Author(s): Ouedraogo, Moussa; Khadraoui, Djamel; De Rémont, Benoît; Dubois, Eric; Mouratidis, Haralambos.
Article title: Deployment of a security assurance monitoring framework for telecommunication service infrastructures on a VoIP system
Year of publication: 2008
Citation: Ouedraogo, M. et al. (2008) 'Deployment of a security assurance monitoring framework for telecommunication service infrastructures on a VoIP system' in proceedings of NTMS'2008, The 2nd International Conference on New technologies, Mobility and Security, 5-7 Nov 2008, Tangier, Morocco.
Link to published version: http://dx.doi.org/10.1109/NTMS.2008.ECP.38
DOI: 10.1109/NTMS.2008.ECP.38

Deployment of a security assurance monitoring framework for telecommunication service infrastructure on a VoIP system

Moussa Ouedraogo*, Djamel Khadraoui, Benoît De Rémont, Eric Dubois, Haralambos Mouratidis**

Public Research Center Henri Tudor - 1855 Kirchberg/Luxembourg

*PhD student at the university of East London, England, United Kingdom, currently working at the CRP henri Tudor.
** University of East London, Dockland campus, E16 2RD, United Kingdom

Emails: {moussa.ouedraogo, djamel.khadraoui, benoit.deremont, eric.dubois}@tudor.lu, H.mouratidis@uel.ac.uk

ABSTRACT

In today's world where most of critical infrastructures are based on distributed systems, security failures, even within big corporations, have become very common. A system with security loopholes can be damaging for companies, both in terms of reputation and finances, while customers are reluctant to use such systems.

In that respect, providing stakeholders with quantifiable evidences that the security countermeasures deployed on the system are operating adequately is an important step towards better control of security failures for network administrators on one hand, and an increase in end users' trust in using those systems on the other. It is in that perspective that the Bugyo methodology [2] was proposed to address the shortcomings of existing security assurance and risks management methodologies in measuring, documenting and maintaining security assurance of telecommunication services. In this paper, we provide an overview of the Bugyo methodology and we specially demonstrate its applicability on a VoIP service infrastructure based on open source components.

KEYWORDS: Security assurance, metrics specification, probes, measurement, aggregation, Telecommunication infrastructures, VoIP service, risks mangement

1. Introduction

The need to provide stakeholders with confidence that deployed security countermeasures meet their requirements at runtime has been acknowledged as a crucial issue [1] [2] [3]. This is mainly because, security countermeasures, even properly elucidated during the risks management stages, may be deployed inadequately or hazards in the system environment may render them less effective. When dealing with

complex and critical systems such as telecommunication infrastructures, providing continuous monitoring so to get some assurance on the well-functioning of the system is paramount. This is due to the fact that such infrastructures are exposed to a continuously and permanently increasing set of risks and security threats [1] and furthermore, that recurrent telecommunication failures can negatively impact on the reputation of the service provider leading often to financial losses.

Current state of the art in security assurance has revealed the existence of several commercial initiatives (CYBERTRUST, Network security consulting, etc...), standards ([3],etc..), risks management methodologies ([8][9][10][11][12]) as well as methodology proposed by the scientific community ([4][5][13][14],etc..).

However none of these methodologies fully addresses the issue of security assurance in telecommunication services or when they do, focus is only put either at products or systems level [3] or on the IT networks [4][5]. This has prompted the need for a new approach in measuring, documenting and maintaining security assurance for complex systems such as telecommunications services. In view of bridging that gap, The Bugyo (**Bu**ildin**g** Securit**y** Assurance in **O**pen Infrastructures) methodology has provided a framework that focuses on the telecommunication infrastructure and services to address the problem of security assurance in telecommunication. The methodology builds upon a well-known security assurance standards (the common criteria) [3] and aims at verifying that the services are securely provided through the infrastructures in addition to providing stakeholders with quantifiable measures on the reliability of the security countermeasures. In other words, Bugyo helps in providing security assurance, here defined as "*the ground for confidence that an entity meets its security objectives"*. The gathering of that measurable evidence is facilitated by the specification of metrics which are necessary for the normalization of the security assurance levels.

