have to be protected rely on this layer. Networks and computer systems connect every piece of data, information and information assets so anyone with granted access is able to browse them. Information assets should be identified and labeled, and its relationship to information should be clearly understood. In short, these architecture components create inputs for supporting decisions related to information security.

After that, come the CIA triad and the security information extensions as reviewed in section 4.2.1. Then there is the information security policy, which is a high-level document that plans specific rules that must be met regarding information security. The information security policy is what guides people during the creation of the processes and definition of technological components in order to help to protect information.

Processes are formal mechanisms to identify, measure, manage and control risks related to information or its value to the organization. Processes should not be seen as a black box or something that is meaningless to the organization. It is important that information security processes are well documented and communicated to human resources that should know about them.

People are a piece of the architecture and represents human resources. People are responsible for developing and implementing each part of an information security policy, creating and maintaining processes, information assets, defining which technology should be used, designing networks, etc. It is important to address points such as strategies related to hiring, access, responsibilities, training, dismissal, damages, and whatever is considered important to help maintain the organization's information security strategy.

Technology is the set of all informatics systems, applications, infrastructure, tools, and defense mechanisms that the organization applies to achieve its goals and also to help address information security problems. Technological elements may frequently change and update, become obsolete very fast, or be the core of an organization's infrastructure. Technology may also be used to solve security threats and mitigate risks.

Normative and procedures are fundamental for prioritizing goals, organizing and planning actions in order to define how things should be done. Normative refers to how things should be and how they should be rated while procedures are step-by-step guides that define how things must be done considering specific contexts as guides to what one should do if a particular condition is met.

Auditing is key to discovering risks, technical flaws, policies, procedures and normative problems. Remember that auditing is a never-ending process. Auditing information security is a process that takes qualitative and quantitative measures to assess the current state (snapshot) of what is being audited regarding particular criteria of information security.

Continuous monitoring keeps track of ongoing knowledge of information security, vulnerabilities, threats and associated risks [36]. It relies on technology, processes, procedures, operating environments and people and it is deeply connected to the understanding of organizational risk tolerance. It also helps to set priorities and to consistently manage risk inside the organization.

This summary intends to show how hard and complex it is to create information security architecture. There is no common approach able to measure all peculiarity of known information security architecture [37]. In doing so it may still not completely address every aspect related to information security.

4.4 Information Treatment and its Relation to Trust and Information Security

Basically in technology, data (bits grouped together somehow) exists in three fundamental states: 1) being used (for example in processing state), 2) in transit (for example, being moved from one point to another) or 3) stored

(for example in a storage system). It is in these three stages that information security must act to avoid information theft, disclosure or destruction. The type of information, operations that can be performed and the supporting components must be considered if you want to protect information. In fact, you cannot protect information if you do not understand where it is stored, how it is represented or who or what manages it. Considering a digital environment, how information is digitally treated is the key to protecting it. Figure 4 summarises how information is basically represented, the basic operations to process it and where these activities may happen from a cyber perspective.

If the whole understanding of Figure 4 is considered, it sounds very difficult to think of information assurance or information security. A trust layer is used as a mean to explain when technology or regular approaches are not enough to guarantee security during the entire process. Normally in digital environment types of information can be images, text, voice or sounds, videos (combination of images and sounds), and when it has no particular characteristic, it can be perceived simply as data. Information may experience specific operations such as access, processing, storage or transport using any kind of device (disks, cell phones, tablets, networks, etc.). Any combination of operations is also possible and is common in digital environments. Taking a deeper look, there are many possibilities when it comes to the combination of information type and information operations. Well, one can see that to protect it in all existing scenarios is rather difficult. The supporting components are where or how the information is used or makes sense considering human manipulation.

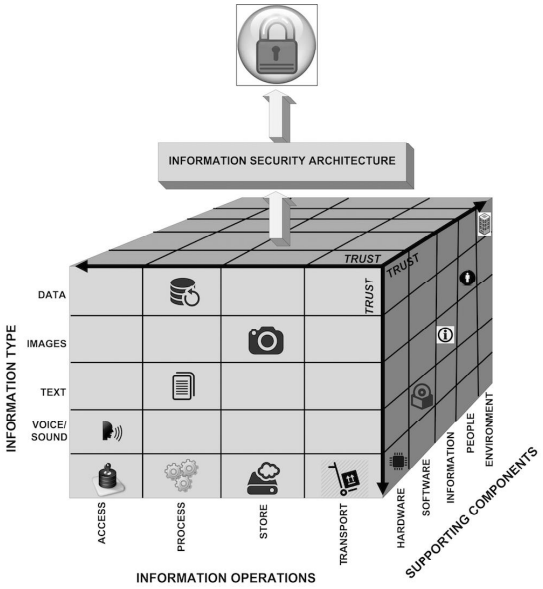


Fig. 4 Information representation and its relation to information security and trust.

Therefore, it should be considered that information security should be guaranteed inside environments where activities such as copying files, printing images, writing data to database systems, using distributed network environments, network connections, users making common mistakes such following unknown hyperlinks, and so on, takes place, which obviously is a very hard task.

It is a complex matrix of at least three dimensions to keep one piece of information secure inside such environments and technology and skilled people are needed to deal with it. In our perspective, we believe it is difficult to achieve information security regarding the cube perspective (Figure 4) without using trust and information security architecture as reviewed in sections 4.1 and 4.3.

5 Exploits and Advanced Persistent Threats

Basically, cyberspace is a place where threats like exploits and Advanced Persistent Threats (APTs) are used to gain benefits, profit or strategic advantage. It all depends on the actor that is behind the deployment of such techniques. Basically this chapter is divided to explain both exploits and APTs.

