



INTERNATIONAL
HELLENIC
UNIVERSITY

Intrusion Detection for the IaaS Cloud Model

Athanasios Argyropoulos

SID: 3307150001

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Communications and Cybersecurity

DECEMBER 2016

THESSALONIKI – GREECE



INTERNATIONAL
HELLENIC
UNIVERSITY

Intrusion Detection for the IaaS Cloud Model

Athanasios Argyropoulos

SID: 330715001

Supervisor:

Prof. Dimitrios Baltatzis

Supervising Committee

Members:

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Communications and Cybersecurity

DECEMBER 2016

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in Communications and Cybersecurity at the Hellenic International University.

The subject of this dissertation is the study of cloud systems, specifically the Infrastructure as a Service model, the Intrusion Detection Systems and finally propose a complete solution and architecture of Intrusion Detection in the cloud.

This was done by initially analyzing the literature, regarding the cloud, the Intrusion Detection Systems, and the specialized IDSs for the cloud. Then we studied the security problems encountered in the cloud environment and using these data as a basis we went on with our own proposal. At the end, there is a small experimental test of our model.

Here, I would like to thank my supervisor prof. Dimitrios Baltatzis, for introducing me to this subject and technologies through his class, “Intrusion Detection and Event Management” and for his support and the guidance he offered for the completion of this dissertation.

Athanasios Argyropoulos

23 December 2016

Contents

Abstract.....	iii
List of Figures.....	viii
List of Tables.....	x
1 Introduction.....	1
1.1 What is Cloud Computing.....	1
1.2 Cloud Service models.....	1
1.2.1 Infrastructure as a service (IaaS).....	2
1.2.2 Platform as a service (PaaS).....	2
1.2.3 Software as a service (SaaS).....	2
1.3 Cloud Types.....	4
1.3.1 Public Cloud.....	4
1.3.2 Private Cloud.....	4
1.3.3 Hybrid Cloud.....	4
1.3.4 Community Cloud.....	5
1.4 Main characteristics of the cloud computing.....	6
1.4.1 On-demand self-service.....	6
1.4.2 Broad network Access.....	6
1.4.3 Resource pooling.....	7
1.4.4 Rapid elasticity.....	7
1.4.5 Measured Services.....	7
1.5 What is an Intrusion Detection System.....	8
1.5.1 Network-based intrusion detection system (NIDS).....	9
1.5.2 Wireless intrusion detection systems (WIDS).....	9
1.5.3 Network behavior analysis (NBA).....	10
1.5.4 Host-based intrusion detection system (HIDS).....	10
1.6 General known techniques of the Intrusion Detection Systems.....	11
1.6.1 Signature based Detection.....	11
1.6.2 Anomaly Detection.....	11
1.6.3 Artificial Neural Network (ANN) based IDS.....	12
1.6.4 Fuzzy Logic based IDS.....	12
1.6.5 Association Rule based IDS.....	12
1.6.6 Support Vector Machine (SVM) based IDS.....	13

1.6.7	Genetic Algorithm (GA) based IDS	13
1.6.8	Hybrid Techniques	13
2	Problem Definition.....	14
2.1	The OWASP organization.....	14
2.2	OWASP Cloud Top 10 Security Risks	14
2.2.1	Accountability and Data Ownership.....	15
2.2.2	User Identity Federation	16
2.2.3	Regulatory Compliance	17
2.2.4	Business Continuity and Resiliency.....	17
2.2.5	User Privacy and Secondary Usage of Data	17
2.2.6	Service and Data Integration.....	18
2.2.7	Multi Tenancy and Physical Security	18
2.2.8	Incidence Analysis and Forensic Support.....	19
2.2.9	Infrastructure Security	19
2.2.10	Nonproduction Environment Exposure Service-to-User.....	20
2.3	Problems the IDS can solve	20
2.4	Reasoning for the Risks choice and selection	21
2.5	Attack taxonomy in cloud computing	22
2.5.1	Service-to-User	22
2.5.2	User-to-Service	22
2.5.3	Cloud-to-Service	22
2.5.4	Service-to-Cloud	22
2.5.5	Cloud-to-User	23
2.5.6	User-to-Cloud	23
3	Related Work	24
3.1	Categories of IDS/IPS used in Cloud Computing.....	24
3.1.1	Host based Intrusion Detection Systems for Cloud(HIDS).....	24
3.1.2	Network based Intrusion Detection Systems for Cloud (NIDS).....	24
3.1.3	Distributed Intrusion Detection Systems for Cloud (DIDS).....	25
3.1.4	Hypervisor-based Intrusion Detection Systems for Cloud	25
3.1.5	Intrusion Prevention Systems (IPS) for Cloud.....	26
3.1.6	Intrusion Detection and Prevention Systems (IDPS) for Cloud	26
3.2	Methods Proposed	27

3.2.1	Central management approach.....	27
3.2.2	IDS as Hardware	27
3.2.3	Open Source IDS	27
3.2.4	Solutions with combining concepts	28
3.2.5	Solutions with a single technology	29
3.3	What we learned from these Methods in Use	31
4	Proposed Solution	32
4.1	Methodology Used	32
4.2	Multi Tenancy and Physical Security.....	35
4.2.1	Denial of Service (DoS) attacks.....	35
4.2.2	Cloud Malware Injection Attack.....	40
4.2.3	Side Channel Attacks	43
4.2.4	Authentication Attacks.....	45
4.2.5	Man-In-The-Middle Cryptographic Attacks.....	47
4.2.6	Malicious Insiders	49
4.2.7	Scanning other tenants	51
4.2.8	Hypervisor attacks	52
4.3	Incidence Analysis and Forensics	53
4.3.1	Malware detection.....	53
4.3.2	Intrusion detection response	55
4.3.3	Honeypot.....	56
4.3.4	Logging	56
4.4	Infrastructure Security	57
4.4.1	Internet Dependency	57
4.4.2	Active Unused Ports/Port Scanning.....	57
4.5	The Proposed model.....	59
4.6	Overview and explanation of the Proposed model.....	62
4.7	Benefits of the proposed system.....	63
4.8	Work Explanation	64
5	Implementation	65
5.1	Methodology	65
5.2	Technologies Selected.....	66
5.2.1	Security Onion	66

5.2.2	Snort.....	67
5.2.3	OSSEC.....	67
5.2.4	System.....	67
5.3	NIDS Issues.....	68
5.3.1	DDOS.....	69
5.3.2	Injection Attack.....	73
5.3.3	Port Scanning.....	74
5.4	HIDS Issues.....	76
5.4.1	OSSEC.....	76
5.5	Future Work.....	77
5.5.1	Open Nebula.....	77
6	Conclusion.....	81
6.1	What we learned.....	81
6.2	What we proposed.....	81
6.3	Does it work?.....	83
6.4	The Future.....	83
7	References.....	84
8	Appendixes.....	93
8.1	Cisco Signature table for Cloud DDoS attacks.....	93
8.2	Images Used.....	94
8.3	Installations / Use of tools selected.....	95
8.3.1	Security Onion.....	95
8.3.2	Snort Verifications.....	102
8.3.3	SGUIL.....	102
8.3.4	Nmap.....	103
8.3.5	Hyenae.....	105

List of Figures

Figure 1 Cloud Model [3]	1
Figure 2 The three models [4].....	3
Figure 3 Types of Cloud Computing Deployment Models [5].....	6
Figure 4 The Essential Characteristics of Cloud [6]	8
Figure 5 OWASP Cloud Top 10 Risks [27]	15
Figure 6 Overview of the Methodology Used	33
Figure 7 Methodology Flowchart	34
Figure 8 botnet example [29].....	35
Figure 9 Cloud / DoS attacks [31]	36
Figure 10 DDoS attack (a)	38
Figure 11 DDoS attack (b).....	40
Figure 12 Cloud Malware Injection Attack	42
Figure 13 Side Channel Attacks	44
Figure 14 Authentication Attacks	47
Figure 15 Man-In-The-Middle Cryptographic Attacks	48
Figure 16 Percentage of Malicious Insider Incidents [55].....	49
Figure 17 Malicious Insiders	50
Figure 18 Scanning other tenants.....	51
Figure 19 Hypervisor attacks	53
Figure 20 Malware detection	55
Figure 21 Active Unused Ports/Port Scanning	58
Figure 22 Proposed Model.....	62
Figure 23 Testing Methodology.....	66
Figure 24 Snort Example [69].....	68
Figure 25 DDOS attacks statistics [76].....	70
Figure 26 SYN attack [72]	70
Figure 27 Hyena SYN flood	71
Figure 28 Hyena SYN flood (2).....	71
Figure 29 Wireshark SYN [83].....	72
Figure 30 DoS Syn test Workflow	73
Figure 31 Finding the VM address	74
Figure 32 Port scan complete.....	75
Figure 33 OSSEC Alert	76
Figure 34 OpenNebula Cloud running.....	77
Figure 35 Logging in the Cloud (1)	78
Figure 36 Logging in the Cloud (2)	78
Figure 37 Cloud Control Interface	78
Figure 38 Cloud Contents (VMs)	79
Figure 39 Cloud Internal Network.....	80
Figure 40 Proposed Cloud IDS Architecture	82

Figure 41 Cisco Signature Track	93
Figure 42 Raw Cloud Architecture image	94
Figure 43 Security Onion in Virtual Box.....	95
Figure 44 Security Onion Interface.....	95
Figure 45 Network interface selection	96
Figure 46 Sensor installation	96
Figure 47 Sniffing interface	97
Figure 48 Production Mode	97
Figure 49 Standalone selection	98
Figure 50 Snort Selection.....	98
Figure 51 Ruleset Selection	99
Figure 52 IDS enabled	99
Figure 53 Full packet capture	100
Figure 54 Security Onion Configured.....	101
Figure 55 First actions	101
Figure 56 Snort Rules default databases	102
Figure 57 Sguil and OSSEC	102
Figure 58 Snort and OSSEC in Sguil.....	103
Figure 59 Nmap setup	103
Figure 60 Npcap setup	104
Figure 61 Nmap Interface	104
Figure 62 Nmap Map.....	105
Figure 63 Hyenae.....	105

List of Tables

Table 1 Differences between NIDS and HIDS [84]	34
Table 2 DDoS attack (a) [21].....	37
Table 3 DDoS attack (b)	40
Table 4 Cloud Malware Injection Attack.....	42
Table 5 Side Channel Attacks.....	44
Table 6 Authentication Attacks	46
Table 7 Man-In-The-Middle Cryptographic Attacks.....	48
Table 8 Malicious Insiders.....	50
Table 9 Scanning other tenants	51
Table 10 Hypervisor attacks	52
Table 11 Malware detection.....	54
Table 12 Active Unused Ports/Port Scanning.....	58
Table 13 Full Table Combination	59
Table 14 Simplified Proposal table.....	60
Table 15 Final Proposed System.....	61

1 Introduction

1.1 What is Cloud Computing

The cloud is defined as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [1] by the National Institute of Standards and Technology (NIST).

To understand the use of this technology and its global acceptance we refer to the financial figures it represents. According to Forbes just one provider, Amazon Web Services (AWS), in 2015 generated \$7.88B in revenue, while in 2016, spending on public cloud Infrastructure as a Service hardware and software is forecast to reach \$38B, growing to \$173B in 2026. [2]

1.2 Cloud Service models

Cloud computing as a meaning and technology can be broken down to three different models. They are deviations of the same holistic model, broken down to three types, as resources and abilities are cut off from the user, as he gains other services to compensate this loss. The lowest and more raw model is the Infrastructure-as-a-service (IaaS) followed by the Platform-as-a-Service (PaaS) and on top lies the Software-as-a-Service (SaaS)

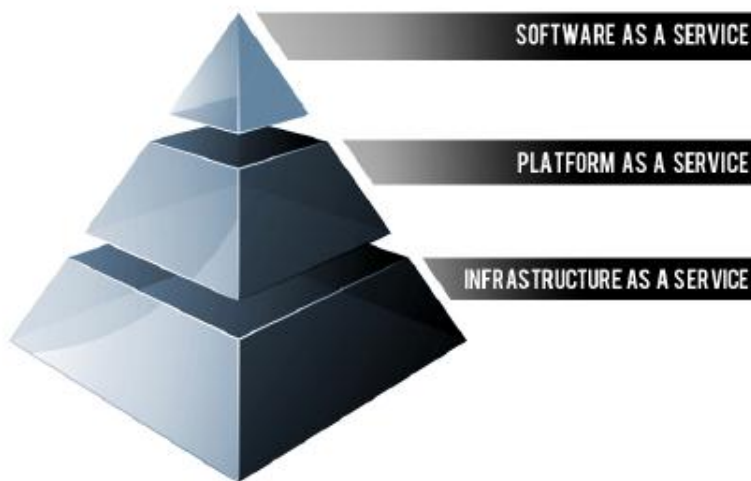


Figure 1 Cloud Model [3]

1.2.1 Infrastructure as a service (IaaS)

Infrastructure as a service is defined as the creation of virtual hardware resources including virtual machines, virtual networks and virtualized storage. The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. [1]

By the definition, we understand that we talk about the lowest level. The whole model regards mainly hardware and resources associated directly with that. The consumer, the cloud tenant is provided networks, processing power, storage and networks. In general, the IT infrastructure is provided as a service and not as a dedicated capability. Examples of this model are Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE), Cisco Metapod.

1.2.2 Platform as a service (PaaS)

Platform-as-a-Service (PaaS) is a Platform tailored environment meant to accommodate the development of applications. The capability provided to the consumer is to deploy onto the cloud infrastructure, consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. [1]

The middle allows the consumer to use the infrastructure with his own software means. Meanwhile he can't manage or directly take control of the IT infrastructure as he could in the previous model, but he can have control over the deployed applications, storage and even the operating system, which essentially provides cloud components to the software. Some constraints are present, as which applications can be deployed, certain limitations may exist and this can make specific applications undeployable, while not the operating system nor the networks are manageable, though it can be regarded as a programming environment with given libraries and an execution environment.