In this paper, we demonstrate how the Bugyo methodology can be applied in computing the assurance level of a VoIP service infrastructure through the specification of metrics.

The remainder of the paper is structured as follows. An overview of the Bugyo framework is provided in section 2. Section 3 focuses on the specification of the metrics and the evaluation of security assurances. An application on a VoIP service infrastructure is also demonstrated in this section 4 concludes the paper.

2. The Bugyo operational methodology

2.1. Steps of the methodology

The BUGYO methodology has three distinct functions that are mainly measurement, monitoring and assistance. To help in fulfilling those functions, the operational methodology [6] has been divided into six steps as depicted in figure 1. For paper limitations, stages of the methodology that are considered less relevant to the focus of this paper have been described just briefly.

2.1.1 Service modeling

The first step concerns **services modeling**. The modelling allows decomposing the service in order to identify assurance critical components. An efficient way of identifying those critical components is an a priori use of a risk assessment methodology. Weights are then assigned to each infrastructure object to account for their respective impact on the overall service

2.1.2. Selection of metrics

The second step is concerned with **selecting the metrics**. Metrics are used to measure the effectiveness of the deployed security countermeasures. The higher the level, the higher the confidence provided. It is however important to notify that a higher level of assurance will require more effort in deploying, monitoring and maintaining the probes.

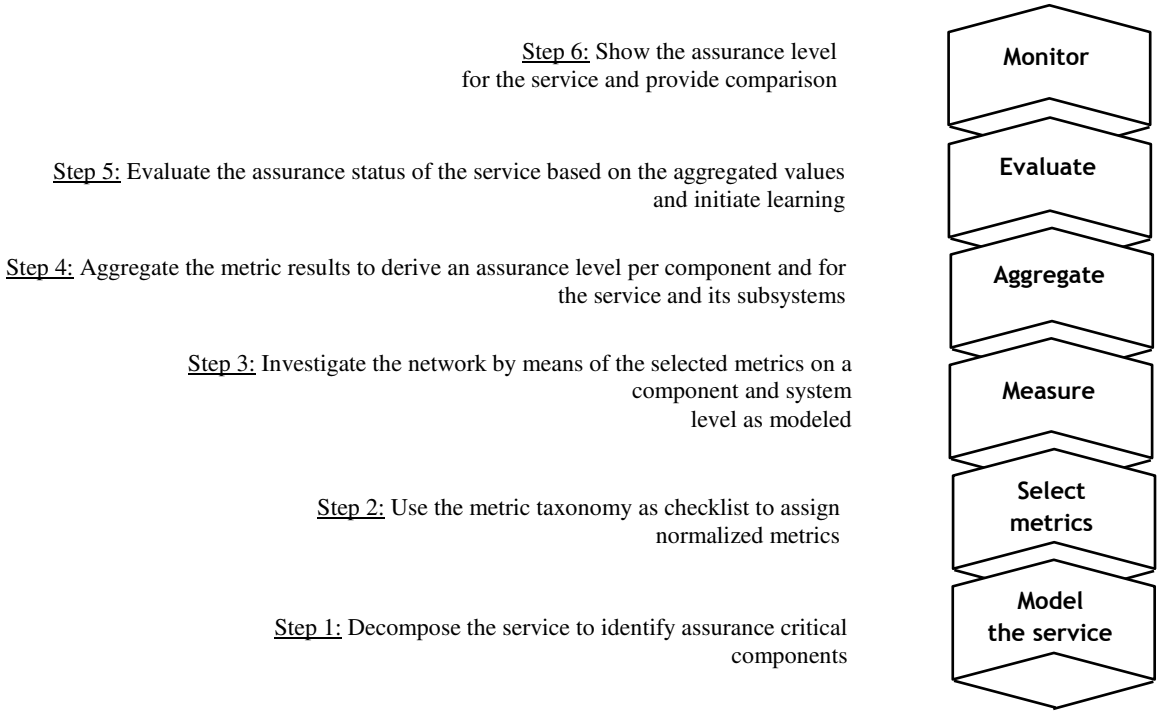

Figure1.Steps of the Bugyo methodology

To allow for a clear definition of assurance levels, a security assurance taxonomy is defined by BUGYO. According to the assurance taxonomy, the distinction between assurance levels is based on the **Scope** (broader scope gives more assurance), **rigor** (More details investigated gives more assurance), **quantity** of

the verification of the security functions and the **timeliness** and **reliability** of the probes.