5.1 Exploit

Exploits are effectively tools used in cyberspace. They can be considered cyber weapons as well, depending on its capabilities and objectives. An exploit can be seen as a threat in a cyber security perspective. The sections below review and explain details about exploits and the market associated with it.

5.1.1 Definition

An exploit is a piece of code developed and compiled with particular hardware architecture, a particular software or application in mind. This code is created with the intention of exploiting a particular vulnerability in some technological resource. In such cases vulnerability is a kind of failure that permits the attacker to have success in exploiting a particular resource, be it hardware or software.

The associated vulnerability may be Zero-day (0day) or maybe a known vulnerability. 0day vulnerability is a kind of failure for which there is no previous defense against it because there is no patch or update for the technology being exploited. Such failure allows attackers, considering its objectives, a high level of success in exploring such resource. Once this failure is publicly known it is no longer a 0day, but it maybe still exploitable because there may be no patch or update for it. In other words, the vulnerability may be widely known but it is not fixed.

Regardless, an exploit makes use of low-level language, may use assembly instructions that are able to manipulate and change the normal flow of execution of a program. Such changes let an attacker executes arbitrary code in the system exploited thus permitting control over the system functions.

From a security perspective, an exploit is a threat that may be used in combination with another piece of code to gain control over the technological resource of interest to the attacker. If an exploit is well deployed and well controlled it allows unauthorized access to computational resources, permitting privilege escalation or denial-of-service to authorized users. If it is badly used, an exploit is a lost resource, may allow identification of the attacker and may be no longer efficient.

An exploit can be local or remote. A local exploit is when the attacker already has access to a particular computational resource and may execute the code locally with the intention of privilege escalation, install some other code or make the computational resource remotely controlled. A remote exploit is when the attacker may execute the code remotely by use of a communication channel or network, thus exploring vulnerability to gain control over a particular system.

In short, an exploit is a technical resource that can hit any user in any cyber environment, either connected or isolated. In isolated environments removable media can deliver an exploit. An exploit alters hardware or software normal functions to accomplish what the attacker desires. Having said that, there is a market where such goods are traded.

5.1.2 Exploit Market

From the perspective of this work, if one considers that an exploit is based on a market approach, it is simply based on the fact that there are buyers and sellers. There is one side interested in buying such goods and there is the seller of such goods. However the use of exploits is very restricted. Normally it is linked to a particular technology. Exploits are supported and developed by organized crime, hacking activities and nation-state actions. There are also specialized actors that research vulnerabilities and create exploits for them.

It is important to understand that an exploit it is not an end in itself. It is a way to perform much bigger activities. Developing exploits is a complex activity, it requires high expertise in information security, it deals with very skilled people, it is about constant development, and may be high-priced in specialized markets. The participation in such markets as buyer requires planning, focus and concrete objectives. It demands a good strategy and needs proper infrastructure to deal with such activities. An exploit may be the guarantee of success in exploring some particular resource. Or it may be the cause of failure if misused.

What basically defines an exploit market is the law of supply and demand. An exploit can be acquired from specialized sources, people may exchange it in closed negotiations, or it may be even sold to Government agencies by private companies. It all depends on the needs of each player. Sometimes it

is a legal market (conducted according to the law) where the prices are very high. Sometimes it is conducted in illegal markets, where there is no control or the law is always forgotten and most of the time this trade is conducted in closed forums, using virtual currency and encryption techniques.

This market is conducted basically by the discovery of vulnerability that the manufacturer normally is not aware of. The value of an exploit for unknown vulnerabilities may reach up to U$ 500.000,00 according to the study conducted by Forbes Magazine [38]. The values are also variable because it all depends of its utility, reach of the exploit or exploitation difficulty. This market is also connected to cyber crime with impact in finance and revenue, which reaches orders of 40% in losses according to the study conducted in [39]. Basically, what is seen in such environments is that when an organization is the victim of a cyberattack, where exploits or an APT are used, it is highly probable that the organization wont be able to mitigate the attack before the damage is already done.

It is very hard to define the real cost of an exploit. Things such as complexity of the exploit, its effectiveness, the target, the opportunity window (see APTs in section 5.2), target localization, penetration difficulty (network perimeter, objective of the attack, etc.) are factors that influence the price of an exploit. Table 4 displays some findings conducted by [38].

Table 4 Mean Price of an exploit according to Forbes Magazine Study in 2012

Product	Estimated Price
Adobe Reader	U$ 5.000 - 30.000
Mac OSX	U$ 20.000 - 50.000
Android	U$ 30.000 - 60.000
Flash or Java Plugins for web browsers	U$ 40.000 - 100.000
Microsoft Word	U$ 50.000 - 100.000
Windows	U$ 60.000 - 120.000
Firefox ou Safari	U$ 60.000 - 150.000
Chrome ou Internet Explorer	U$ 80.000 - 200.000
IOS	U$ 100.000 - 250.000

Another interesting point of the exploits market goes beyond the price of exploits or its kind. It is based on who is paying for such threats. According to a broker known as Grugq, most of the clients are occidental government (USA and Europe) simply because they pay more than China or Russia [38]. Another important thing to remember is that when there are many offers, the price tends to decrease because there is too much competition. Besides that, the quality of such goods may be doubtful if one has not been able to check the effectiveness of what one is buying.

Symantec Labs studied cyber crime and cyberattacks [40]. It showed that, among others information related to security, in the year 2013 there was an increase of 91% in targeted attacks, an increase of 62% of vulnerabilities, more than 552 million identities were exposed, at least 23 0day vulnerabilities of high

impact were public discovered, and one in 392 emails had theft of passwords attacks.

A study conducted by Goncharov [41] shows that many parts of the Russian underground are highly specialized. Hackers with good social network contacts do not have to create all his tools anymore. He may rent or buy it from others. There are experts for almost everything one might need. There are service offerings for deny-of-service or distributed deny-of-service, traffic redirection, pay-per-install, malware development, etc. The study [41] also points that, for example, the price paid by valid credit cards is on the fall (Table 5) as are the prices of stolen credentials (Table 6).