Such popular Platforms are AWS Elastic Beanstalk, Heroku, Force.com, Google App Engine, Apache Stratos.

1.2.3 Software as a service (SaaS)

Applications are offered to end users through a web browser or some other kind of thin client and are almost entirely stored managed and updated in the cloud. This model is termed as Software-as-a-Service. It is essentially the capability provided to the consumer, to use the provider's applications, running on a cloud infrastructure [1]

Here the clients are provided with their own access to application-software which are called the on-demand software. The provider is responsible for the installation, and the user does not interfere with the setup and running procedures of the application. By paying his agreed fees, the user is able to use this software to match his needs, and

of course his data are stored within. The user has the less freedom here, he can't directly interact and use the hardware, he does not deploy his own software but he can only use the preinstalled ones, so the choice to match his need must be even more strict and well thought here. E-mail, financial management and customer service are some services that generally have taken their place prominently in this cloud model. Examples of such platform are Google Apps, Microsoft Office 365 and e-mail hosts.

In the schema below we can see the restrictions mentioned for the three models, as well as which areas are managed by whom. The IaaS gives more responsibilities but more freedom and it is a great choice for a new company which can't yet afford its own setup or if the company is growing rapidly and needs the extra resources as soon as possible. PaaS is useful, where a group of developers are working in a project, which is collaborative and many parties have to alter it, and interact with the project and each other. The popularity of agile software development also favors the choice of this model because of its attributes and the fact that it aids rapid software development. SaaS supports use of certain applications because it provides the needed services. We have countless implementations, and the majority of the web services fall into the SaaS model from Google Search to Facebook and Gmail.

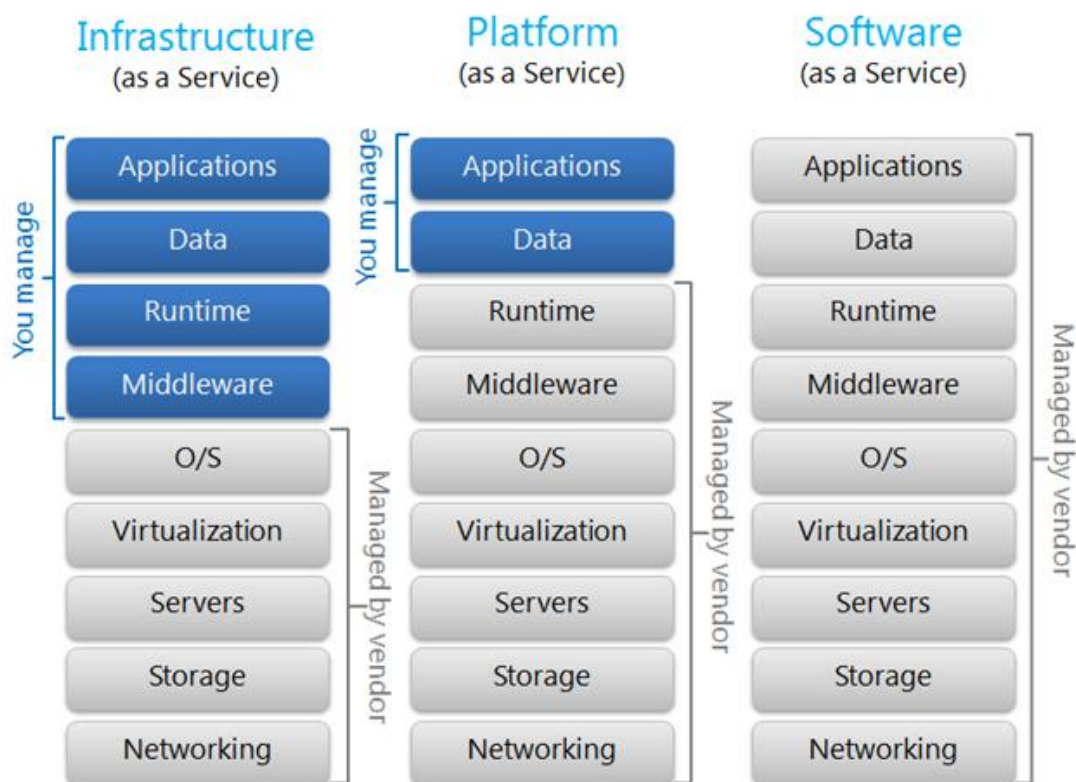


Figure 2 The three models [4]

1.3 Cloud Types

Another differentiation is the cloud type, with four distinctive cases existing. This categorization falls into end user classification, and his affiliation with the provider, meaning, for example, that it is different if the user is a leased host/tenant or the infrastructure owner.

1.3.1 Public Cloud

This cloud infrastructure's goal is open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It operates under the responsibility of the cloud provider. [1]

A public cloud is basically, the internet as a user can understand it, as a simple single user can perceive it, it has the same mechanics and dynamics, so, in an abstract way, we can say that a single user can see many similarities from his point of view. He does not have any authority and control over the location of the infrastructure. He indirectly interacts with it and only through internet, as direct access is not available but this way it is open for public use. Technically, as architecture, there are no differences between the Public and the Private cloud, except when it comes to the legitimate user. The use of it can be either free, or it can be a paid service. An example of such a cloud is Google.

1.3.2 Private Cloud

The cloud infrastructure is offered for exclusive use by a single organization including multiple consumers (e.g. business units). It may be owned, managed, and operated by the organization, a third party, or some combination of it, and it may exist on or off premises. [1]

The private cloud sits in the inner layers and departments of companies. It permits only the authorized users to enter it, and so in general users have many permissions to alter data, or use it by their means and needs. It is used either for inner management, for the use of services in a more local and, or organized manner. The resources, meaning purely the hardware infrastructure, can be located either internally or externally to the company's premises, as this makes no difference for the model. However often security issues or the occurrence of natural disasters, make this model fail to guarantee continuity if implemented in the incorrect way.

1.3.3 Hybrid Cloud

The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) which remain unique entities, but are bound together with standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds). [1]

Having analyzed the two previous types, and by having combinations of them, with different implementations and architectures, leads to Hybrid clouds. It is like a company that uses the public cloud for its normal operations, but also have in the background, it it's networks' backbone a private one, for more sensitive or private issues, which is connected with the public for various operations. So, for example an organization can store data in its private cloud, but it uses these data in a public cloud for operations. This, results to a new model, the Hybrid model. The private and the public models are integrated, but yet they remain distinct entities. Private clouds can be expensive so they are usually not a viable or realistic option for the average business, while issues with security and legislation apply, which makes it even more hard to support and operate.

1.3.4 Community Cloud

The cloud infrastructure is provided for exclusive use by a specific community of consumers, a group of from organizations that have shared goals (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, or a third party, or some combination of them, and it may exist on or off premises, meaning the physical hardware can be or not in a remote location. [1]

The definition leaves quite a gap here as which organizations can be considered as close enough to share a cloud, or how access is defined. It clearly states that there can be different ownership states for Community clouds. Organizations that have understood the potential of cloud hosting, they share common attributes and goals. As for example banks, they could also share the same computing needs and in order to achieve their business-related objectives, they can team up and create a community cloud.

This cloud model can be also managed by a third-party entity, and not necessarily by a community member. It can be hosted externally or internally, and it can be private, public or hybrid while the organizations benefit from the shared cost, know-how, or even computational power and resources. Imagine account transactions between different banks, being serviced within the same cloud eco-system, rather than having data traversing through the net, or even universities sharing resources and platforms for research.

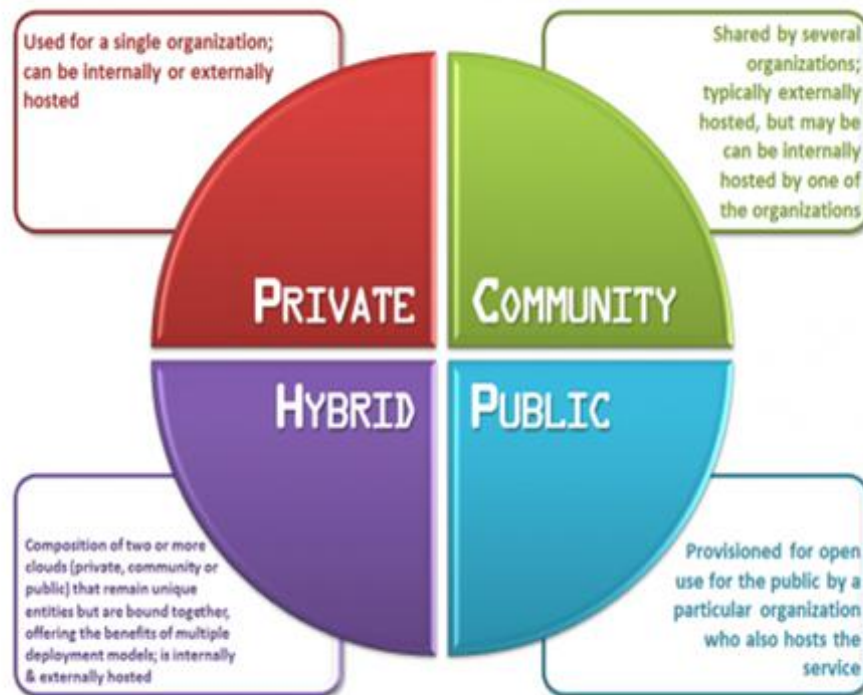


Figure 3 Types of Cloud Computing Deployment Models [5]

1.4 Main characteristics of the cloud computing

After the organizational aspects of the cloud, we come to see the characteristics that define such infrastructures and setups as cloud computing ones.

1.4.1 On-demand self-service

A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider. [1]

This essentially implies to have the rights to change cloud services through an online control panel or by interacting with the provider. This also users changing software, managing resources and altering networks (depending on the model used). These terms and options vary from vendor to vendor, but it is common to have available a variety of them.

1.4.2 Broad network Access

Having the obvious need for a considerable bandwidth from the cloud service, to provide adequate access for the users, the cloud infrastructure should manage laptops, smartphones, tablets and of course tabletop workstations. The mobility is a necessity in modern technology, according to users' needs, and is a common request from employees to log into their business accounts for remote working from home. This

doesn't exclude the private clouds, because them being private doesn't mean they don't need to provide the same services, even regarding the mobile and remote access we analyzed here.

1.4.3 Resource pooling

The provider's computing resources are pooled together en masse to serve multiple consumers using a multi-tenant model, with different physical and virtual resources. They are dynamically assigned and reassigned according to the incoming consumer demand. There is a strong sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Some examples of the resources considered pooled include storage, processing, memory, and network bandwidth. [1]

This model, clarifies, the ability and the need of the user to alter data from any location, at any given time, while having his own private sector, special place, as a tenant of the system. Though these resources are spread, perhaps in even different physical data centers all over the world the end user has no knowledge or mere understanding of this fact, and can only perceive the cloud system as a whole, and not separated. Even if the whole system is in the same place, it's fewer parts, as networks VMs etc. are still unperceivable by the user.

1.4.4 Rapid elasticity

Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time. [1]

The cloud is flexible to match the users' needs, so it enables each user to get allocated, more space, computational power according to each one's needs. As an example, a mail provider gives 1GigaByte storage to all his users. This is static, and for a user that requires less, it doesn't mean that the system will allocate less to him even if he only uses 10 Megabyte. But an option can exist that when a user gets close to the mentioned limit he will get another Gigabyte, adding up to 2 Gigabytes reserved space. This option is the key meaning of rapid elasticity. So, resources and features can be added or subtracted to offer better service and experience.

1.4.5 Measured Services

Cloud systems automatically control and optimize resource use, by leveraging a metering capability¹ at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service. [1]

Cloud can have a more affordable nature by paying only the true usage, even in form of micro-transactions. Since not all users have the same needs, different billing packets can be deployed, and each customer can make his own choice of program, or have a pay-as-you-use policy and getting charged for the made usage. This is possible because the services of the cloud are measurable, in terms of storage, bandwidth, processing etc. Normally the usage can be monitored transparently by both the provider and the user for the user to have better use of the system under his budget.

5 Essential Characteristics



Figure 4 The Essential Characteristics of Cloud [6]

1.5 What is an Intrusion Detection System

Intrusion detection is the process of an examining the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or about to happen threats of violation of computer security policies, acceptable use policies, or standard security practices [7], while an intrusion detection system (IDS) is software that automates the intrusion detection process. [8]

So, an Intrusion Detection System (IDS) is essentially the software used to scan and analyze packets and behaviors to conclude, whether an intrusion incident or other

violation is happening to a computer system, which may include but not limited to hijacking, direct intrusions, malware, or violation of policies.

A common mistake is to misinterpret the IDS as a firewall, but they are two completely different things. This mistake based on the misconception that they are both network devices, and the installations are between the network and the host, which is not completely true as explained later. Several types of Intrusion Detection Systems exist.

1.5.1 Network-based intrusion detection system (NIDS)

A network-based ID system monitors the traffic in its network segment or devices as a data source to identify suspicious activity, or even the whole network, and analyzes protocol activity. Mostly they are deployed at the edges between networks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. Two types of these systems exist the On-line NIDS which deals with the network in real time, as it analyses the packets and applies rules to decide and distinguish the packets. To examine if they are ill-intended or not and the Off-line NIDS which deal with stored data and passes them through some processes to decide if it is an attack or not. [7]

Commercial solutions for this type include AXENT (Symantec) [10], Cisco Catalyst [11], CyberSafe [12] and ISS [13]

1.5.2 Wireless intrusion detection systems (WIDS)

In general, a WIDS monitors a wireless network for suspicious traffic by analyzing wireless networking protocols. [9] The typical wired Intrusion Detection System cannot do much in a wireless environment, and thus there was need for these wireless detection systems.

A wireless Intrusion Detection system performs this task exclusively for the wireless network. In their core, they have the same logic as their wired counterparts, regarding packet analyzing, but they are more difficult implemented, because of the wireless networks nature, which have to take in consideration many attributes as range, signal strength. It is also common for the wireless lan's range to be bigger than that of the company's, in area coverage which adds even more difficulties.