Unlike the common criteria [3] that defines seven levels of assurance levels; BUGYO justifies the use of only five assurance levels base on the following pragmatic reasons:

- We considered it important to have an odd number of assurance levels so that it is possible to have a medium assurance level (i.e. although only ordinal, the scale is required to have a conceptual middle point).
- The CC evaluation assurance scheme contains seven evaluation assurance levels. However, to our knowledge (almost) none of the system has been evaluated higher than level five. With our scheme, we do not intent to reproduce levels that are not reachable. In general, systems are highly complex and it is unlikely for us that they can satisfy formal evaluation methods. Therefore it did not seem suitable to include levels that are only achievable formally.
- To provide only three assurance levels did not provide enough granularity in our view as our intention is (a) the highest assurance levels should be very hard to achieve (but not impossible) and (b) the lowest assurance level can be achieved by setting up a control system (as it provides already a basic level of confidence).

Our basic postulate is that assurance levels are ordinal (i.e. ordered with an undefined distance between values). A second postulate is that a higher level, we call it B, includes all assurance indications (i.e. requirements) of the level below it, which we call A.

$$A < B \Rightarrow \{A\} \subset \{B\} \qquad \textbf{(Equation 1)}$$

To be complete (and compliant with our intention of an ordinal scale) we have to define an assurance level AL0 with no assurance indication (AL0 is the empty set)

$$AL0 = \{\} \qquad \textbf{(Equation 2)}$$

Assurance Level	Definition
AL1	Rudimentary evidence for parts
AL2	Regular informal evidence for selected parts
AL3	Frequent informal evidence for selected parts
AL4	Continuous informal evidence for significant parts
AL5	Continuous semi-formal evidence for the entire system

Table1. Assurance level table

For each security function (that performs a countermeasure), a metric is evaluated. A metric is a set of measures belonging to one of the three categories: **Availability, Compliance and Vulnerability.**

There are two levels of aggregation to obtain the assurance metric of a security function (figure 2). The first one aims to aggregate measures in meta-measures (availability, compliance and vulnerability). The second one aims to aggregate these meta-measures. The aggregation functions might be generic or specific to a

metric.

2.1.3 Measurement

During the third step, **measurement**, specific probes implementing metrics are deployed through the network. Those probes will help capture raw data from the infrastructure referred to as **base measure** in conformity with ISO 15939 [7]. Depending on the level of knowledge that the expert developing the probes hold on the security function, a measurement can be identified as **white box measurement** (he/she possesses the knowledge) or **black box measurement** (lack of the knowledge). By comparing the base measure to a reference measure, the concerned probes will evaluate whether a security function is operating in an effective way. The evaluation process will result in a **derived measure** that will then be normalized to produce an assurance level.

2.1.4 Aggregation

Once the assurances level are determined at components level, an **aggregation** process is undertaken during step four, using a linear or non-linear algorithm, to compute the overall security assurance at infrastructures and service level. There are three main algorithms used for the operational aggregation [6]: **the recursive minimum algorithm, the recursive maximum algorithm and the recursive weighted sum algorithm**.

The **recursive minimum algorithm**, applied to systems with several critical points, implies that the overall assurance of a service is represented by the lowest metric of its components while the **recursive maximum algorithm**, however, commands that the assurance of a service be the highest of its components metrics. The **recursive weighted sum algorithm** is used for systems composed of security functions that contribute in different, non critical manner to the required service security implemented by independent observed system parts. It uses weight properties model to calculate the assurance level value at each level of the assurance model of a service and is more representative of the assurance needs described in the model.