Table 5 Price of stolen credit card data in Russia market [41]

Country of origin of the stolen data	Year		
	2011	2012	2013
Australia	U$ 7	U$ 5	U$ 4
Canada	U$ 5	U$ 5	U$ 4
Germany	U$ 9	U$ 5	U$ 6
United Kingdom	U$ 7	U$6 - 8	U$ 5
United States of America	U$ 3	U$ 1	U$ 1

Table 6 Price of stolen credentials of web services [41]

Service	Year		
	2011	2012	2013
Facebook	U$ 200	U$ 160	U$ 100
Gmail	U$ 117	U$ 120	U$ 100
Hotmail	U$ 107	U$ 100	U$100
Mail.ru	U$ 74	U$ 70	U$50
Twitter	U$ 167	U$ 40	–

There is another market that is part of exploits and vulnerabilities. In 2013, the NSS Labs conducted a study that analyzed data from two major bug bounty programs [42]. The results show that in the last 3 years (before 2013), at any given day, privileged groups had access to at least 58 vulnerabilities reaching systems such as Microsoft, Apple, Oracle or Adobe and those vulnerabilities remained private for at least 151 days. Table 7 details some data reported by the study [42].

Table 7 Mean price paid by bug bounty programs [42]

Company	Price Paid	Description
Google	˜U$ 580.000	Mean price paid for at least 3 years for 501 vulnerabilities disclosure in Chrome web browser. Corresponds to 28% of updates on the same period.
Mozilla	˜U$ 570.000	Mean price paid during 3 years for 190 vulnerabilities disclosure in Firefox web browser. Corresponds to 24% of updates on the same period.
Facebook	˜U$ 1.000.000	Mean price since 2011 for Facebook bug bounty programs.
Microsoft	˜U$ 130.000	Mean price paid since 2013 by Microsoft for new exploitation techniques in its products.

Also, regarding bug bounty programs competition, the CanSecWest Pwn2Own 2015 had payouts for vulnerabilities in four major browsers (Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari). The payouts for all competitors reached U$ 442.500 in two days of competitions [43].

The exploit market continues to run. There are many sites specialized in selling exploits. Such sites are available in networks using anonymous access such as TOR or I2P. As another example, $17,000 in bitcoin was the price of a new method of hacking Apple iCloud accounts [44]. Other prices of exploits to be used against Androids browser, to attack Internet Explorer browser in Windows XP, Windows Vista and Windows 7, are available for around $8,000 in bitcoin [44].

An online article from help net security [45] shows that the price of stolen information sold on underground market from users and corporations also vary depending on the good itself. Table 8 shows some of them [45].

Table 8 Price List of Stolen Information [45]

Type of information sold	Price
1.000 stolen email address	From U$ 0.50 to U$ 10
Stolen Gaming Accounts	From U$ 10 to U$ 15
Stolen Cloud Accounts	From U$ 7 to U$ 8
Scans of Real Passports	From U$ 1 to U$ 2
Custom Malware	From U$ 12 to U$ 3.500
Credit card details	From U$ 0.50 to U$ 20
1.000 Social Network Followers	From U$ 2 to U$ 12
1.000.0000 Verified Email Spam Mail-outs	From U$ 70 to U$ 150
Registered and Activated Russian Mobile Phone SIM Card	U$ 100

5.1.3 Basic Process of Exploit Usage and Development

An exploit can be used as executable code for a particular system architecture (hardware and operational system), can be deployed throughout web pages,

it may be an attached file to an email message, it can be as a text message in a cell phone or it can be embedded in digital files with a lot of different extensions.

Once it is executed, an exploit allows an attacker to carry out the most different actions desired. It can control the system locally or remotely; interrupt its functioning, extract data or do anything that the computational resource may permit.

The development of an exploit goes through specific activities that vary according to the system of interest of the attacker. In order for a particular vulnerability to be exploited, time, skills and domain of the target technology are necessary. Sometimes it may even require an ability to fully duplicate the target system, what information should be gathered by other activities, depending on the case.

If it is considered an action supported by a particular demand, an exploit may be locally developed or bought from sellers in a specialized market. It all depends on the opportunity window of the vulnerably in the target system. The simplified process of exploit deployment can be seen in Figure 5 (adapted of [46]) and Table 9 summarises each process phase.

The opportunity window (Figure 5), where the resource is exploitable, is variable in time, depends on various aspects such as update available time, discovery time, failure complexity, and other. According to a study by Bilge and Dumitras [47], the mean time of Zero-Day detection is about 300 days before the resource is already in phase of exploitation, which gives an idea of the opportunity window time. The duration of attacks that explores Zero-Day vulnerability may last from 19 days to 30 months and normally a Zero-day attack is directed to a particular target, so its discovery is restricted, complex, and depends most of the time of the target it self.

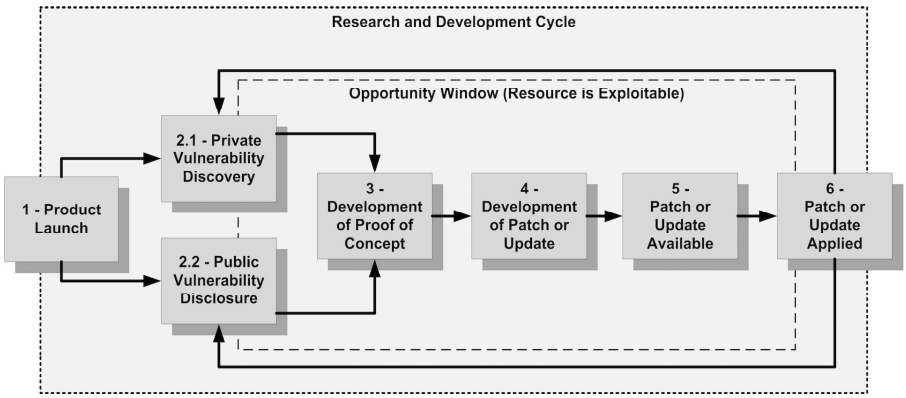


Fig. 5 Exploit process lifecycle.