The main components of typical wireless network are the Station which is a wireless endpoint device such a smartphone, tablet or laptop, and the Access Point which logically connects the stations with a distribution system. The WIDS is implemented either as a sensor (and deployed as a Station, static or even mobile) or can be integrated within the Access Point. [15]

1.5.3 Network behavior analysis (NBA)

These systems examine network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDoS) attacks, certain forms of malware, and policy violations (e.g., a client system providing network services to other systems). These systems usually have sensors and consoles, while more sophisticated ones offer also management servers (commonly referred as analyzers). Usually these sensors act as the ones NIDS uses, but some do not monitor the networks directly, but mainly rely on the network flow information provided by routers and other networking devices, and from this data the sensors reconstruct a series of observed events to determine the origin of a threat. Still they rely on signatures, and so they are rather slow on picking up new threats. [16]

To conclude the NBA is close to the NIDS with the differentiation that, instead of looking for protocol violations and rules regarding the network packets, it focuses in a more general way in the network and its behavior as a whole entity. Their similarities explain the use of same sensors.

1.5.4 Host-based intrusion detection system (HIDS)

A host-based intrusion detection system (HIDS) is an intrusion detection system that monitors and analyzes the internals of a computing system as well as (in some cases) the network packets on its network interfaces (just like a network-based intrusion detection system (NIDS) would do). [17] This was the first type of intrusion detection software that have been designed, with the original target system being the mainframe computer where outside interaction was infrequent. [18]

As a concept, they monitor the behavior of applications on the host by observing the interaction of those applications with the underlying operating system. In practice, security-relevant interactions typically take the form of system calls, and so their scheme works by examining the trace of system calls performed by each application. [19]

The differentiation is that it is installed software on a single host and it monitors for suspicious activity by analyzing events and system calls in that particular host. So, while the first three types of IDS had a network aspect and more holistic approach in the detection, here we get involved with just one host, and the events occurring within that host.

1.6 General known techniques of the Intrusion Detection Systems

Different techniques could be used to enable the detection the intrusions to a system. They are implemented, as procedural methods within the system to allow it to work without interferences while deciding about the existence or not of a threat.

1.6.1 Signature based Detection

Signature-based IDS refers to the detection of attacks by looking into specific patterns, which are found inside the transmitted network packets. The name was borrowed from malware detection, as they use the same methodology.

A database of known signatures is installed. Each passing packet is sniffed and inspected. If it matches the patterns of the database (the signatures), which is more commonly data sequences, data-bits streams, the system is alerted. With this knowledge base the system is well protected against all known attacks, but can't react in newer ones, the ones the database lacks. It is impossible to detect new attacks, for which no signatures are available [20] [21]

The success of these systems is based on the extent and variety of the signature database, on how often is updated, how agile, and in case of an experienced attacker how safe and untampered are designed. Because of the need for every packet to be analyzed, this creates quite a big overload for the CPUs as it takes a lot of computational power. [20] [21]

1.6.2 Anomaly Detection

Anomaly-based IDS were introduced to detect unknown attacks in contrast with the Signature-based IDS. The basic approach is to use machine learning, to train the IDS what is the systems' normal behavior, and then it will understand by itself, when something wrong is happening, as it will show different behavior and attributes. So, it detects computer intrusions and misuse, by monitoring system activity and classifying it as either normal or anomalous. [20]

This method can find new attacks, but it's main problem is the number of false positives, because legitimate, normal activity can be taken as malicious, while it isn't, just because it has differences with the normal behavior the IDS got trained and used to. Example: a user logs on and off 20 times a day while the normal behavior is 4-5 times. This change might alert the system. The basic approach is to use machine learning to create a model of trustworthy activity, and then compare new behavior against this model. [20]. The key element for using this approach efficiently is to generate rules in such a way that it can lower the false alarm rate for unknown as well as known attacks. [21]. Many techniques have been used to implement this such as hidden Markov models, statistical modeling and data mining.

1.6.3 Artificial Neural Network (ANN) based IDS

An ANN is an information processing system that is inspired by the biological nervous systems, such as the brain process information. It is made out of many highly-interconnected processing elements which are called neurons (like in the nervous system) and they work together to solve specific problems. Each of these is a summing element followed by an activation function. The output of each neuron (after applying the weight parameter associated with the connection) is fed as input to all of the neurons in the next layer. [21]

It shares the benefits and the drawbacks of the Anomaly detection, as it also has training process and can identify new unknown attacks, but it is considered to be more sophisticated. The false positives still exist here.

The learning process is making the neural network better with every stage, it optimizes it and consists of the following basic steps.

- Present the neural network with a number of inputs (each vector representing a pattern)
- Check how closely the actual output generated for a specific input matches the desired output.
- Change the neural network parameters (weights) to better approximate the outputs. [21]

1.6.4 Fuzzy Logic based IDS

Fuzzy logic can be used to deal with inexact description of intrusions. It provides some flexibility to the uncertain problem of intrusion detection. Tillapart proposed Fuzzy IDS (FIDS) for network intrusions like SYN and UDP floods, Ping of Death, E-mail Bomb, FTP/Telnet password guessing and port scanning [26].

A combination of the fuzzy logic and the neural networks was introduced to reduce the large time needed to train the ANN, and it is done by closely supervising the training process. It is called Evolving fuzzy neural network (EFuNN). It shows better accuracy than when using only ANN. It is mainly used for large scale attacks such as, DoS/DDoS but it can be used for any unknown attacks (especially in the Cloud) [21]

1.6.5 Association Rule based IDS

Association rules mining started as a technique for finding interesting rules from transactional databases. In the IDS an attempt is being made to correlate the attributes of the network data. Essentially with association rules we mine to find the correlation between these attributes. Modified association rule mining is used to generate the attack rules from the network data. Algorithms are being applied to effectively built the item sets from the training set. The rules are then built from the item sets and further tested on the data set. [33]

1.6.6 Support Vector Machine (SVM) based IDS

SVMs are used to detect intrusions based on limited sample data, where the amount of the input or the dimensions of data will not greatly affect the accuracy. The false positive rate of the SVM is considered to be better than that of the ANN, even though the ANN requires a larger training period. SVM has less parameters and is used only for binary data. So, it is usually combined and works with other techniques for improved results. In Cloud, if limited sample data are given for detecting intrusions then use of SVM is an efficient solution than ANN, since dimensions of data are not affecting accuracy of SVM based IDS. [21]

1.6.7 Genetic Algorithm (GA) based IDS

A genetic algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection [59]

In IDS, these algorithms are used to determine parameters, or select the proper features of the network to help to improve the accuracy of the IDS. We won't commonly find them alone supporting the IDS but in combinations with other techniques, and is used to improve the procedures and the accuracy. [21]

1.6.8 Hybrid Techniques

Hybrid techniques are those which by default use the combination of two or more of the above techniques. It is a common practice, as this way we can have cover the weak points of a technique with the introduction of another one.

2 Problem Definition

2.1 The OWASP organization

The Open Web Application Security Project (OWASP) is an online community, established as a not-for-profit charitable organization, which creates freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security. [22] It has raised to dominance and estimation of the community through the individual contribution of many members, and as there is no existence of an international committee or organization to take similar surveys around these cybersecurity issues, it has become the status quo and point of reference for both scientists, vendors, and authorities.

It is an open community that promotes transparency from their code to their finances, and encourages innovation, for experiments to be made to provide solutions to security issues, in the global scale that is now accepted and even revered, and these ethics are to guarantee the integrity of the organization, as being true and honest towards the community [23]

Annually reports are being published regarding various topics in a form of a top-10, with issues such as “OWASP Top 10 Vulnerabilities”, “OWASP Top 10: Application Security Risks”. The OWASP Top 10 is free to use. It is licensed under the Creative Commons Licenses.

2.2 OWASP Cloud Top 10 Security Risks

‘OWASP Cloud Top 10 Security Risks’ is another top 10 from OWASP which we studied to understand the problems, vulnerabilities and risks of cloud computing and estimate which of them can be solved by the intrusion detection systems. It can be found at
“https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project”.

These ten categories are presented here with some key explanations about them, what they regard, and whether an IDS can provide a solution, or other methods should be considered and used.

Cloud Top 10 Risks

- **R1: Accountability & Data Risk**
- **R2: User Identity Federation**
- **R3: Regulatory Compliance**
- **R4: Business Continuity & Resiliency**
- **R5: User Privacy & Secondary Usage of Data**
- **R6: Service & Data Integration**
- **R7: Multi-tenancy & Physical Security**
- **R8: Incidence Analysis & Forensics**
- **R9: Infrastructure Security**
- **R10: Non-production Environment Exposure**

Cisco Public

9

Figure 5 OWASP Cloud Top 10 Risks [27]

The list is used in the paper about the “Top 10 Cloud Risks That Will Keep You Awake at Night” by Babu Chebrolu, Vinay Bansal and Pankaj Telang, courtesy of Cisco. With Cisco being arguably the biggest retail seller of network infrastructure in the world, we can understand the significance, of the OWASP lists, and how both the academia and the industry take them under consideration.

2.2.1 Accountability and Data Ownership

The knowledge center of a company is beneath the complete management of that organization. The organization coherently and physically protects the information it owns and possesses. An organization, that chooses to use, presumably, a public cloud for hosting its business, loses management of its knowledge center. This poses important security risks that the organization must deliberately consider moderating and mitigating. There must be a guarantee regarding the assurance of the recuperating Data and Information. When this information is entrusted to a third-party operator, they must also be checked for the security standards set by the cloud administrator. [24]

The key points here include the issues of Data Sensitivity and Ownership, and has to do with the management of access to info or data and the control of access to these, that may bring loss of leverage or level of security if revealed to other parties. [25], Data storage Location matters, as whether it is protected properly, or even for the regional reason, if they are physically in a different country, they might abide to a different set of laws.

A present Method of Data encryption and/or the Presence of Multiple Encryption Keys will certainly aid to the protection of these data, as they will not be comprehended from those who are not supposed to. The Logical isolation of the data of multiple consumers and the Proper deletion of the no longer wanted data, will provide coverage from inside attackers, due to the neighboring physical existence of the data.

There is a trust issue, towards the cloud administrator and operators, as he might have unlimited access to the stored information, and guarantees must be given, that this will not occur. Back-ups are needed for safekeeping, but also their use and storage must be handled with care, as they can be a point of an information leak.

We see that these problems, severe as they are, do not fall into the categories mentioned in the introduction, as where the IDS can be used. Data leakage for example can be avoided and detected not directly, (as it is still data traversing) but in regards of the method which is being used. But as these methods fall in another OWASP category we will analyze them later.

2.2.2 User Identity Federation

It is vital for the enterprises to stay upon the management of the user identities as they move their services and applications to the various cloud suppliers. As opposed to letting cloud suppliers produce multiple identities for the same entity (which can occur, for example by creating different accounts (considered here as a different “identity” to make the example more obvious) for logging in the cloud service, whilst an account for the service is already existent. Then the same “entity” the user, will have two “identities”, one to log him in the service, and another to log him in the cloud) which will become much harder to manage, making it nearly impossible to oversee them. Users, clients must be unambiguously distinctive, identifiable with a unified authentication / validation that works over the cloud suppliers. Client experience is upgraded when he/she doesn't deal with various user-ids and credentials. This permits less demanding back-end information reconciliations between the two parties. [24]

This is the theory around managing Identities across multiple providers, where as easy and benefiting it is for the user, security issues do arise. However, this falls to the privacy theory for cybersecurity, and its theoretical part, while IDS deal with more technical aspects and can't provide adequate solutions, as this can be handled by encrypted communications to and from the cloud and secured access lists.

2.2.3 Regulatory Compliance

Things that are perceived to be secure in one country may not be so in another one, due different regulatory laws across countries or even regions. Not having a globalized legislation and ruleset leads to this problem, and the lack of straightforwardness in these technologies and their implementations make it very hard to even prove their compliance to these rules. [24]

As the cloud and the IOT information systems run in a globalized environment, the lack of these general international standards, create problems for regulatory compliance, as the data flows and crosses multiple borders. However, this is not the IDS have been designed for, as it is a purely regulatory issue.

2.2.4 Business Continuity and Resiliency

Business Continuity is a practice that an IT association uses to guarantee that the business can still be led and go on in the case of a disastrous circumstance. In the case of a company that uses cloud, the responsibility of business continuity gets delegated to the cloud supplier, as the supplier is designated for that. This creates a risk to the organization of not having applicable business continuity as the business coherence falls to the provider. Regarding Service Continuity and Quality of Service, written agreement guarantee must be provided and both parties must have a Service Level Agreement [24]

The issue of the knowledge, the know-how, technology and capabilities needed to ensure continuity and resiliency exists, and it is important as large Monetary losses can be sustained due to an outage. There is also the rare chance, but still possible, of the Cloud provider to get acquired, bought out, by a users' competitor who will have direct access to his opponents' data.

To solve this there is no need for an IDS but rather a Service Level Agreement between the two parties, which may include monetary penalty for the downtime it can define expected recovery time, and the provider should get certified to a business continuity standard such as BS 25999.

2.2.5 User Privacy and Secondary Usage of Data

Client's own information get in the cloud as clients begin utilizing social sites. The majority of the social sites are obscure about how they handle clients' individual information. Moreover, a large portion of the social sites run with the default "share all" options, (least restrictive) setup for the client, to enable the platforms to have even more data available. There is the need to get a guarantee from the cloud suppliers, regarding what information can or can't be utilized by them for optional purposes, as they can be sold for e.g. coordinated publicizing of adverts. [24]

The issues dealing here are the Privacy of users' data. The users' personal data can be mined or used and even sold without his given consent. Also as a direct link to the previous chapter regarding jurisdictional borders, the differences in legislation also have an effect in this category, as the private data, can be treated differently in different countries.

The user must always have the control over his data and the option to delete them, so more control has to be given to him, to decide for the proper selection of data usage, sharing and protection, of his own personal data and information.