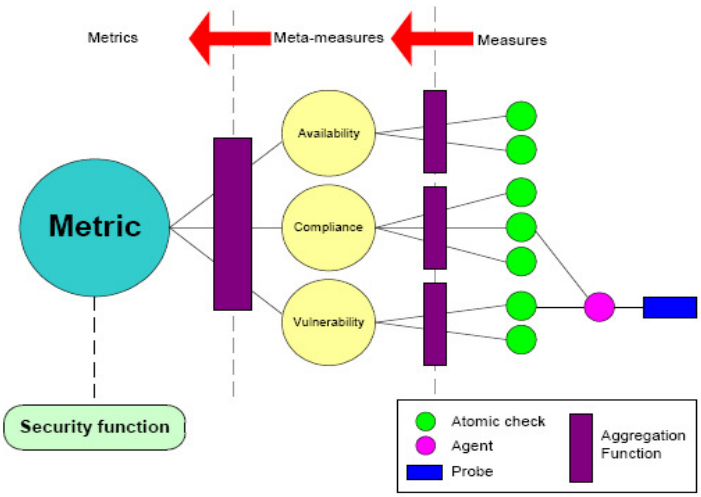

Figure2. Metrics representation

2.1.5. Evaluation

The **evaluation** process consists in comparing the current value of the assurance level to the previous measure or to a certain threshold and issuing an appropriate message. In addition, it helps the operator in making appropriate decision based on the evaluation result.

2.1.6. Monitoring

The final step or **monitoring** step is concerned with providing a real time display of security assurance of the service to help the operator identify causes of security assurance deviation and also assist him/her in making decisions.

2. 2. Architecture of the operational methodology

The security assurance system is composed of three main components: the **security cockpit**, the **security assurance server** and the **assurance measurement framework** linked between then as shown in figure 3. The **security cockpit** is made of three distinct screens, one displaying a real-time value of the security assurance for each service, another presenting measurements history and other details so to enable the operator to appreciate on the deviation of the security assurance value. The last screen displays relevant information for the operator to take action in order to bring the security assurance level to an acceptable level.

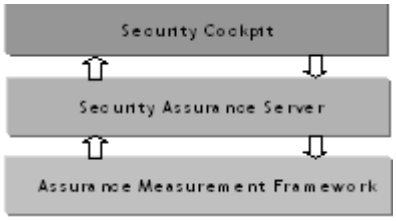

Figure3. Architecture of the security assurance system

Operation such as base measures interpretation; normalization and aggregation take place at the **server** level.

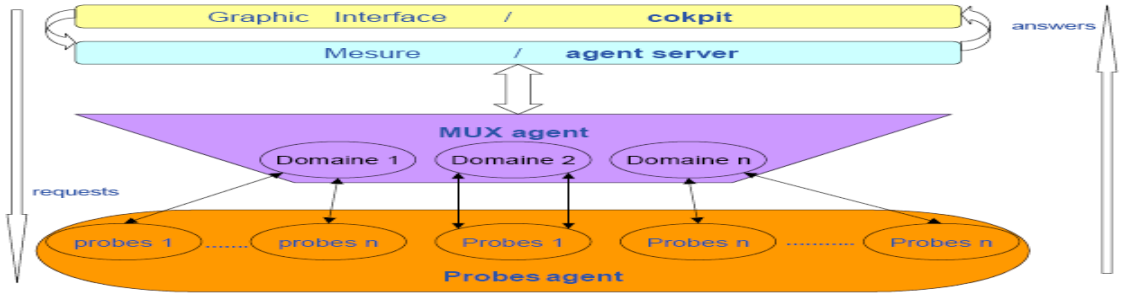

Figure4. Multi-agents architecture for the measurement of security assurance

As soon as measures results are received, they are consolidated in order to evaluate the new service assurance level. It is the server which is responsible of issuing alert messages when the security assurance reaches a critical level. The server can be interfaced with existing tools such as network Management Systems or Security Incident Managers for an efficient coordination of the security management.