Table 9 Exploit process lifecycle resume

Phase	Description
1 - Product Launch	A manufacturer makes a product available in information technology business market. This product may be hardware, software, application, operational system, client-server platform or any technological resource.
2.1 - Private Vulnerability Discovery	Through research and development, a private organization or researcher, discovers a particular vulnerability. If it is not made publicly available it is a Zero-day (0day) vulnerability.
2.2 - Public Vulnerability Disclosure	It is when vulnerability is discovered and public made available in the Internet or specialized sources. At this phase, normally the manufacturer is aware of it previously and develops a patch or update for its product before vulnerability announcement in the Internet. Notice that before the patch or update the product is vulnerable to exploitation, which is an indication of the opportunity window.
3 - Development of Proof of Concept	It is when a proof of concept code (PoC) is made available to demonstrate that the vulnerability in question is exploitable. Normally the PoC is a step towards the creation of an exploit.
4 - Development of Patch or Update	The product manufacturer develops a patch or update so it cannot be efficiently exploitable by the known vulnerability.
5 - Patch or Update Available	The product manufacturer makes the correction public available so users can deploy the patch or update in its environment.
6 - Patch or Update Applied	The users do apply the patch or update in their system so it cannot by exploited by the known vulnerability.

5.1.4 Specialized Companies in Exploit's Market

Table 10 resumes some companies and country of origin, which deals with exploits and Zero-Day vulnerabilities and vulnerabilities management systems.

Table 10 Resume of companies that deals with exploits and vulnerability management systems

Companies	Country of origin
Vupen	France
ReVuln	Malta
Netragard, Endgame Systems, Exodus Intelligence, Rapid7, ImmunitySec, CoreImpact, GIF, BeyondTrust, BlueCoat	United States
Hacking Team	Italy
Agt, FinFisher	Germany

5.2 Advanced Persistent Threats (APT)

It seems that the American Air Force General Greg Rattray created the term APT [48]. It was used to designate any adversary engaged in technological conflicts of long duration with planned and defined strategic objectives. In the cyber security area, an APT is part of planned actions and strategy, which are related to specific goals using the cyber environment to achieve its objectives.

An APT has high level of analytic and technical knowledge and uses a large amount of resources. With such characteristics an APT creates means of achieving strategic goals using multiple vectors, be it physical, virtual, social, stealthy, or others. Normally an APT is used to establish and keep continuous covert access, for as long as the attacker desires, inside a targets infrastructure and performing the attackers intentions. Through an APT, an attacker may cause damage, delay actions, destroy information, extract data and information, provoke wrong signals in systems, and change information and system parameters. It all considers a present state or future where the APT is inserted and used.

An APT seeks its objectives during a long period of time, is able to adapt itself to new circumstances, avoids technological defenses, and keeps in constant communication with its command and control system, even using air gap techniques, where the system is not connected to the Internet. An APT is capable of advanced exploitation techniques, uses attacks with no proper defensive measures, and may use 0day exploits as start point of infection.

An APT uses stealthy capabilities so it is not easily detected by common defense perimeter technology. It may change its behavior to bypass detection, may attach itself to another program, and may even alter firmware and device drivers, so it cannot be removed from the infected resource without enormous amount of efforts. Such characteristics make an APT hard to detect thus creating counter measures efficiently is also difficult.

The use of rootkits is also very common when dealing with APTs. Rootkits allows an APT to perform privilege escalation, hide or spread itself, monitor the infected resource, capture network information, and anything an attacker may desire. It basically depends on the attackers resources, intentions and capabilities.

Most known APTs are believed to be work sponsored by nation-state actions because of its capabilities, resources and development process. Also the fact that normally most APTs have specific targets and goals and they are not related to profit points to that direction. This belief has support in facts such as high technology used, high investments, high evasion techniques and very specific targets and goals. Such characteristics are very unusual in common hacking groups or even in organized crime activities. Studies regarding APTs show that to create such tools, it requires large amount of investment.

5.2.1 APT Basic Flow Process

Considering a minimal flow an APT may use Figure 6 illustrates the basic flow steps, but it is not restricted to the following logics. It is divided in 8 steps, and each step is described in Table 11.

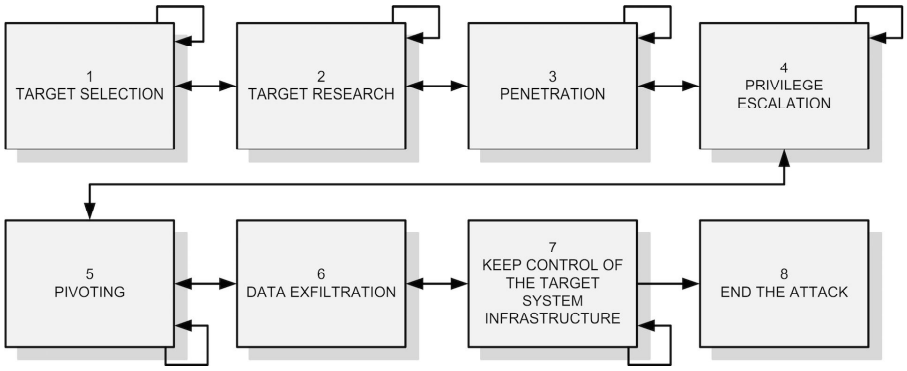


Fig. 6 APTs basic flow process.