Solutions for this can again be the encrypted storage of the said data, and the De-identification of the personal Information, but still it's mainly a policy issue, regarding the Privacy and Acceptable Usage, and the Secondary Usage, and not something an IDS copes with.

2.2.6 Service and Data Integration

As the data traverse through the internet, there can be interceptions, before the end users reach the cloud data centers. The unsecured packets can be compromised and this creates problems. Every organization should be of concern for the interception of data but it gets worst in the cloud environment as the data are transmitted over the Internet.

In order to be safe, the data in both ends should be encrypted, while stored at rest, and while travelling in transit. This can be done with the use of various Encryption protocols and keys.

2.2.7 Multi Tenancy and Physical Security

Multi-tenancy in Cloud means sharing of resources and services among multiple clients (networking, storage/databases, application stack). This increases dependence on logical segregation and not so much physical, as users get their own logical separated part and not their own physical drives and independent hardware. There must be a way to ensure that tenants won't tamper with this drawback, and by doing so, the confidentiality and security of other users and tenants. [24]

These, probably inadequate, logical separations can lead to several problems, especially though malicious or ignorant tenants who can take advantage of this co-mingled tenant data. This hides many performance risks as we can have side channel attacks and users scanning other tenants. In the security aspect, even the Denial of Service attacks fall into this category.

To solve these, many things can happen and help, such as a Virtual Private Cloud, to be separated more from the other users, and having better architecture for these multi-tenancy situations. The IDS can also help, as they can trigger on side-channel attacks and DoS attacks and several other physical security issues. Plus, controlled and

coordinated change and access management is needed, and as always data encryption can solve many drawbacks.

In order to find the associated vulnerabilities, we cross-referenced the source material, “the OWASP top-10 Cloud security risks”, with other OWASP lists such as the “OWASP top-10 Application Security Risks”.

Of course, it is impossible to find and identify them all in a dissertation, but by using these lists we can locate at least the top-10 categories, and continue with our work, knowing that we have covered a vast field. In this research, we found the vulnerabilities being associated to be Denial of Service (DoS) attacks, the Cloud Malware Injection Attack, the Side Channel Attacks, Authentication Attacks, Man-In-The-Middle Cryptographic Attacks, Malicious Insiders, the issue of Scanning other tenants and the Hypervisor attacks.

2.2.8 Incidence Analysis and Forensic Support

In the event of a security incident, some reverse engineering and back-tracking, in order to understand what happened, is needed. This analysis can help to detect the malware which can still be resident in the system. Also, the analysis may be needed for a more immediate response, in order to mitigate the impact.

Still, having different regulation through the countries, the authorities designated to act in the forensics field, need to have some help and mutual understanding with the cloud provider, in case their contribution is needed after a security breach event. The checking of the logs is one of the difficult parts, as the points of interest may be distributed in many hosts and machines, and even physical data centers, which may be subject to different laws. Forensic recovery is even more difficult due to the physical multi-tenancy of the storage means, as it may affect continuity, or disrupt, corrupt and essentially alter the data.

To solve this (apart from abiding to some global standards as mentioned earlier), dedicated forensic Virtual Machine images can be present, and the use of a form of honeypots can also be useful. Comprehensive logging is essential, in order to keep track and work backwards to understand certain events, while of course keeping in mind the necessity of not compromising the performance of the service.

Apart from logging, again with cross-referencing to other OWASP projects, and other sources, we find Malware detection, Intrusion detection response and the use of Honeypots to be relevant to our subject.

2.2.9 Infrastructure Security

All infrastructure, as a framework, should be hardened to become more solid and must be designed and arranged firmly, while the setup configuration baselines must be supportive towards the business best practices. Applications, work frameworks, systems and networks should be architected and designed by creating tiers and

security zones, including inherent countermeasures such as demilitarized safe zones (DMZs), to provide a safeguard, and to get access to these, must be arranged and designed to be done by only the needed network and application protocols and services.

As always policies and methods for the user access must be present, and evaluations, auditing and risk assessments must be done by an autonomous associate group or organization. [24]

The vulnerabilities of this section include the active unused ports which can be scanned and exploited, the use of default passwords and even default configurations on the systems, or on hardware used by the systems. Apart of the solutions mentioned, the installment of an IDS is necessary for this category.

2.2.10 Nonproduction Environment Exposure Service-to-User

There is a higher probability of an unauthorized user getting access to the non-production environment, as typically standard authentication is in use. This leads to severe Security flaws, and even data leakage is possible, as the files can be copied. So, there is a need for authentication, or even better a need for layers of authentication, with some user concern. For example, if not extremally needed, perhaps some sensitive data, or the development of a highly sensitive application shouldn't be stored or done in the cloud environment.

2.3 Problems the IDS can solve

Having studied in the first chapter the use, abilities and capabilities of the Intrusion Detection Systems, we have a clear view of what these systems can and what they cannot do. While a useful tool, there is a necessity for them to exist and so to create a safe harbor online environment, especially in something as large, complex and visible, as a cloud environment, which has different needs than a home computer connected to internet where a firewall would suffice.

They provide safeguarding, in both external and internal attackers, but they lay more in the packets use, the processes called to operate, and the way they operate. They cannot make policies, only monitor certain aspects of them. They cannot be globalized, or play the role of other programs, such as cryptographic tools, firewalls, access lists and they don't give privileges to users.

Useful as they are, they are not the panacea of on-line security attacks, risks and vulnerabilities, everything has its own countermeasure or countermeasures, and the IDS focus and are able to respond on certain things such as packet inspection. These limitations, which every technology and countermeasure of course has, makes the IDS to be useful to some of the categories mentioned earlier, while irrelevant to the rest.

2.4 Reasoning for the Risks choice and selection

We are the position now of knowing the capabilities of the intrusion detection systems, and having analyzed the risks the cloud environment faces according to OWASP.

For each risk, there was a judgment made, through analytical thinking and common reasoning, it comes obvious whether the presence of an IDS would help to mitigate this risk.

To clarify, to make crystal clear, how the selection of the Risk Categories was made, we can refer back, where we have analyzed every category. There through the collation process, it was shown if the IDS is associated with the Risk or not. If we found the Risk to be irrelevant with the ID systems, we moved on. If not we tried to find the main vulnerabilities, which will be used in the synthesis of our model.

So, the process went as:

- Analyzing the Risk
- Find associated vulnerabilities
- Explain whether the IDS can be used to cease the vulnerability
- If so, add the category to our solution, if not continue to the next

Through this procedure, we come to focus on three out of the ten main categories which are:

Multi Tenancy and Physical Security,
as the title indicates the term physical security, the choice could have been made blindfolded, without even making the analysis. The category vulnerabilities here are:

- Denial of Service (DoS) attacks
- Cloud Malware Injection Attack
- Side Channel Attacks
- Authentication Attacks
- Man-In-The-Middle Cryptographic Attacks
- Malicious Insiders
- Scanning other tenants
- Hypervisor attacks

Incidence Analysis and Forensics

as we saw that the logging abilities of the IDS can provide invaluable help to the forensics analysts in order to backtrack and the we are pointed out:

- Malware detection
- Intrusion detection response

- Honeypots
- Logging

And finally, Infrastructure Security, where still by no coincidence the word security is present with:

- Internet Dependency
- Active Unused Ports/Port Scanning
- ARP Spoofing

We see that the risks selected are more of a technical issue, or perhaps more technological, which shouldn't be surprising, as the IDS systems, no matter how well-developed they are, they are still a tool for auditing, supervising, analyzing, logging and alerting. An advanced tool, but still a product of technology, while the rest of the risks fall more in to the rule base set needed by the countries, infrastructures, organizations and companies involved.

2.5 Attack taxonomy in cloud computing

The attacks can be organized and categorized by the source of the attack, and their specific target. This is very useful, as they have different attributes, and individual behavior which cannot be associated with the rest. This helps in both preventing and identifying the attack. The categories are quite self-explained by the title but it is essential to be overviewed. When user is mentioned, could mean unintentional action by the cloud user, or an attack attempt from a hacker.

2.5.1 Service-to-User

When a specific service can be manipulated and affect users, most prominently cloud tenants.

2.5.2 User-to-Service

When the user himself tries to abuse a specific service provided by the cloud

2.5.3 Cloud-to-Service

This refers to the use of the cloud infrastructure to attack a specific service. The Cloud becomes a pool of resources for the attacker.

2.5.4 Service-to-Cloud

The Cloud itself is the target, most commonly by zero-day found on some of the commonly used applications, and trying to bypass the established security for malicious means

2.5.5 Cloud-to-User

When the Cloud resources are used to take advantage of independent users.

2.5.6 User-to-Cloud

An individual user with his own resources trying to create an exploit to manipulate the cloud, while not falling to the previous categories.

3 Related Work

3.1 Categories of IDS/IPS used in Cloud Computing

3.1.1 Host based Intrusion Detection Systems for Cloud(HIDS)

As we saw earlier, what we call a host-based intrusion detection system (HIDS) is nothing more than an intrusion detection system, which monitors and analyzes the information that a specific host machine collects. The HIDS is running on a host machine and its role is to detect the intrusions that happens to that machine, for the machine. That process is taking place by collecting information such as:

- File system used
- Network events
- System calls

The HIDS works by observing the modification in the host kernel, the host file system and the program behavior. An existence of attack is detected due to an expected behavior, which has been reported. The HIDS efficiency is depending on the chosen system characteristics that is called to monitor. The HIDS is able to detect an intrusion for only the machines that is placed on.

To be more cloud specific, a HIDS can also be placed to Cloud computing, and it can be installed either in a VM or in hypervisor, which is a different layer in cloud computing. Its role is to detect the intrusive behavior, by monitoring and analyzing the files which have been logged, the security access control policies, and the users' login information. When it'd been installed on VM, the HIDS can be monitored, something that is recommended for several reasons. [21]

3.1.2 Network based Intrusion Detection Systems for Cloud (NIDS)

A Network based Intrusion Detection System (NIDS) is an intrusion detection system. Its role is to detect malicious activities, such as:

- DoS attacks,
- Port scans
- Attempts to crack into computers

This is made by monitoring network traffic. The system collects information from the network. Then it compares the collected data with known attacks, so to detect intrusions. The NIDS detects threats with its stronger detection mechanism, commonly signatures, but behavior analysis is also found. It's necessary so it can detect network intruders. The process to do that is by comparing the current behavior

of the monitored system, with the already observed behavior, in real time. The NIDS monitors IPs mostly.

As we care about the Cloud at this moment, the NIDS can be installed, in the VMs, the perimeter of the Cloud or in the internal network. There it also monitors the transport layer headers of the individual packets. Then it detects the intrusion activity. The NIDS uses techniques which are based on signature and anomaly intrusions. The NIDS visibility is very limited inside the host machines.

When the network traffic is encrypted, the NIDS has no really effectiveness to decrypt the traffic for analysis, and this is something we must care for, especially with the incoming connections through the internet in the cloud system [21]

3.1.3 Distributed Intrusion Detection Systems for Cloud (DIDS)

A Distributed IDS, also called DIDS, consists of several IDS, such as the HIDS, the NIDS etc., It is a combination of them. It works over a large-scale network, and this ability makes it an excellent choice for the cloud. Mind, that we had found HIDS and NIDS, as general terms, but we mentioned DIDS only in the cloud section. All of them (the members of the distributed IDS) communicate with each other, or with a central server. This enables them for network monitoring.

Components are collected and may show a possible intrusion, this outcome is derived by detecting the system information, and then are converted into a standardized form, so it can be passed to the central analyzer. Most of the times the network has a central analyzer, which is a machine that aggregates the information's from multiple IDS, which is not obligatory, but yet it is the most common practice. It analyzes the information in the same way.

Detecting anomalies in behavior with combinations, which as we saw earlier, exist in various algorithms and signature detection approaches, are the base for the detection itself, and they are also used for analysis purpose. The DIDS can detect known and unknown attacks. It can do that by taking advantage of both the NIDS and HIDS, which are complement of each other. [21]. This hybrid approach, seems to be very appealing for use, as it can combine so many elements, that communicate together and share knowledge and responsibilities.

3.1.4 Hypervisor-based Intrusion Detection Systems for Cloud

The Hypervisor-based intrusion detection systems are found exclusively in the Cloud environment. It is an intrusion detection system that it is specifically designed for hypervisors. The Hypervisor is “a platform to run VMs”, a different layer of cloud computing.

When the system hypervisor layer is running, this IDS type allows to user “to monitor and analyze communications between VMs, between hypervisor and VM and within the hypervisor based virtual network”. Information availability is only one of the

benefits when a hypervisor based IDS is been used. The challenges that IDS faces are “novelty in the technology and lack of experience”. [21]

3.1.5 Intrusion Prevention Systems (IPS) for Cloud

The IDS can help IPS to monitor a network traffic and the system activities. By that way, it detects the possible intrusions. After that it dynamically responds to the intrusions by blocking the traffic or even quarantine it. The IPS can also configure the accurately expected results, which is its first priority. If that’s not possible, the IDS stop the packets flowing. That makes unavailable the network.

To prevent intrusions, computers uses firewall with IDS. They contain rules with signature specifying network traffic. IDS is based on the preconfigured rules. The IPS system decides whether “*network traffic should be passed or blocked*”. Then it detects attack in response, and the IPS can also stop the attack itself. It can also change “*the attack contents or change security environment*” [21].

These systems are also implemented in the cloud environment, and that’s why we acknowledge them here. The can be considered as an extension of the typical IDS with retroactive controls, which can take predefined actions as reactions, and not only generate alerts.