The **assurance measurement platform** role is to gather the measures for the computing of the security assurance level. Owing to the high complexity of telecommunication infrastructures and the distributed nature of these systems, BUGYO proposes multi- agents system architecture for the accomplishment of the measurement. The hierarchical structure of that architecture is depicted in figure 4 where a **server agent** receiving the server requests and determining the measures to be performed and the sub domain where the target network entity is; **mux-agents** responsible for dispatching the measure requests to the concerned **probe-agents** (adaptation interface between the probes performing the measures and the assurance measurement framework). The probe-agents perform the measures and send the results back to the server through the mux-agents.

2. Application to a VoIP infrastructure

The Bugyo framework is applied on a VoIP platform (figure 5), based on open sources components, and composed of several servers:

- The IPBX server (open source in license GPL) allows interoperability between a phone switch deployed on Linux and:

- Message recorder
- Conference call
- Calls Distribution

➢ The DNS server.

➢ RADIUS and TFTP server for the identification of the users

➢ Virtual machines (VMware) that has the following advantages:

- Decreasing the number of needed physical machines
- Portability of the virtual machines
- Linux operating system on a Windows machine (and/or inversely)

➢ Soft-phones and hard-phones allowing testing our architecture**.**

➢ Classical network components such as Firewall, routers and VPN gateways.

The deployment of the methodology is based on the usage of a Multi Agents System (MAS) platform using Jade.

For the purpose of applying the methodology, we consider two scenarios that will enable the specification of metrics. In the first case, errors have been injected into the address resolution file for waiter DNS Bind9. The

second scenario is about modifying the configuration files for waiter TFTP. It is important to notify that this demonstration mainly focus on step 2-5 of the methodology, especially on the specification of metrics.

2.1. The DNS example

The objective of this metric is to provide security assurance information about the DNS service, and uncorrupted name resolution. Therefore the infrastructure of concern has been identified as the DNS 9 and the countermeasures to control, the security functions that **checks the address resolution files integrity** and the **DNS user (bind) integrity**.

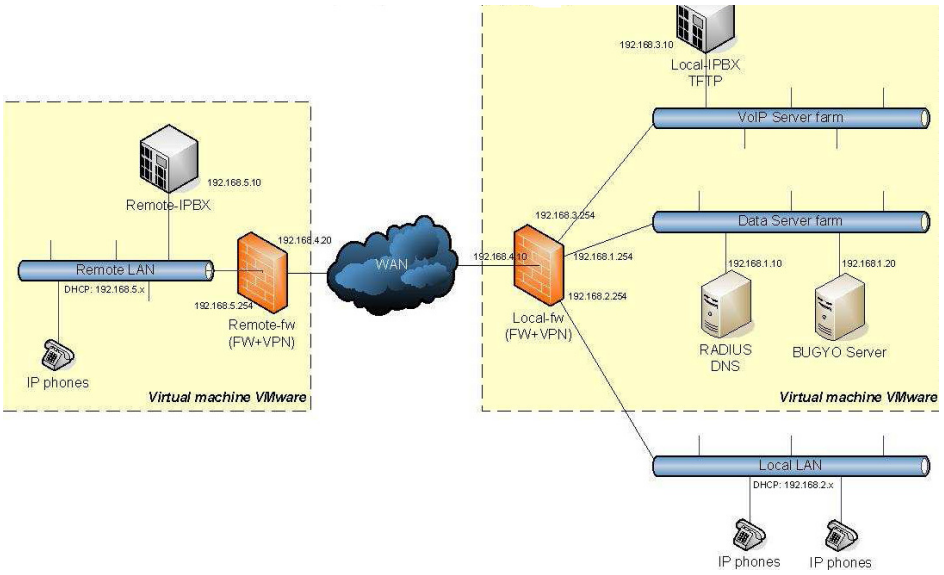

Figure5. VoIP observed system

This metric provides an assurance level for the compliance and non-vulnerability of the controlled counter-measures. A summary of the metric and the associated derived measures are shown in the diagram below.