5.2.2 APTs, Advanced Malware and Cyber Campaign examples

This topic describes some known malware, cyber campaigns and APTs. These examples are merely informational with no intention of deep analysis. All information listed in this topic is widely available online, even with multiple sources. Indeed, some of the examples below may not fulfill all of what is required to considerer such malware as APTs, but the impact it caused, complexity and targets, among other things was also considered.

– **Titan Rain** was a cyber campaign that targeted American defense companies back in 2003. The investigations traced back the source as being in China. This attack called attention at the time because it already used furtive techniques and trojan horses, also it used multiple attack vectors and social engineering attacks directed to specific users. The nature of this particular attack remained restricted to government sources, but it had

Table 11 APT basic flow process

Step	Description
01 - Target Selection	Normally, selected targets are high valuable users, infrastructure, industry with high research and development technology, diplomatic sources, etc.
02 - Target research	Once the target is chosen, the process starts with information gathering, discovering target capabilities, collect open source information on the target, etc. In this step the objective is to detail the target information, used systems, software, hardware, defense resources, defense perimeters, etc.
03 - Penetration	In this step the target may suffer social engineering attacks, may have some of its system attacked and accessed, may receive e-mail with malware, may receive some threat from social networks relationships, may have passwords captured, etc. The objective is to access some of the target's resources. The attacker may use any means he has depending the target and its objectives.
04 - Privilege Escalation	In this step the attacker may have penetrated some system, but it lacks administrative control on the system being target. It may use exploits, key loggers, network sniffers, etc. to try to gain privileges so it can deploy new resources of its control on the target system. Rootkits may be used in this step.
05 - Pivoting	The attacker spreads itself in the infrastructure of the target so he can control multiple target resources. To be able to perform such actions, the attacker may use sniffers, exploits, information gathering, exploit new systems, etc. The idea is to have multiple sources of access to the targets infrastructure.
06 - Data Exfiltration	Once the attacker controls the system, he performs data exfiltration from the target infrastructure using communication channels, disguised packets, email, command and control systems, etc. In this step the attacker collects network information or any piece of information he considers useful to keep its actions.
07 - Keep control on the	After being successful target infrastructure in taking full control of the target system, the attacker may use rootkits to keep its activities stealthy on the victim system. The attacker may erase previous steps, advance in command and control techniques, search for new data or information, etc.
08 - End the attack	If the objective is fulfilled, the attacker may end the campaign in the particular system and fully erase all of its steps, so an investigation will not be able to fully understand the attack or even find any indication of compromise on the target system.

success hitting industries, research facilities, aerospace agencies, finance institutions, etc.

– **GhostNet** was the name of a cyber espionage campaign that is believed to have started in 2009. It used Trojan horses, command and control systems, and malware that took control over the affected system. It could record audio and video of the victim and it is estimated that more than a 100 countries were attacked.

- **Cuckoo's** Egg is one of the first attacks established in military computer networks. Back in the eighty's, a German hacker invaded computer networks in California to steal secrets related to the American Star Wars research program. The researcher that revealed the malicious activities tracked back to the origin using digital traps and discovered a satellite connection from a German University and identified a student who was likely selling the stolen secrets to the old Russia KGB.
- **Moonlight Maze** was an attack directed to the US government web sites back in 2000. This attack went undiscovered for approximately 2 years and compromised systems in Pentagon, NASA and the American Department of Energy. Also Universities and other military facilities were targeted. This attack was able to steal military installation maps, military hardware projects and other information. The US authorities back at the time estimated the impact of this attack in millions of dollars in lost secrets.
- **Skpyot** was hidden for many years and it is believed to have started in 2006. It was directed to intellectual property theft including projects, manufacturing, finances, and others interests. It used Zero Day exploits back that time and targeted companies in UK and US. The areas of interest include defense, telecommunications, energy, government and chemical industries.
- **Hydraq** was a cyber campaign that used zero-days exploits to infect target systems. This campaign was also known as Aurora. This attack was under detection for at least a year at the time it was reported and targeted technological companies, oil and gas industries and defense sectors.
- **Stuxnet** is probably the most famous APT up to now because it is considered by most as the first developed cyber weapon with nation-station origin and goals. Its detection backs to 2010 but it is public appearance backs in 2008 or 2007 depending on the source. This APT was able to control low level hardware and destroyed hundreds of centrifuges in Natanz Nuclear Facility.
- **Gozi** was a malware that was discovered in 2007 by security experts, and it had already infected more than one million computers. It targeted countries such as US, UK, France, Italy, Germany and others. It was reported that Nikita Kuzmin developed this malware and sold it in the Internet with profit intentions. This malware had a lot of variants with different purposes.
- **Zeus** used social networks as way of spreading itself. It was used to data exfiltration in targets in US and other countries. It is considered a modular platform that uses multiple attack vectors. It is estimated that this malware could have generated more than U$ 70 million in profit for specialized groups.
- **Flame** was firstly public reported by Irans Cert in 2012. It was a sophisticated cyber espionage campaign that affected Government, Universities and specific targets in Iran, Israel, Sudan, Egypt and others. It used USB cards to spread itself and could take screenshots, network traffic capture, key logger capabilities, audio recording, etc. Its command and control system had self-destruction capabilities and the attacks stopped shortly after

it was public reported. It is likely that this campaign remained undercover for at least 5 years.