3.1.6 Intrusion Detection and Prevention Systems (IDPS) for Cloud

Individual IDS and IPS have their own strengths and weaknesses. By their own, both programs are not capable to provide full-fledged security. The combination of IDS and IPS is more effective in use. Then it is called IDPS. It doesn’t only identify possible intrusions, the IDPS also stops and reports threats to security administrators. The configuration and management of IDS and IPS much be the proper one, and their combination can improve the security to any system. NIST explains how intrusion detection and prevention can be used in combination to strengthen the system security. It also presented different ways on designing, configuring and managing IDPS. They also can be found in the cloud in general terms. It is not a cloud specific technology, but it can be used, and it is used, in the cloud environment. The IDPS is classified into three broad categories:

- Signature-based,
- Anomaly-based, and
- Statefull protocol analysis. [21]

3.2 Methods Proposed

The Intrusion Detection Systems are used in the cloud as we saw, with different variations. Now, we will study some of the frequently used commercial models and academical proposals. Of course, many commercial models are not disclosed for reasons of market competition, and even in academia there is a large number of new proposals, as it is an evolving field, which grows alongside the Cloud technology.

3.2.1 Central management approach

Xin Wang, [85] proposed and developed a more centralized ID for the cloud, a framework of overseeing the security management in more authoritative approach. This created a very fast and accurate model however, it lacked scalability. It is not as efficient, as the scale increased, and the cloud technology infrastructures tend to expand and this makes the implementation and maintenance of this proposed system difficult.

3.2.2 IDS as Hardware

Parag K, [86] had a completely different approach to the problem. Instead of inserting IDSs in the internal network of the cloud, to construct a new device with the IDS firmware installed in it. This device would be placed in the outer network, and would check the incoming traffic. It is a NIDS based system, however, placing it outside the internal network, makes it more vulnerable to attacks, and it can't do anything for attacks, from inside the cloud e.g. scanning other tenants. These are its major disadvantages; however, it could be a could addition to a bigger architecture, but not as a standalone solution.

3.2.3 Open Source IDS

Amirreza Zarrabi, [87], proposed an implementation of the open source Intrusion detection software Snort adapted for the cloud environment. They used Linux as an operating system, Snort is an NIDS with all the benefits and negatives this one-sided choice has. Another drawback is the constant upgrade of the signatures needed. We find the use of Snort as part of our system a really intriguing idea, however just using this is not enough secure for our set standards.

Mazzariello [98], also used Snort to integrate it as a NIDS into an open source cloud computing environment. The cloud is the open source Eucalyptus Cloud. To do so, they installed Snort in both the Cloud controller as well as the hosting virtual machines. This is done, and efficiently detects attacks, but only from the external network, and is too reliable to the Snort signature database.

3.2.4 Solutions with combining concepts

Patel A, [88], proposed a system that combined many technologies and approaches. Those included Risk management, fuzzy logic, ontology and autonomic computing. While in theory, this model is at least promising, the algorithms needed haven't been released, nor an implementation of this has been made. So, it's strengths and weaknesses in practice are yet to be proven. However, this hybrid approach looks really interesting to us, especially the Risk management aspect of it.

Vieira k, [89] proposed a system for grid environments which can be implemented also in the cloud ones. It combines methods such as misuse intrusion detection and anomaly based intrusion detection, so it is actually an interesting hybrid, in regards of the techniques used, method. However, its major drawback is that it can only detects and alert specific kinds of attacks, and all of them. So, we don't find it that efficient.

Ahmad's [90], approach was inspired by the artificial immune system, especially the self / non-self-discrimination technique. The proposed architecture had integrated many systems and techniques like a Hybrid Analysis Engine, with both rule-based and artificial immune system engine techniques, where the hybrid engine, will analyze the packets based on the rule-set and the system engine. The system also has sensors installed on users' network, so before reaching the Cloud IDS, the packets will have been already selected and inspected. This makes the use of this model viable for certain only applications, as the installation of a part of the IDS in the client is a great solution for the SaaS model, but the more abstract nature of IaaS, prohibits us from using it.

Rajendrana [91], wanting to take advantage of the self-adaptability of a hybrid system architecture, proposed a system that doesn't change its characteristics and will still be efficient in detection process, whenever the cloud system is increased or decreased. The proposed system was implemented using .Net framework as front end and SQL Server as back end to store the information. It is fully functional, using also a user's interface. It runs both on hosts and network and uses anomaly Intrusion and misuse intrusion techniques. From this approach, we like and want also to make use of the self-adaptability property of the hybrid systems in our own proposal.

Arshad [92], while only being an abstract model, with no implementation, combined six different components. These are:

- system call handler
- detection module
- security analysis module
- profile engines
- global components
- intrusion response system

The detection module uses both anomaly and signature based techniques, and thus this proposal falls in this category. This architecture is really appealing, and we like the use of many components, which do a different job, and all together have a holistic approach towards the IDSs. It has profiles to understand different behaviors, a handler to inspect the system calls, but it is a HIDS specific system (and that's the reason of the call handle). A combination of this proposal with its NIDS counterpart would provide a better solution.

Singh [100], with the Collaborative IDS Framework for Cloud is a NIDS system. It has three self-explanatory phases.

- Audit Phase
- Learning Phase
- Detection Phase

It combines the use of the signature based Snort, to inspect incoming traffic, but yet trains its own algorithm. They use the data collected, and create the anomaly detection system based to them. It also self-generates signatures to be imported in Snort.

3.2.5 Solutions with a single technology

Lee [93], used only HIDS, in a layered system, to maximize their efficiency. It's user behavior is set in a different layer (low, medium, high) and so their potential risk is calculated. Then with the use of anomaly detection techniques, the appropriate layer of IDS, that is responsible for the same security level is selected. This approach provides a fast detection mechanism which is always useful. As a drawback, it is single-minded as being only host based.

Kwon [94], for his system, chose only the Host based approach, but did a great improvement to it, by making it more lightweight. Using two techniques, host and a hybrid one, calculates the self-similarity of behaviors. In great deviations, from the normal behavior, the system is alerted.

Hemairy [95], proposed a system based solely in NIDS, that solved only a specific (but also serious problem). That of ARP spoofing. It was tested through experiments and found to be secure, and here we understand the work needed for a holistic IDS to be successful. As we can see a whole system to be created for only one attack. This approach has been evolved even more and now can alert for other threats, which take advantage of the ARP protocol.

Roschke [96], took under consideration the Virtual Machines, which are used in the cloud. It is only NIDS, but uses two components.

- IDS sensor
- IDS management unit

The sensor checks for the malicious behavior, while the management unit gathers, stores, and analyses the events. The IDSs in the VMs are contacted remotely from a main controller. Through the analysis of the dataset gathered, the alarms are being risen or not. This is an interesting model, as it combines security for all the VMs and a database for the events to be stored, and being compared. However, being only an NIDS, can't manage all the threats efficiently.

Bakshi [97], used a single NIDS instance for detecting the incoming DDOS attacks in the cloud system. The NIDS is being installed in a virtual network node (as the system protects also only the VMs and not the whole infrastructure), and the packets traversing through are being inspected and compared to known signatures. If an attack is identified as a DDoS it blocks all the zombie machines. The system is not only alerting, but also responds, by modifying the firewall to block the incoming connections, from the identified IP addresses. This is a very good solution, but only applicable for a single attack methodology.

Mane [99] proposed an anomaly based IDS using Backpropagation Neural Network. The system has a step by step flow with the stages being:

- Data collection
- Data Preprocessing
- Representation and Normalization,
- Dimensionality Reduction
- Selection of Network Structure
- Training and Testing
- Attacks Classes

The system uses Artificial Neural Network (ANN) for its implementation. It has a two-layer feedforward network with sigmoid hidden neurons & linear output neurons fit multidimensional database. Basically, it is a NIDS system with behavioral properties regarding inspection. An interesting implementation and practical solution to hierarchical anomaly intrusion detection system using supervised learning method., but lacks the attributes of hybrid systems [99]. Only the algorithm was tested in Matlab.

Babiker [101] designed a Hybrid Algorithm for Cloud Computing Security, proposed an IDS which relies on the genetic algorithms. Their work defined as a mix of Cloud

Intrusion Detection System and two types of chromosomes based on different criteria. The first type is created based on the job length, the second type is created based on the bandwidth of the resources criteria are the input parameters of the fuzzy system. [101]. So, they used genetic algorithm as the basis of their approach and modify it with the aid of fuzzy theory to cover security requirements.

Ansari [102] had a Hybrid approach towards NIDS in his Framework for Hybrid Network Intrusion Detection and Prevention System. This approach uses three techniques

- Clustering
- Classification
- Feature Selection

The hybrid approach however regards the techniques used, not the use of different systems, as only the use of NIDS is proposed.

3.3 What we learned from these Methods in Use

We could keep mentioning more and more methods, architectures and technologies proposed and used. Notice that in this chapter (3) there was no reference to any of the systems referenced in the first (1) chapter as examples, which are more examples we studied.

The intrusion detection problem, is a problem that can be solved with many ways. No silver lining exists, and all the methods have their positives and negatives.

However, we noticed in some proposals, a one-sided view of the problem. Meaning, that they proposed e.g. a new algorithm for detection, and tried to build a whole system upon this innovative idea, rather than combining this, with other established technologies and techniques. Possibly this was done to demonstrate all the benefits of their innovation, but in practice, such an implementation will fail, because of the security gaps it leaves open.

We want to have a more holistic approach, to combine the best elements for these methods, to a big, practical and reliable model.

4 Proposed Solution

4.1 Methodology Used

In the previous chapters, numerous implementations and variations of topologies and technologies for the IDS were presented, as the IDS are essential for the functionality of the cloud. However most of these models fell in the same design pattern. Taking a new technology, an innovative idea, and try to manipulate it, adjust it and have it parameterized to face the needs, match the already global success standards of other systems, and make it efficient in an ever-changing environment. These solutions, are like making a tool, which can be very good for certain uses, but is not that great for others and its owner still uses it, narrow-mindedly as he doesn't have a collection of tools, each of it specifically designed for use with a different task.

The following proposed solution takes a completely different approach. Instead of using just an idea, and altering in to fit the goals and requirements of a cloud system, it will use the knowledge acquired, regarding the real risks of the cloud, study each risk separately and refer to the existing technologies, deciding which known IDS is the better for each one. Doing that for each known risk, a complex, large and of course resource demanding, unnecessarily huge IDS model will be the result.

From this complex and large model, we will proceed with abstractions, aggregations and simplifications, and by doing so a new model will derive, which will tackle each problem. With this methodology, the problem always dictates the right solution, and not the other way around, which is having a solution, and trying to adjust it to all problems. With the procedures mentioned the model will be lightweight (with the abstractions being made), presumably decentralized, (this waits to be shown but it is expected), and last but not least expandable and adaptable to new risks and threats found. That is because by not using a single technology, or a small set of them, when there is a need to implement a new countermeasure, it can happen easily without disturbing the whole existing system. Starting with the categories and analyzing the threats in every one of them we get the following: The added partial solutions will make the complete one.

So, every threat will be analyzed, paired with the best solution, and then a table will be presented to make the result even more clear. Also, a schematic to show the placement of the IDS will be also useful. At the end, a bigger table will have all the individual results to derive the proposed system.

A visualization of the description of our work process follows.

METHODOLOGY



Figure 6 Overview of the Methodology Used

Here we can see what we explained, the steps that will be followed to create our model. Take each category (with vulnerabilities that can be detected with an IDS as found in chapter 2), examine each vulnerability (as they were found in chapter 2 also) and find the best IDS for each one of them. Repeat this for every vulnerability for each mentioned category, and then at the end combine all these mini-models, to create our proposal. (the model will be simplified of course, but this will be described in detail in time).

This is the general process, but we will use another figure to visualize the process as a flow chart.

In order to find and determine the right IDS for each vulnerability, we will use a combination of bibliographic research, but mostly with using our own association rules to determine the right solution. The association rules, are not just general assumptions but the product of understanding of the IDS systems, made possible from the previous chapters.

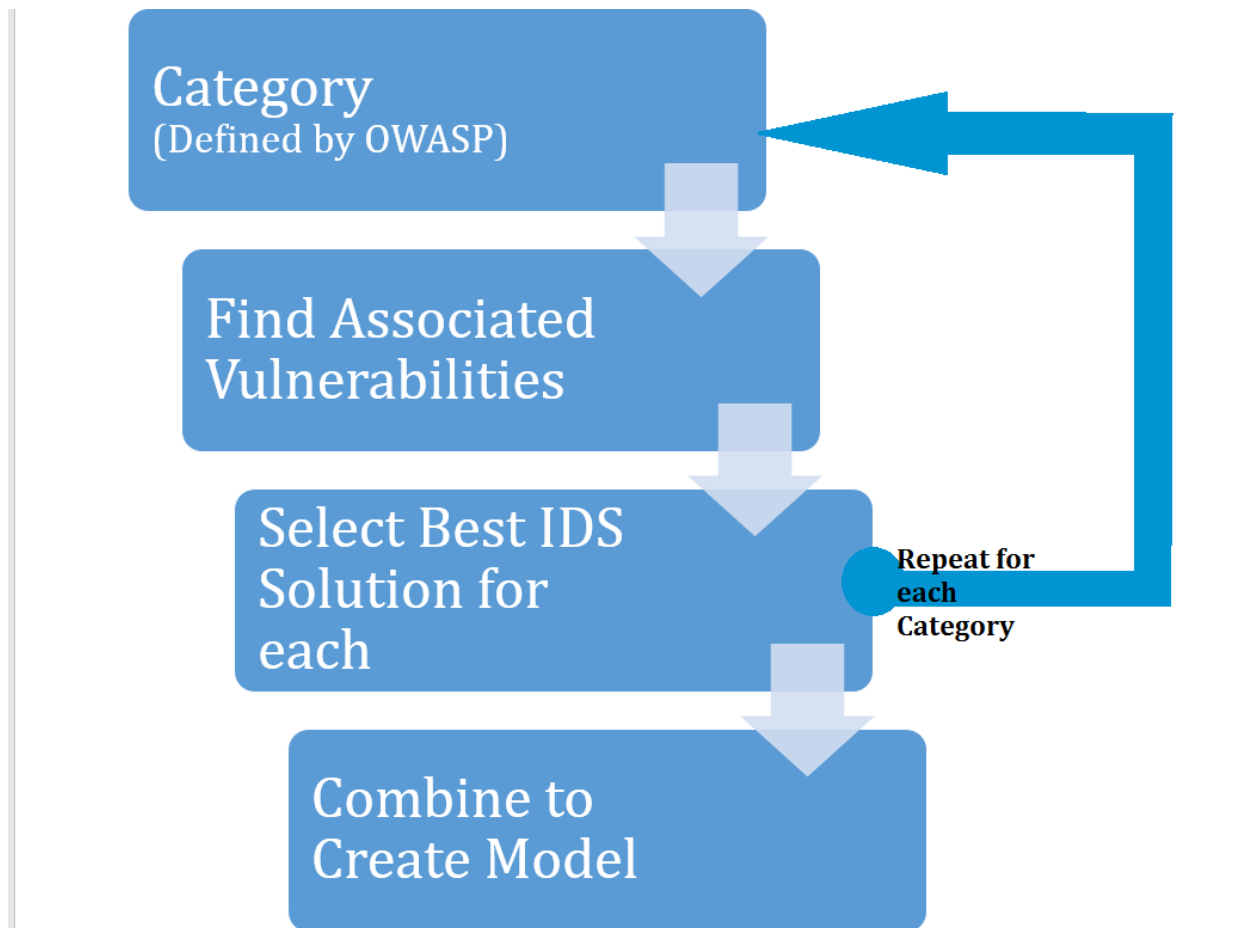


Figure 7 Methodology Flowchart

To further harden the Association rules we talked about, we also include academic bibliography, that show the benefits and drawbacks of each system, to make the selections even more easier and profound.