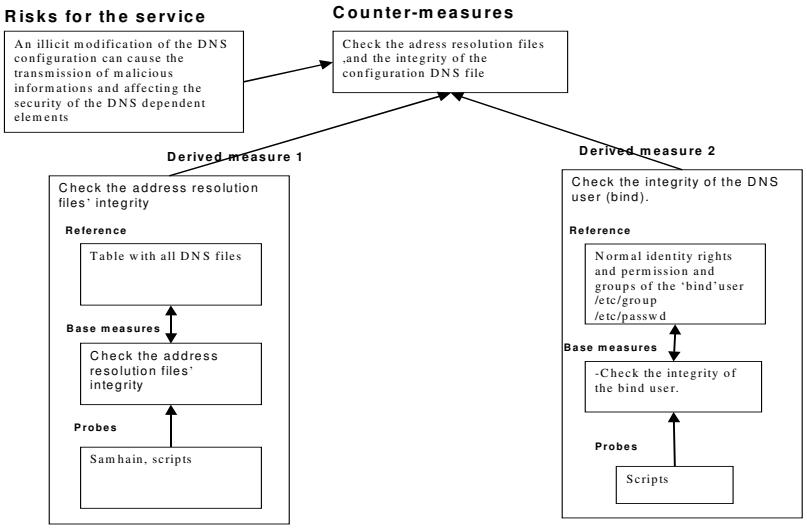

Figure6. Metrics diagram for DNS

For pragmatic reason, we assume that the achievable assurance level is **2**, i.e. *regular informal evidence for relevant parts of the countermeasures* are sufficient to judge on the effectiveness of the security functions.

3.1.1. Derived measure 1

In this test, two probes are used for the measurement of the derived measures:

- The Samhain using cryptographic checksums of files to detect modifications. An effective functioning of that probe will allow the detection of any malicious attack that could lead to the address resolution files integrity being corrupted.
- Scripts checking whether the resolution files are well constructed.

The checking frequency assigned to these two probes is one hour. The table below lists the necessary base measures and references to provide complete security assurance information.

Reference	**Base measures**	**Required probes**	**Frequency**
DNS conversion files integrity	Check the integrity of address resolution file.	Samhain	1 hour
DNS conversion files construction.	Files corrupted: Check if the resolution files content is well constructed.	Scripts	1 hour

Table2. Requirements for DNS metrics measurement (derived measure1)

Taking into account the achievable assurance level provided in 1, three case scenarios are possible:

- If the address resolution files integrity is compromised: corrupted files, errors (configuration errors) then the derived measure value is 0.
- If the address resolution files integrity is compromised: corrupted files, evil-minded modifications then the derived measure is 1.

  In both above cases, an appropriate alarm message is generated to the operator, providing details on the problem.
- Otherwise, the resulting derived measure is 2.

3.1.2. Derived measure 2

For this derived measure, only scripts are used to perform checks on whether the bind user has a correct identity, specific right and permission and if he/she belongs to a specific group. In that respect, specifics checks are made (hourly rate) on the user's identity through file, password checking…and also on whether the user is affiliated to the right group. Table 3 lists the base measures necessary to provide security assurance information.

Similarly to Derived measure 1, three case scenarios are predictable:

- The bind user's identity (/etc/password) is compromised, then the derived measured is 0 and an alarm message is issued to the operator notifying him/her of the risk.
- The bind user's identity is safe, but he/she belong to the wrong group. In that case the derived measure is 1 and an appropriate message is displayed to the operator.
- Otherwise, the derived measure is 2.

Reference	**Base measures**	**Required probes**	**Frequency**
Integrity of bind user: identity.	Check if the bind user has a correct identity, specific right and permission. etc/passwd).	Scripts	1 hour
Integrity of bind user: groups.	Check if the bind user belongs to the wrong groups.	Scripts	1hour

Table3. Requirements for DNS metrics measurement (derived measure2)

3.1.3 Aggregated measure for the DNS metrics

A finale table on both derived measures and the metric values is summarized in table 4.