- **SpyEye** has its origins in the Russia Internet underground. Analysis reported that a command and control system of this malware had data of more than 256 different finance institutions. It is believed that a hacker from Algeria is responsible for the first development of this malware.
- **Doqu** was revealed in 2011 in a limited number of companies that worked with industrial control systems. It had similarities with Stuxnet and a platform known as Tilded. Its command and control systems were spread over countries such as Germany, Belgium, India and China as a source of hiding its origin. This malware could gather information in particular industrial control system that could later be used to attack such systems efficiently.
- **Uroburos** also named snake, is a rootkit based APT malware with two main components, one device driver and one encrypted file system. It was publicly reported in 2014 and had data exfiltration functions, network sniffing, modular design so it can be extended as desired by its creators. This APT has evasive techniques and is hard to be detected by normal methods. It is believed that this malware kept its undercover activities for at least 3 years. Public analysis reported similarities with Agent.BTZ and the cyber campaign codenamed Turla which remained active in 2014 and target systems in United States and Europe, and East European countries as Estonia, Lithuania and Ukraine.
- **Gauss** was first reported in 2012 with its target in Middle East. It was considered a nation-state initiative with rootkit projected to steal sensitive information on the target system. Some analysis reported similarities with Flame as structural modules, code bases, and command and control communication capabilities. It used the same vulnerabilities used by Stuxnet and Flame to infect USB devices. However it had more functions, for example, being able to keep hidden stolen files in USB sticks.
- **Agent.BTZ** is considered one of the worst security failures in US Military Computer Networks. This malware was used to attack the US Defense State Department in 2008. It was left in a lost USB Stick in the parking lot of the department at the Middle East. According to sources, this malware took more than 14 months to be completely removed from the computer network it attacked in Pentagon. Due to its impact it was considered as a military incident and was classified as secret, so very little can be verified using open source information, but it is reported that this malware could scan networks, open backdoors and communicate with command and control systems.
- **Careto**, also named the mask, was reported in 2014. The analysis report showed high levels of sophistication and expertise. Targets in South America, Morocco and Gibraltar were discovered. Targets of this APT were diplomatic offices, embassies, universities and government agencies. This APT can capture keyboard interruption, sniff network, record skype conversations, copy SSH keys, VPN configuration files, and other capabilities.

- **MtGox Bitcoin** attack stole more than 850.000 bitcoins from Japanese MtGox Company. Some security experts consider that this attack remained undercover for years without being detected. Later investigations found 200.000 bitcoins in a wallet that was assumed to be inactive. The impact of this attack at the time was in magnitude of U$ 300 million.
- **BlackPOS** was a malware used to attack the American discount retailer store Target. This attack expected to reach more than 100 million credit card customers. It was able to read credit card data from memory, send the data to a gateway and then to a command and control server at Russia. The investigation showed that the attack started at a third party company. From there the point-of-sale systems were infected.
- **RSA SecureID** attack was able to steal credentials used by the token SecurID. Companies listed as the 500 biggest companies by Forbes magazine used this token. Sources reported that the actions taken to mitigate the damage had costs more than 100 million dollars. This malware used part of another known malware named Poisonvly, which had already attacked chemical companies and human right institutions.
- **Red October** was a cyber campaign designed to steal secrets from Government and research laboratories. It was reported that the campaign was active for at least 5 years before being disclosed. Its victims were in more than 35 countries with focus on Energy, Military, Diplomacy, Space. Systems at North Atlantic Treaty Organization (NATO), Europe Parliament and Europe commissions were infected.
- **Volatile Cedar APT** is part of a group that deploys remote access tools and USB propagation components and performs targeted and managed cyber campaigns. Their targets are chosen with care and the deployment takes only the necessary to achieve the attackers goals supported by previous intelligence gathering processes. This APT can connect to command and control servers or other domains accordingly to the infection process.
- **Eurograbber** was a malware that was used to steal more than 36 million of euros from customers from at least 36 different finance institutions in Italy, Spain and Netherlands. It used Trojan horses and used a previously known malware variant. This malware could bypass SMS systems codes asking users to install security software on their mobile devices.
- **Shamoon** was a cyber campaign using malware targeting the energy sector. It could spread itself using network shares and perform alterations at the infected system. The oil company Saudi Aramco was the biggest target having approximately 75% of its infrastructure compromised.
- **Operation DeputyDog** used Zero-Day exploits in Internet Explorer browser where its main targets were users in Japan. This cyber campaign was reported in 2013 and had strong relations with attacks directed to BIT9 company.
- **NR4** cyber campaign was reported in 2011 and attacked mainly government institutions and diplomatic offices. It used fake email accounts and its messages with political body to call attention of its victims. It had a

command and control system that was used for data exfiltration. Messages used in this campaign target both English and Chinese language users.

- **The Sony case** was made public at November 24, 2014. However it is believed that this attack started more than a year before. This attack exposed data from employees, executives, films and much more information from Sony Corporation. The US intelligence service accused North Korea as being responsible for the attack, which arouse many suspicious about the veracity of the accusation. This attack was discovered because a malware previously installed, rendered many computers inoperable at the time.
- **Equation Group** was exposed to the public in February 2015. It was reported as the most advanced cyber espionage platform known to the time, being operated by a highly sophisticated threat actor involved in multiple computer network exploitation operations, dating back to 2001 and maybe earlier. Its modules use high technological capabilities, strong encryption and before it is fully deployed in the computer victim, it first checks with its command and control system if the target is correct. Versions in multiple systems were found.
- **Trojan.Laziok**, recently discovered, is used as a reconnaissance tool that allows attackers to gather information and tailor their attack methods for each compromised computer. This malware was part of a targeted attack campaign against energy companies around the world, with a focus on the Middle East.
- **Barbar** malware first appeared in 2009. Reports states that babar is part of cyber espionage campaign named Animal Farm Group or Snowglobe. This malware is able to steal keystrokes, clipboards and listen to Skype conversations among other functions. It is reported by some online sources as being controlled by the France Intelligence Agency. It appears to have been used in actions against Syrian targets using zero-day exploits hosted on a Syrian Government website. Analysis showed that the authors of this malware had in depth knowledge of the how some antivirus products worked.
- **Regin** malware is a sophisticated toolkit widely reported in 2014. This malware was used in a cyber espionage campaign that target Belgian Belgacom telecommunication provider. The first samples dates back 2003. Sources from the news point to UK and US as the origin for controlling this malware. Infections were found in Russia, India, Mexico, Ireland, Austria, Afghanistan and other countries. This malware was compared to Stuxnet and developed as being a multi-purpose data collection tool.
- **Dragonfly**, also named Havex, successfully managed to spy on strategic organizations. If the affected systems were explored the way they could have been, it could have caused damage to energy supplies in those who were affected. Their targets include the energy sector, major electricity generation firms, petroleum pipeline operators, and energy industry industrial equipment providers. Victims were located in the United States, Spain, France, Italy, Germany and others. This cyber operation caused companies to install the malware when downloading software updates for computers running Industrial Control Systems (ICS) equipment.