Differences between NIDS and HIDS

NIDS	HIDS
Broad in scope	Narrow in scope
Easier setup and configure	More complex setup and configuration
Better for detecting attacks from the Outside	Better for detecting attacks from the inside.
Less expensive to implement	More expensive to implement
Detection is based on what can be recorded on the entire network	Detection is based on what any single host can record
Examines packet headers	Does not see packet headers
OS-independent	OS-specific
Detects network attacks as payload is analyzed	Detects local attacks before they hit the network
Detects unsuccessful attack attempts	Verifies success or failure of attacks

Table 1 Differences between NIDS and HIDS [84]

4.2 Multi Tenancy and Physical Security

In chapter 2.2.7 we found that in the category Multi Tenancy and Physical Security risks that associate with the use of an IDS. These are Denial of Service (DoS) attacks, Cloud Malware Injection Attack, Side Channel Attacks, Authentication Attacks, Man-In-The-Middle Cryptographic Attacks, Malicious Insiders, Scanning other tenants and Hypervisor attacks. These will be analyzed using the model of the last paragraph to find the best solution for each one.

4.2.1 Denial of Service (DoS) attacks

A denial-of-service (DoS) attack is a cyber-attack where the attacker seeks to make a machine or network resource unavailable to its nominal users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled [28]

An example of a cloud attack is given below.

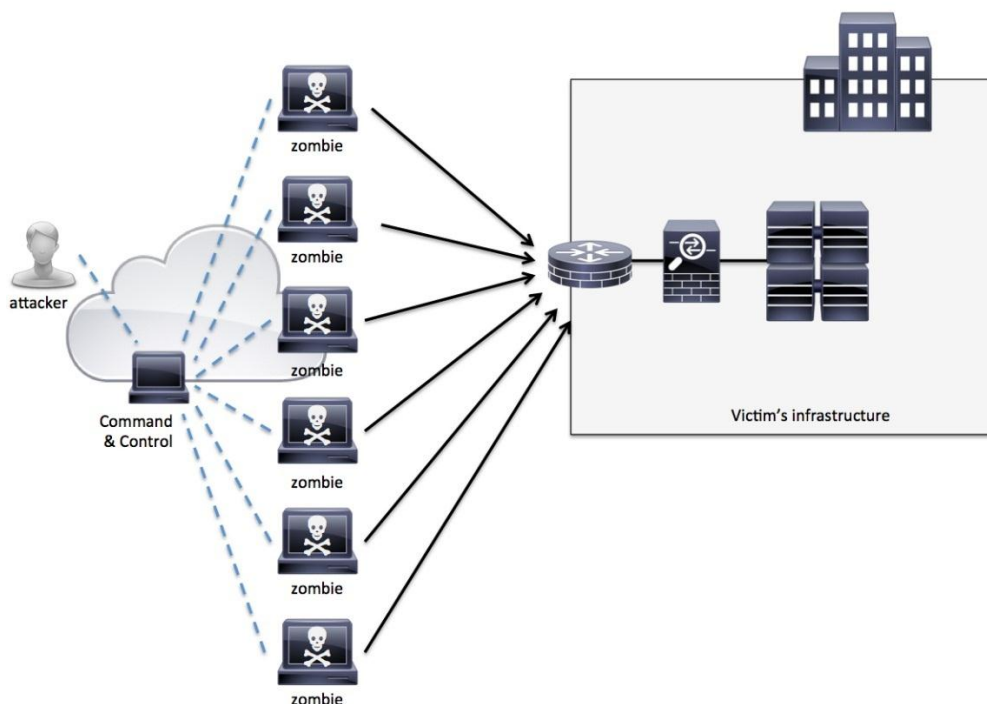


Figure 8 botnet example [29]

Traffic monitoring is essential when dealing with these flooding attacks (DoS attacks are of this kind), as they include several types of methods, which all regard the traffic itself. They can be divided in three large categories

- these with a large of number of bytes (for each packet)
- these with large number of packets
- these with packets with malicious behavior protocols such as SYN attack [30]

The last one is the generalization of many types of attacks, which include (smurf attack, ping of death attack, IP spoofing attack, buffer overflow attack, teardrop attack, land attack, SYN flood attack, Internet Control Message Protocol (ICMP) flood,). This is visualized at the figure below.

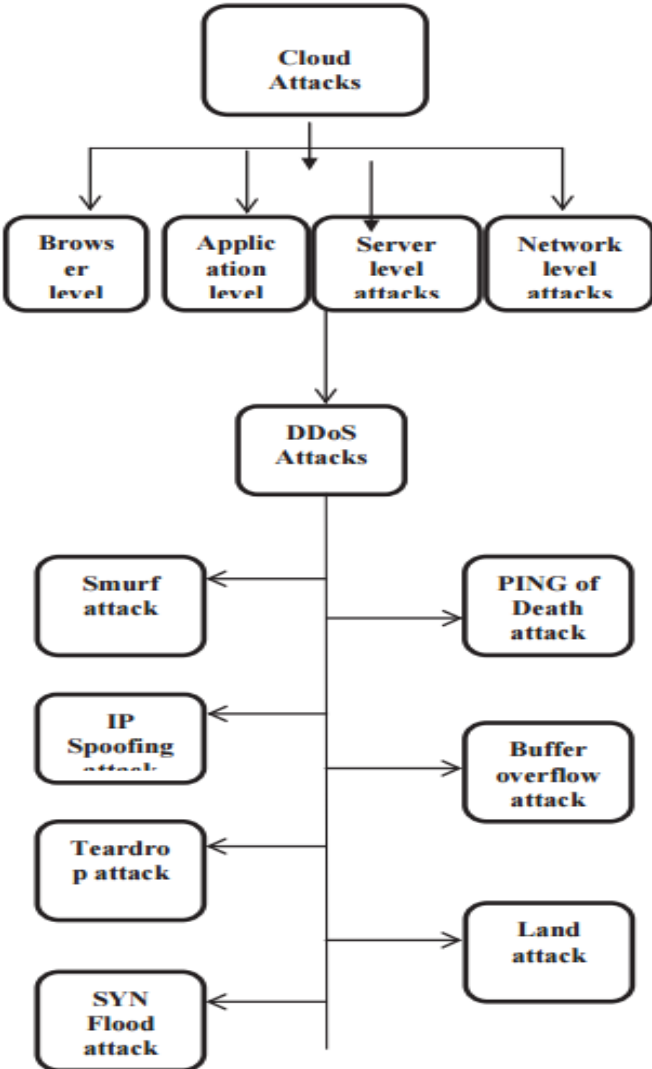


Figure 9 Cloud / DoS attacks [31]

The affected Server level and Network level attacks fall into the IaaS model, as pointed earlier, as they affect purely and directly the infrastructure of the installment. However, we

must notice that such an attack might not only target on the Cloud's providers gateway on internet, but even inside the cloud in the specific clients' hosts, hosting granted virtual machines.

To choose the correct IDS is more difficult because of these special conditions, (the target of the attack, it can be either the Cloud's gateway or each VM) therefore we must have two different cases. So,

- For the case of the target being each Virtual machine
- For the case of the target being the Cloud's gateway

a) For the case of the Virtual machine.

Regarding the detecting the DoS attacks in the cloud computing environment, it is widely proposed to use the Dempster - Shafer Theory (DST) of operations in 3-valued logic and Fault-Tree Analysis (FTA) for each VM-based Intrusion Detection System (IDS) [43]. It is a general framework for reasoning with uncertainty, with understood connections to other frameworks such as probability, possibility and imprecise probability theories. [61]

To start elaborating with our logical association rules,

- For IDS type → NIDS as it identifies intrusions by monitoring the network traffic, which is what is needed here.
- Technique used → Signature based detection as being the most commonly used technique in the cloud, large databases of signatures already exist, and can be reliable [5], and we don't have to turn to the of Dempster- Shafer behavioral models.
- Positioning → On each VM, as they are the target.

So, for the DDoS attack detection in virtual machine

IDS Type	NIDS
Technique used	Signature based detection
Positioning	On each VM
Pros	VM from DDoS attacks
Cons	Can only detects known attacks

Table 2 DDoS attack (a) [21]

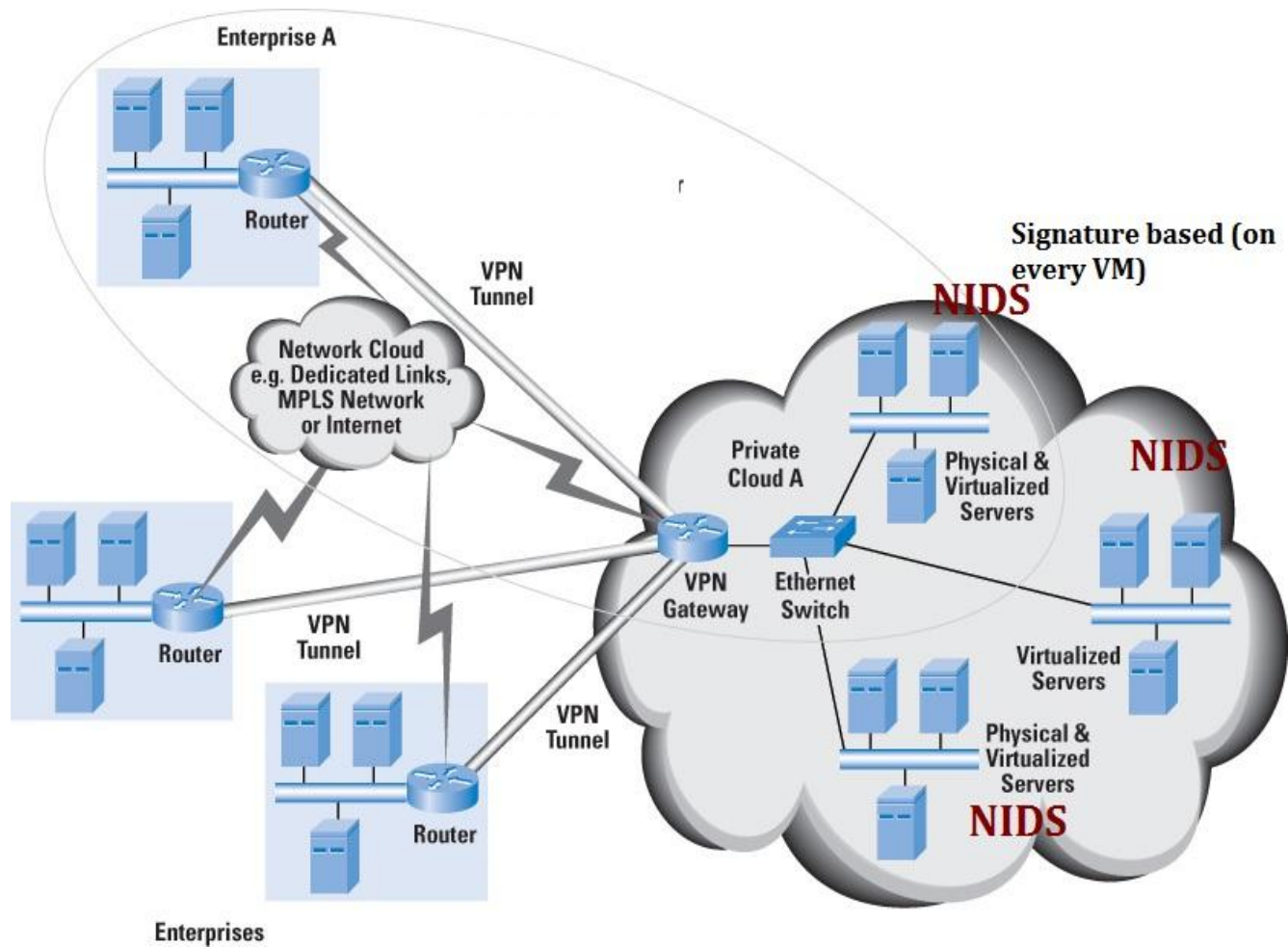


Figure 10 DDoS attack (a)

b) For the case of the Cloud's gateway being the target.

This can be considered as an attack against any network device or gateway or server as it has the same characteristics, and it is not differentiated because it targets the whole cloud environment.