Derived measures	Metric value		
	0	1	2
Check the address resolution files' integrity	Files corrupted: Errors, misconfigured.	Files corrupted: evil-minded modifications.	OK
Check the DNS user's (bind) integrity	Full integrity unsatisfied	Integrity satisfied but bind user belongs to the wrong groups.	OK

Table4. DNS metric value

Using the minimum recursive function, the metric value of the countermeasures is given by the equation below:

Metric value= Min (derived measure 1, derived measure 2) = Min (address resolution files integrity, checking the DNS user (bind) integrity)

3.2. The FTFP metric example

This metric aims at providing security assurance information about the configuration and the security of Trivial Transfer File Protocol server. Therefore the critical infrastructure is the TFTPD and the security countermeasure to control is the security settings of the TFTP configuration file. The metric indicates an assurance level for the conformity of the controlled countermeasure.

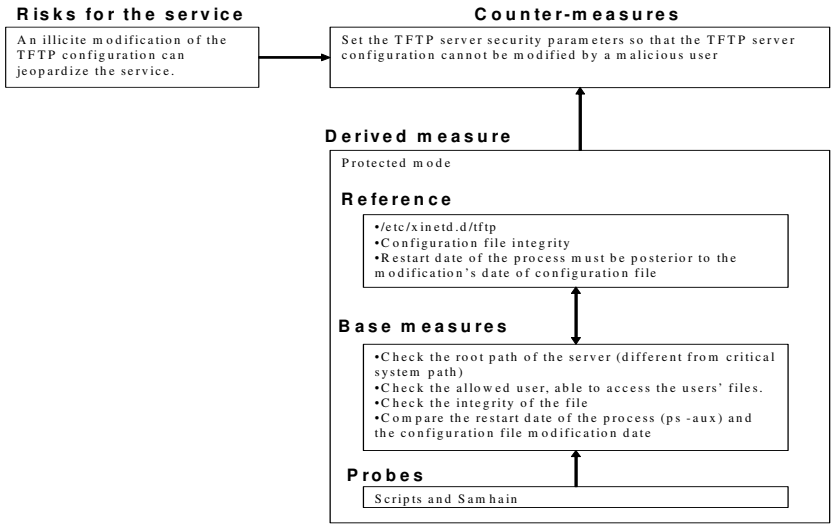

Figure7. Metrics diagram for DNS

Here the achievable assurance level is set at 3 i.e. *frequent informal evidences for important parts of the countermeasure* is enough to appreciate on the effectiveness of the security functions.

Using probes (scripts and Samhain) the following checks are made:

- Check the security settings of the TFTP server configuration file (/etc/xinetd.d/tftp) as shown below

```
service tftp
{
protocol        = udp
port            = 69
socket_type     = dgram
wait            = yes
user            = nobody
server          = /usr/sbin/in.tftpd
server_args     = /tftpboot
disable         = no
}
```

- Check if the server is "chrooted" (root path (server_args) of the hosted files is not a critical system path (e.g. "/", "/root", etc).
- Check if the specified Unix user ("user = nobody") through which the TFTP users are able to access to the TFTP hosted files is uncorrupted or not too powerful (e.g. "root" user).
- Check the integrity of the TFTP server configuration file.
- Check the integration of configuration changes (re-starting date of the process).
  To check if the TFTP server has been restarted after a modification of the configuration file, we need a script to check the last modification date. This date must be anterior to the start date of the process (xinedt) and can be obtained with the "ps –aux" command.

Reference	**Base measures**	**Required probes**	**Frequency**
Server path	"server_args" value	Script	1 hour
TFTP user	"user" value	Script	1 hour
TFTP configuration file integrity	Integrity is compromised	Samhain	5 minutes
Configuration enforcement	Server runs with new/previous configuration	Script	5 minutes

Table5. Requirements for TFTP metric measurement

Based on the interpretation made by the probes and also on the achievable assurance level, four cases are predictable:

- The content of the TFTP file configuration is not satisfactory and in that case the derived measure value is 0.

- The content of the TFTP file configuration is satisfactory but the restart date is anterior to the last modification date, the derived measure value is 1.

- The content of the TFTP file configuration and restart date are satisfactory but the integrity is compromised, the derived measure value is 2.

- All base measures are satisfactory; the derived measure value is 3.

In all the above cases, exception made of the last one, an alarm message is generated to the operator specifying the nature of the security risk.

3.2.1 Aggregated derived measured

Since there is only one derived measure, the Assurance level of the TFTPD is equal to the derived measures "protected mode" as depicted in the table below.