– Germany recently suffered from a target attack in a steel mill facility but the malware or APT did not have its common name reported. The attack manipulated and disrupted control systems to such a point that a blast furnace could not be properly shut down, causing damage to the system. The attackers infiltrated the corporate network using a spear-phishing attack, than after gaining access to the network they explored other networks and compromised multiple different systems, including industrial components on the production network.

6 Conclusion and Future Work

This paper reviewed information security architecture, trust, cyber security, advanced persistent threats, exploits and its market. Government, organizations and enterprises states that cyber security management needs a systematic approach to consistently address security in every layer, reducing unmanaged risks and improving operational security efficiency.

If one takes a deeper look in the area of information security, just cryptography by itself is not enough to keep information as safe as many may think. Inside an environment where more people, computer networks, technology, wearable devices are connected together, due care with information security should be much more than network perimeters, antivirus, intrusion detection system, intrusion prevention systems, etc. It is a continuous task.

Trust has to keep advancing in cyber environments. It is no longer possible to trust software or hardware without taking careful steps towards information security. Cyberattacks on infrastructure where human lives are at risk and people using APTs and exploits causing harm to innocents are concerns that must be treated properly. The correct understanding of cyber security is fundamental to increase the security level of critical infrastructure systems to avoid harm to humans and assets.

In a connected world, cyber space is also a place where technology is used and developed by highly specialized players, with high capability that explores vulnerabilities to gain strategic benefits in many fields of knowledge. Many countries face daily threats to their information assets. Most of them do not even know they are already compromised.

Technology will keep changing the way we live and will keep allowing people and organizations to get connected. In this highly technologically linked world real threats are being forgotten in many situations. Vulnerabilities will keep coming up and hackers will keep exploring them. Nation-states will keep their activities to gain knowledge over what interests them. That has a lifecycle that will require even more knowledge as technology advances. Information security will keep increasing its importance even more as the cyber space keeps growing. Information security is a mean to achieve an end [49]. It is in constant development because of the ever-increasing scale and complexity of security threats.

APTs and exploits will keep resurfacing. There will be more research in areas such as deep packet inspection, operational systems, hardware, software, wearable devices, gadgets, and as industries creates new dependable technological products, the more APTs and exploits will be used to bypass protections, controls and security measures to perform data exfiltration and provide strategic advantages to those who use them. The way Internet is nowadays, to certainly identify the origin of an attack is highly improbable for those who are the victims.

Considering further security research it is important to develop some new manner of dealing with computational trust in cyber space, for example, developing trust protocols, trust metrics, advance in cyber security strategies, etc. It is also important to carry out thorough research to address trust and its place along information security.

Acknowledgements Rafael and Robson acknowledge the Laboratory for Decision Technologies at the University of Brasilia (LATITUDE/UnB) for its support to this work. Rafael would like to thank the support provided by the PNPD/CAPES - Programa Nacional de Ps-Doutorado/CAPES in Brazil.

References

1. D. Geer. Cybersecurity as Realpolitik (2014). URL https://www.blackhat.com/us-14/video/cybersecurity-as-realpolitik.html
2. G. Greenwald, E. MacAskill, L. Poitras, The Guardian News and Media Limited **9** (2013)
3. J. Menn. Politics Intrude as Cybersecurity Firms Hunt Foreign Spies (2015). URL http://mobile.reuters.com/article/idUSKBN0M809N20150312?irpc=932
4. ENISA. European Union Agency for Network and Information Security. National Cyber Security Strategies in the World (2015). URL https://www.enisa.europa.eu/activities/Resilience-and-CIIP/national-cyber-security-strategies-ncsss/national-cyber-security-strategies-in-the-world
5. G. Miller. CIA Plans Major Reorganization and a Focus on Digital Espionage (2015). URL http://www.washingtonpost.com/world/national-security/cia-plans-major-reorganization-and-a-focus-on-digital-espionage/2015/03/06/87e94a1e-c2aa-11e4-9ec2-b418f57a4a99_story.html
6. S. Shah, B.M. Mehtre, International Journal of Electronics Communication and Computer Engineering **4**(6) (2013)
7. E. Byres, J. Lowe, in *Proceedings of the VDE Kongress*, vol. 116 (2004), vol. 116, pp. 213–218
8. A. Teixeira, S. Amin, H. Sandberg, K.H. Johansson, S.S. Sastry, in *49th IEEE Conference on Decision and Control (CDC)* (2010), pp. 5991–5998
9. S.L. Harrington, Richmond Journal of Law & Technology **20**(4) (2010)
10. R. de Oliveira Albuquerque, L.J. García Villalba, O. Ribeiro Torres, F.E. Gomes de Deus, in *Autonomic and Trusted Computing* (Springer, 2011), pp. 75–91
11. R. Wojtczuk, Black Hat USA (2014). URL https://www.blackhat.com/docs/eu-14/materials/eu-14-Wojtczuk-Lessons-Learned-From-Eight-Years-Of-Breaking-Hypervisors.pdf
12. N. Elhage. Virtunoid: A KVM Guest - Host Privilege Escalation Exploit (2011). URL http://media.blackhat.com/bh-us-11/Elhage/BH_US_11_Elhage_Virtunoid_WP.pdf
13. M. Seaborn, T. Dullien. Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges (2015). URL ://googleprojectzero.blogspot.com.br/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