Considering the gateway device, as the target, the only logical decision for the IDS system is to be directly integrated inside that gateway. While being a different logical entity (and in order to maintain this) having the IDS system inside the gateway has no practical use. The system position has to be just before, or just after the device. Installed after, in order to make sure that the checked traffic is intended for the cloud system, (and have the benefit of scanning the traffic packets already inside the ecosystem), or in the network perimeter,

The most sophisticated use is based on adaptive behavioral-based (anomaly) and signature based technologies, in order to provide integrated intrusion detection and DDoS attack prevention systems. This works against both network and application-level DDoS attacks. Another popular solution is that of Radware. [34]. They use behavioral based models DoS feature which rapidly mitigates zero-day DDoS/DoS attacks, by automatically generating

real-time solution signatures. Other companies like Cisco use signature based and have even published the signatures of the attacks covered. [35]

Being a vast research area many other models have been proposed such as, an Entropy based approach [36] which is a Network IDS in a Cloud Computing Environment, but is included in the aspect of the Anomaly techniques, being a measurement of randomness. For every incoming request session the entropy is calculated and is compared to a predefined price. Then the system proceeds and checks using a defined standard to check if a larger deviation is found. If this happens then the user request of that session is asserted as abnormal. The Fuzzy Network Profiling for Intrusion Detection [37], which still is anomaly detection but was selected as a different category due to its special properties. Semantic rule based approach where it is wont to observe anomaly in the cloud application layer. A settled finite automated procedure, is employed to represent completely different malicious characteristics. [40]

In the previous paragraphs, we find the repetition of the combination of Signature based and anomaly based detection. Having the need to monitor the incoming network traffic, the NIDS is the obvious choice, and it needs to be placed only on the underlying network., and not on Virtual VPNs which can be supposedly used, and is appropriate for detecting external intruders (as this attack is)

- For IDS type → NIDS as it identifies intrusions by monitoring the network traffic, which is what is in need here.
- Technique used → Hybrid approach the combination of the Signature based detection and the Anomaly detection, in order to benefit from the existing signatures, but have the behavior models, just in case, considering the severity of the attack, as it can cripple the whole cloud infrastructure.
- Positioning → On Network perimeter, as it deals with the incoming traffic, and has to react before they enter the native network.

IDS Type	NIDS
Technique used	Signature based detection/ Anomaly Detection (Hybrid approach)
Positioning	Network perimeter
Pros	Can detect known attacks/ react to 0days
Cons	Training time for Anomaly detection might take a bit long

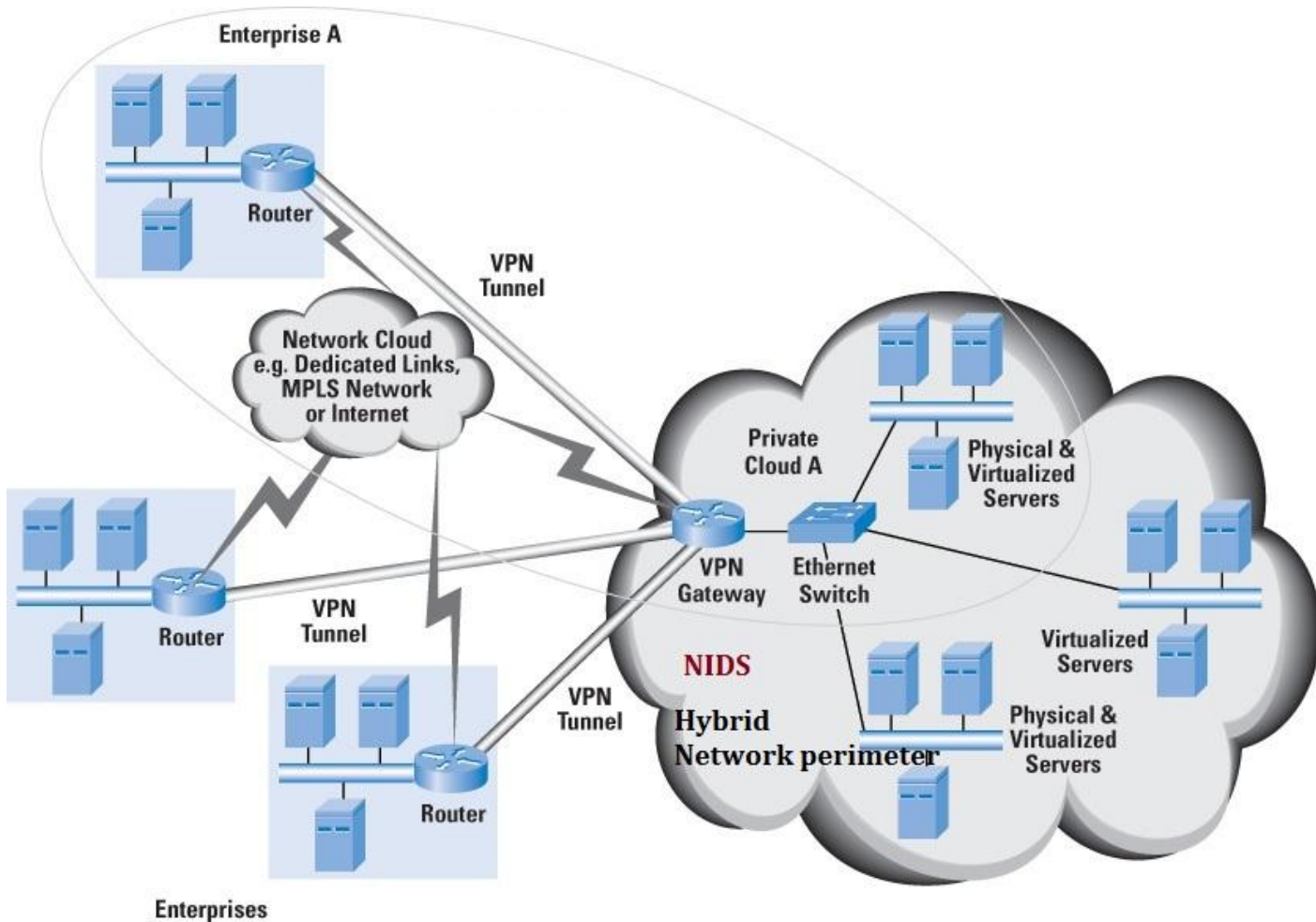


Table 3 DDoS attack (b)

Figure 11 DDoS attack (b)

4.2.2 Cloud Malware Injection Attack

Malware injections in the cloud environment are attempts to inject malicious services, scripts or code embedded into the cloud services, which then will appear as a valid service of the cloud.

These attacks include cross site scripting, injection flaws, information leakage and improper error handling, broken authentication and session management, failure to restrict URL access, improper data validation, insecure communications, and malicious file execution. [38]

Malware injection attack is one class of web-based attacks, with which hackers exploit vulnerabilities of an internet application and introduce malicious code into it, that changes the course of its traditional execution. Like web-based applications, cloud systems also are at risk of malware injection attacks. Hackers craft the malicious application, inject malicious services, scripts and inject them into the target cloud service, which includes the IaaS model which is our target. Once the injection is completed, the malicious module joins

the valid instances running in the cloud; then, the hacker can accomplish eavesdropping, data manipulation, and data theft. [39]

Regarding the IaaS model, the instance hijack which the intruder wants to create is inside a Virtual Machine instance of the structure. [42] So this attack is a direct one to the Cloud Infrastructure in order to gain access.

Usually when the customer/client initiates a cloud service, he essentially opens an image to a Virtual Machine within the cloud, and the service provider creates a picture of the customer's VM within the image repository system of the cloud. The applications that the client can run are thought-about with high potency and integrity to be the main objective. A tendency exists to propose to contemplate the integrity of the security within the hardware level, because it is terribly tough for the wrongdoer to intrude within the IaaS level. [48]

The attacks are being targeted directly to the user as the entity and the host are contained and restrained inside a Virtual Machine. Considering the difficulty of checking these attacks at a cloud wide scale, makes the positioning of the IDS, absolutely necessary inside each VM.

For the detection, we propose the Aho–Corasick Pattern matching algorithm, which has been proposed, as an anomaly detection system which engages two steps. The first being the Anomaly Score value calculation and the second the Pattern Matching Algorithm, with the known patterns, the system has a self-learning process. [41]. While the signature based recognition can be in theory achievable, the anomaly detection seems to provide a better safeguard to this specific threat, due to its dynamic reaction, and it is accordingly more suitable.

In order to achieve better analysis and results, and since it needs to be installed in every VM in the cloud, the Distributed Intrusion Detection System is the better option as it provides inter-connection of the IDS with every VM to collect data. Working in hive mind census and with the use of the central analyzer, it will react and eventually protect all the machines. The self-learning period will be shorter, and the VMs will share their results, and behave as the members of the system which they are, and not as stand-alone machines. Thus, in the occurrence of the detection of an intrusion to a cloud is VM machine / area, the other areas are also informed.

- For IDS type → DIDS, Distributed Intrusion Detection System to get info from all the machines, and get the benefits from both NIDS and HIDS, as malware can behave with various ways, (many different cases), and can be detected only by one of the two systems.
- Technique used → Anomaly Detection, as proposed by bibliography, with proper system training, due to the dynamic reaction.

- Positioning → Every Virtual Machine, as being Host based, to guarantee the integrity of all devices and VMs. Also, creates a big network of reporting new and possible attacks.

IDS Type	DIDS
Technique used	Anomaly Detection
Positioning	Every Virtual Machine
Pros	Provides IDS to all hosts (regardless of their location)
Cons	Bigger computational overhead, probable higher network load.

Table 4 Cloud Malware Injection Attack

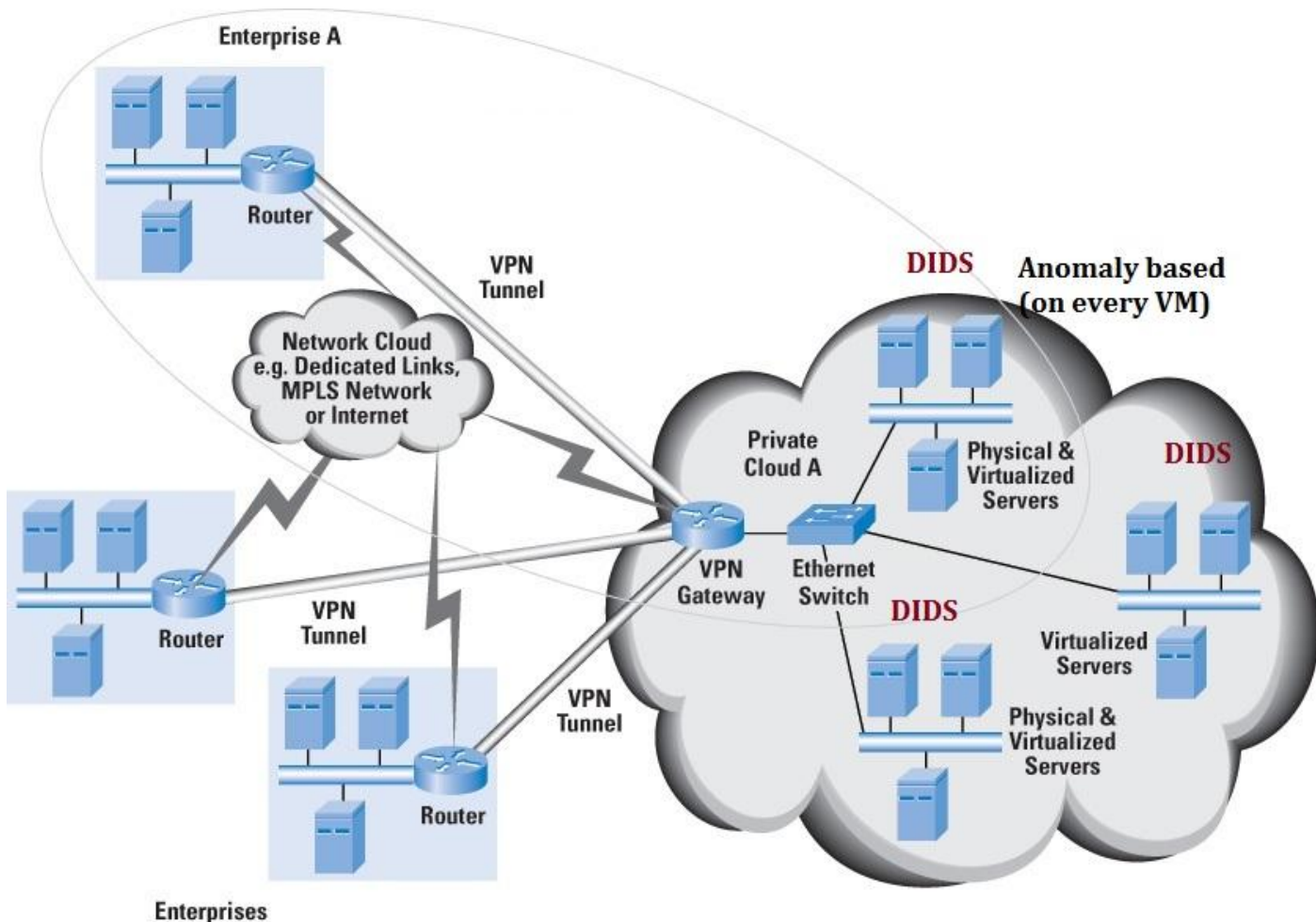


Figure 12 Cloud Malware Injection Attack

4.2.3 Side Channel Attacks

These attacks exploit the physical properties of materials to collect information that will provide a diagram or pattern of the system to attack. The fact that multiple virtual machines share same or similar hardware, side channel attack makes it comparatively straightforward to gain access. Thus, without implementation of a security device within the hardware, equipment sharing is dangerous. [44]

In cloud computing environments, it's feasible to map the infrastructure and determine wherever the virtual machine resides. It is then possible to initiate new VMs and those to be placed relatively close with the target VM, as a hardware entity, as it is an infrastructure issue after all. And once this has been instantiated, the VM assailant will retrieve sensitive knowledge from the legitimate VM attacked. This can be an aspect channel, the so said side channel attack-type. [45] Side channel attacks follow two steps.