Metric value= protected mode

Derived measures	**Metric value**			
	0	**1**	**2**	**3**
Protected mode	Security parameters are not set.	Security parameters set but restart date is older than last modification date	All BM OK except integrity	All BM OK

Table6. TFTP metric values

4. Conclusion

In this paper, we successfully deployed a security assurance methodology for telecommunication infrastructure, BUGYO, by applying it to a VoIP infrastructure. The specification of metrics has helped determine the effectiveness of the security measures deployed on the infrastructure. From our experience, a good understanding of the infrastructure is imperative for a better specification of the security model which will guarantee a better selection of metrics.

The Bugyo methodology could be used by companies to document, measure and monitor their security assurance and be compliant of governments' requirements such as Sarbanes-Oxley and Basel II.

In terms of perspectives, the extension of the methodology in a way to handle security assurance in a

dynamic environment is being explored along with the use of self-learning agents for the collection, measurement of the assurance levels. Furthermore, if it agreed on the need to know how effective the security countermeasures are in critical infrastructures, integrating automatic reaction measures that allow the system to respond to an imminent failure is also paramount and requires some attention.

Acknowledgement

The results shown in this paper are based on a demonstration carried out at the public research center Henri Tudor in close cooperation with the partners of the Bugyo project which involved Alcatel, EADS, CRP Henri Tudor, Tellindus, Karlstad University, OPPIDA, and ENST...
It is also important to notify that the CRP Henri Tudor was the leading organization with respect to the methodology specification and that the project was supported by the Ministry of culture, higher studies and research (MCESR) in Luxembourg.

References

[1] Bulut E, Khadraoui D, Marquet B (2007), Multi-Agent based security assurance monitoring system for telecommunication infrastructures, In proceedings to the Communication, Network, and Information Security conference, Berkely/California, September 2007.

[2] Celtic BUGYO project, http://projects.celtic-initiative.org/bugyo , march 2008

[3] CC, Common Criteria for information Technology, part 1-3, version 3.1, September 2006.

[4] Vaughn RB, Henning R, Siraj A (2002) Information Assurance Measures and Metrics – State of Practice and Proposed Taxonomy, in *Proceedings of the IEEE/HICSS'03*, Hawaii, Jan. 2002

[5]Arca Systems, Inc. (1995) Using an Assurance Framework to Reason about Security Evaluation Methods, White Paper, December 1995

[6] Zuccato A, Marquet B, Papillon S, Alden M (2002) Service oriented modeling of communication infrastructure for assurance, in *Proceedings of the 2002 IEEE Workshop on Information Assurance and Security*, West Point/NY, June 2002

[7] ISO standard 15939, Software Measurement Process.

[8]Alberts C, Dorofee A (2001) Octave Criteria – Version 2.0, TECHNICAL REPORT, CMU/SEI-2001-TR-016, Software Engineering Institute, 2001

[9] MEHARI (2004), CLUSIF, Version 3, Octobre 2004.
http://www.clusif.asso.fr/

[10]Expression des Besoins et Identification des Objectifs de Securite (EBIOS)(2004), Direction Centrale de la Securite des Systemes d'Information (France), February 2004.
http://www.ssi.gouv.fr/

[11] *CRAMM (CCTA Risk Analysis and Management Method)(2003) User Guide version 5.0*, Insight Consulting – SIEMENS, September 2003.

[12] Vraalsen F, Mahler T, Lund M.S, Hogganvik I, den Braber F and Stølen K(2007) "Assessing Enterprise Risk Level: The CORAS Approach," in *Advances in Enterprise Information Technology Security*, Khadraoui D and Herrmann F, Idea Group Reference, March 2007. ISBN : 978-1599040905

[13] Jürjens J, Secure Systems Development with UML (2005), Springer, Berlin 2005

[14] Mouratidis H, Giorgini P, Manson G, and Philp I, A Natural Extension of Tropos Methodology for Modelling Security (2002), Proceedings of the Agent Oriented Methodologies Workshop (OOPSLA 2002), Seattle-USA, November 2002.