14. The National Institute of Science and Technology (NIST). Developing a Framework to Improve Critical Infrastructure Cybersecurity (2013). URL http://csrc.nist.gov/cyberframework/rfi_comments/040813_forrester_research.pdf
15. D. Wang, T. Muller, A.A. Irissappane, J. Zhang, Y. Liu, in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, 2015), pp. 791–799
16. R. de Oliveira Albuquerque, L.J. García-Villalba, T.H. Kim, International Journal of Distributed Sensor Networks (2014)
17. Department of Communications, Information Technology and the Arts and the Trusted Information Sharing Network. Secure your information: Information security principles for enterprise architecture (2007). URL http://www.tisn.gov.au/Documents/Secure_Your+Information+-+Information+Security+Principles+for+Enterprise+Architecture+-+Report.pdf
18. M. Whitman, H. Mattord, *Management of Information Security* (Cengage Learning, 2013)
19. R. van Os, Comparing Security Architectures: Defining and Testing a Model for Evaluating and categorising security architecture frameworks. Master's thesis, Department of Computer Science, Electrical and Space Engineering, Lule University of Technology (2014)
20. I. Friedberg, F. Skopik, G. Settanni, R. Fiedler, Computers & Security **48**(0), 35 (2015)
21. E. Gandotra, D. Bansal, S. Sofat, in *Intelligent Computing, Communication and Devices*, *Advances in Intelligent Systems and Computing*, vol. 308, ed. by L.C. Jain, S. Patnaik, N. Ichalkaranje (Springer India, 2015), pp. 247–253. DOI 10.1007/978-81-322-2012-1_26
22. G. Szappanos. Exploit This: Evaluating the Exploit Skills of Malware Groups (2015). URL http://media.scmagazine.com/documents/105/sophos-malware_group_exploit_s_26091.pdf
23. N. Ben-Asher, C. Gonzalez, Computers in Human Behavior **48**, 51 (2015)
24. Memex (domain-specific search). Information Innovation Office; Darpa (2014). URL http://www.darpa.mil/newsevents/releases/2014/02/09.aspx
25. F.B. Schneider, et al., *Trust in cyberspace* (National Academies Press, 1999)
26. D. Gambetta, Trust: Making and Breaking Cooperative Relations **2000**, 213 (2000)
27. P. Lamsal, Department of Computer Science, University of Helsinki, Finland (2001)
28. P. Dasgupta, Trust: Making and Breaking Cooperative Relations **4**, 49 (2000)
29. S.P. Marsh, Formalizing Trust as a Computational Concept. Tech. rep., Department of Computing Science and Mathematics, University of Stirling. Doctorate Thesis (1974)
30. T. Wadlow, Queue **12**(5), 30 (2014)
31. [31] Trusted Computing Group. How to Use the TPM: A Guide to Hardware-Based Endpoint Security (2014). URL http://www.trustedcomputinggroup.org/files/resource_files/8D42F8D4-1D09-3519-AD1FFF243B223D73/How_to_Use_TPM_Whitepaper_20090302_Final_3_.pdf
32. D.E. Bell, L.J. LaPadula, Secure computer systems: Mathematical foundations. Tech. rep., DTIC Document (1973)
33. K.J. Biba, Integrity considerations for secure computer systems. Tech. rep., DTIC Document (1977)
34. J.H. Burrows, Guideline for Computer Security Certification and Accreditation. Tech. rep., DTIC Document (1983)
35. R. de Oliveira Albuquerque, L.J.G. Villalba, A.L.S. Orozco, F. Buiati, T.H. Kim, Sensors **14**(12), 22754 (2014)
36. K. Dempsey, et al., NIST Special Publication pp. 800–137 (2011)
37. H. Susanto, M.N. Almunawar, Y.C. Tuan, International Journal of Electrical & Computer Sciences IJECS-IJENS **11**(5) (2011)
38. A. Greenberg, Forbes, Mar (2012)
39. HP Research. Cybercrime Costs Rise Nearly 40 Percent, Attack Frequency Doubles (2012)
40. Symantec Labs. 2014 Internet Security Threat Report (2014)
41. M. Goncharov, Cybercriminal Underground Economy Series (2014)
42. S. Frei, NSS Labs p. 14 (2013)

43. C. Brook. All Major Browsers Fall at Pwn2Own Day 2 (2015). URL https://threatpost.com/all-major-browsers-fall-at-pwn2own-day-2/111731
44. G. Andy. New Dark-Web Market Is Selling Zero-Day Exploits to Hackers (2015). URL http://www.wired.com/2015/04/therealdeal-zero-day-exploits/
45. Help Net Security. Attackers Use Deceptive Tactics to Dominate Corporate Networks (2015). URL http://www.net-security.org/secworld.php?id=18208
46. Tiedata. What are Web Based Exploits? (2014). URL http://www.tiedata.com/webexploits.asp
47. L. Bilge, T. Dumitras. Before We Knew It - An Empirical Study of Zero-Day Attacks in The Real World (2012)
48. S. Gold. APTs: Not as Advanced as You Might Think (2014). URL http://www.scmagazineuk.com/apts-not-as-advanced-as-you-might-think/article/345953/
49. T.R. Peltier, *Information Security Fundamentals* (CRC Press, 2013)