Those are:

- Placement. Placing virtual machines and arranging them in the cloud environment
- Extraction. After placing the Virtual Machines, they start extracting confidential information from other servers or machine in cloud computing environment, though the main procedure of the attack. [49]

While a straightforward approach might suggest direct solutions to this threat, as the risk can be mitigated with the use of, for example, a virtual firewall across the cloud topology, or by encrypting the VM's contents [50], the use of IDS can considerably mitigate the threat.

Considering the similarities of the side channel attacks with the cloud malware injection attacks, as both reside and have to do directly with the Virtual Machines of the system, it is natural to investigate the possibility of a similar approach. As the attacker tries to place virtual machines in cloud environments, as his instance must be an instance on the same physical machine as the target instance to accomplish the extraction. He tries to learn information about the co-resident instances and get the ability to inject a malicious instance. Since the whole area is around the VM network, the approach of the Cloud Malware Injection Attack seems to be appropriate, as it safeguards the mass of the Virtual Machines. An approach with an NIDS system, deployed over the whole cloud ecosystem, as according to the virtual firewall, which is deployed over the system, being a similar architecture.

The first solution (the one with the VMs) is being selected though due to the possibility of encrypted communication traffic of the attacker, and while it is operational, it is more difficult from NIDS to detect intrusions in a virtual network.

- For IDS type → Distributed Intrusion Detection System, to get the combination needed due to the complexity of the attack (as defined in its stages)
- Technique used → Anomaly detection, as the malicious behavior can manifest itself with various ways, and this makes the signature based detection unreliable.
- Positioning → Every Virtual Machine, to protect (and eventually get info from all the machines) as they all are a potential target.

IDS Type	DIDS
Technique used	Anomaly Detection
Positioning	Every Virtual Machine
Pros	Provides IDS to all hosts (regardless of their location)
Cons	Bigger computational overhead, probable higher network load.

Table 5 Side Channel Attacks

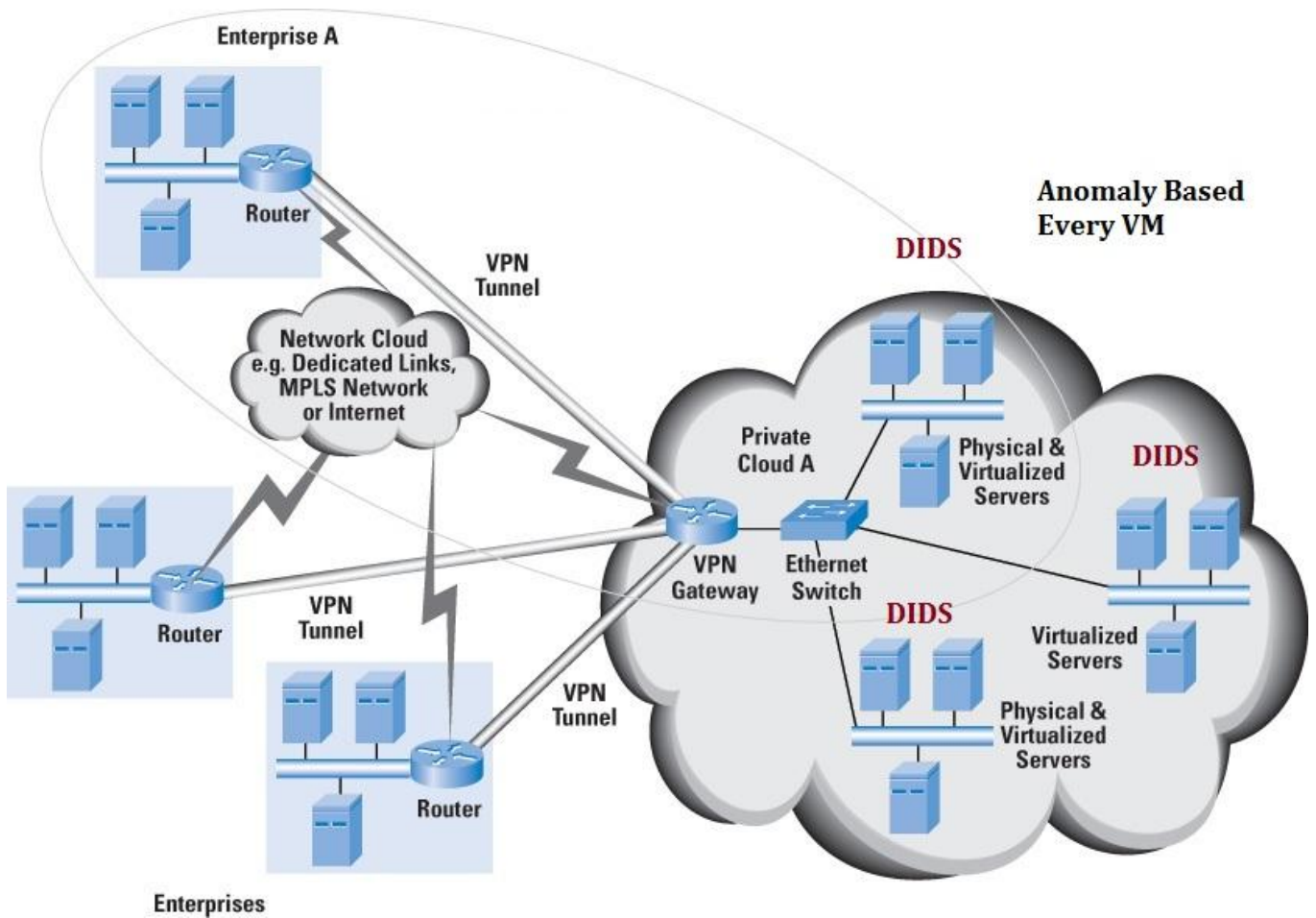


Figure 13 Side Channel Attacks

4.2.4 Authentication Attacks

These types of attacks can easily occur in cloud environments. The attackers easily target servers with these types of authentication attacks [51].

It's a category with various attack methods, which all conclude to hijack, using different techniques. These include:

- **Brute Force Attacks:** In this type of attack, all possible combinations of password apply to break the password. The brute force attack is generally applied to crack the encrypted passwords where the passwords are saved in the form of encrypted text.
- **Dictionary Attack:** This type of Attack is relatively faster than brute force attack. Unlike checking all possibilities using brute force attack, the dictionary attack tries to match the password with most occurring words or words of daily life usage.
- **Shoulder Surfing:** Shoulder Surfing is an alternative name of “spying” in which the attacker spies the user’s movements to get his/her password. In this type of attack the attacker observes the user, how he enters the password i.e. what keys of keyboard the user has pressed
- **Replay Attacks:** The replay attacks are also known as the reflection attacks. It is a way to attack challenge response user authentication mechanism.
- **Phishing Attacks:** It is a web based attack in which the attacker redirects the user to the fake website to get passwords/ Pin Codes of the user.
- **Key Loggers:** The key loggers are the software programs which monitors the user activities by recording each and every key pressed by the user. [50].

Not all these attacks can be treated with an IDS system. To start excluding, Key loggers are mostly local problems to the users’ machine(s). Phishing attack also happens, without the user reaching the cloud services. Shoulder Surfing also has a kind of physical approach with regards to the user and has nothing to do with cloud infrastructure.

The situation only leaves the Dictionary and the Brute Force attacks, which in this area can be treated as more or less the same problem. This means, not comparing the different complexity, and attempt of the attack, but rather the similarities in the victim’s end of the execution, as both have similar attributes, for the repetitiveness of the attempts.

Both these attacks can be identified from Intrusion Detection Systems, as they have some characteristics, the systems can relate to. The attacker might run user names and/or password attempts sequentially, providing identifiable pattern, but most common is a large number of login attempts from the same IP (or not) trying to access the same account, and thus Brute force attacks (and to a certain point the dictionary ones) are one of the few hacks detectable by their volume, the really large number of attempts and rather than their type.

Being a rather old attack technology, many IDS solutions have been deployed to counter this attack and this include both HIDS and NIDS technologies. HIDS can analyze system behavior and configuration status to track user access and activity by default as a main function, and the user activity (which is being imitated by the attacker) is the concern here. While most HIDS tend to have signature detection integrated, and some attacks can be indexed to a database with a signature, as they may share the password attributes, the behavior is the concern here, and not the individual packets. So the anomaly detection is preferable, which might require minimal training for such a well-known attack.

- For IDS type → Host-based intrusion detection system, as the attack is directly being made to a single machine, with the network packet sniffing not being a real countermeasure.
- Technique used → Anomaly detection (with the use of ANN for less false positives), as the behavioral analysis will lead to the detection of such attacks.
- Positioning → Each network node, all being a potential target, and being host-based, make it the only choice.

IDS Type	HIDS
Technique used	Anomaly detection (ANN for less false positives)
Positioning	Each network node
Pros	Less false positives (with the use of ANN)
Cons	Clever attackers who compromise a host, can also attack and disable host-based IDSs.

Table 6 Authentication Attacks

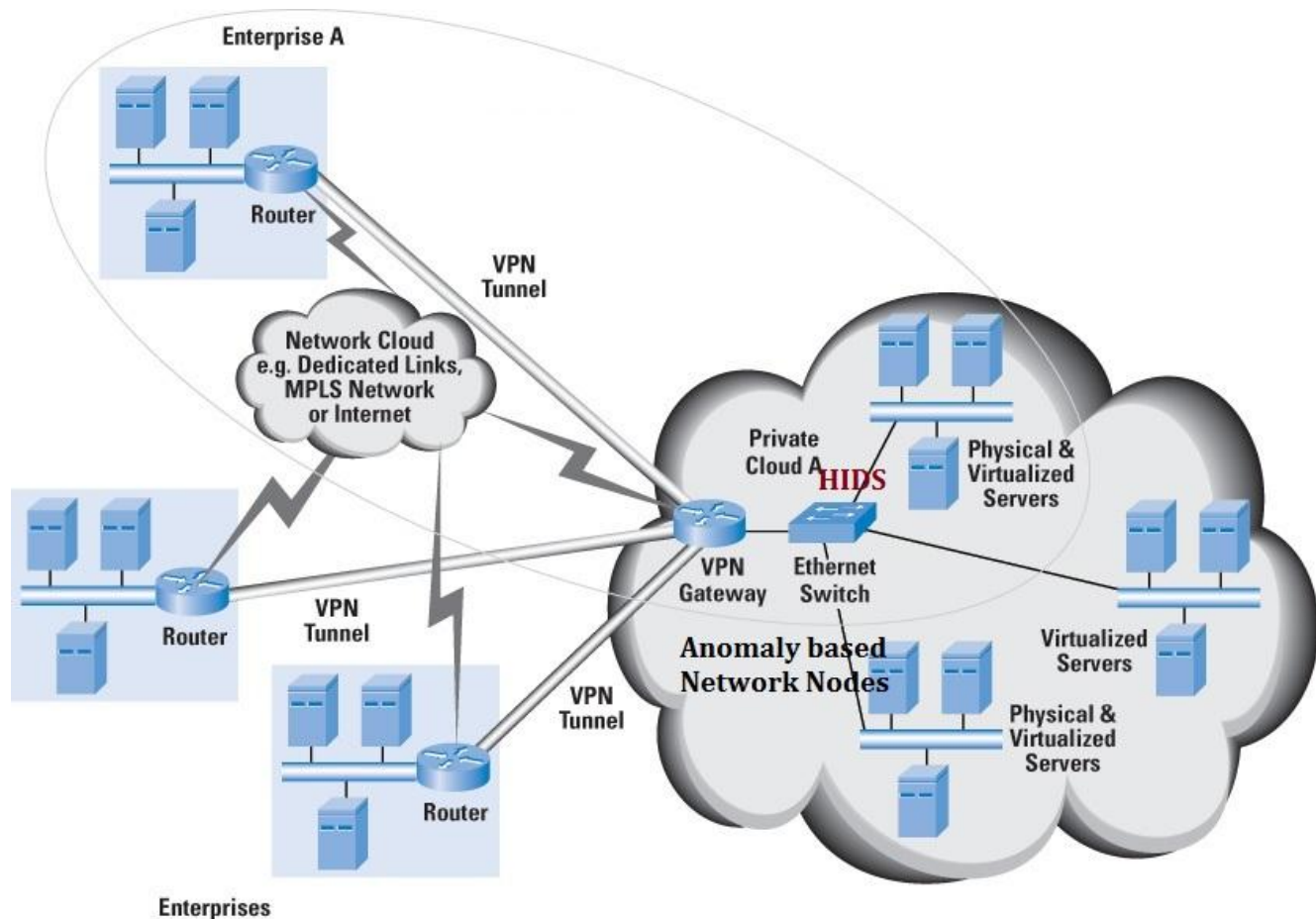


Figure 14 Authentication Attacks

4.2.5 Man-In-The-Middle Cryptographic Attacks

This attack is performed once the offender has been placed between two users in a cloud system. Anytime attackers are placed within the communication path, there's the likelihood that they'll intercept and change communications, less commonly eavesdropping, and more frequently altering the content. [46]

Intrusion detection systems which use signature based detection as Snort can monitor the network traffic of hosts. This technique is only feasible for single host monitoring but will not be adequate for an entire network. [52]

As this attack is as a communication mainly between the user (visitor) of the network and the attacker, this attack cannot be easily identified, in the host's environment as it does not fall into its responsibilities.

- For IDS type → Host-based intrusion detection system. The attack is directly being made to a single machine. Having to do with the communication between the user, and the VM in the cloud, without the packet sniffing procedure giving adequate info.
- Technique used → Signature based, as it is a standardized procedure, with clear and known attacks, in case of a new attack, the chance of a behavior based method to detect it is very slim.
- Positioning → Each network node, all being a potential target, and being host-based, make it the only choice.

IDS Type	HIDS
Technique used	Signature Based
Positioning	Each network node
Pros	more accurate than any other IDS (in the specific vulnerability)
Cons	Prone to new attacks

Table 7 Man-In-The-Middle Cryptographic Attacks

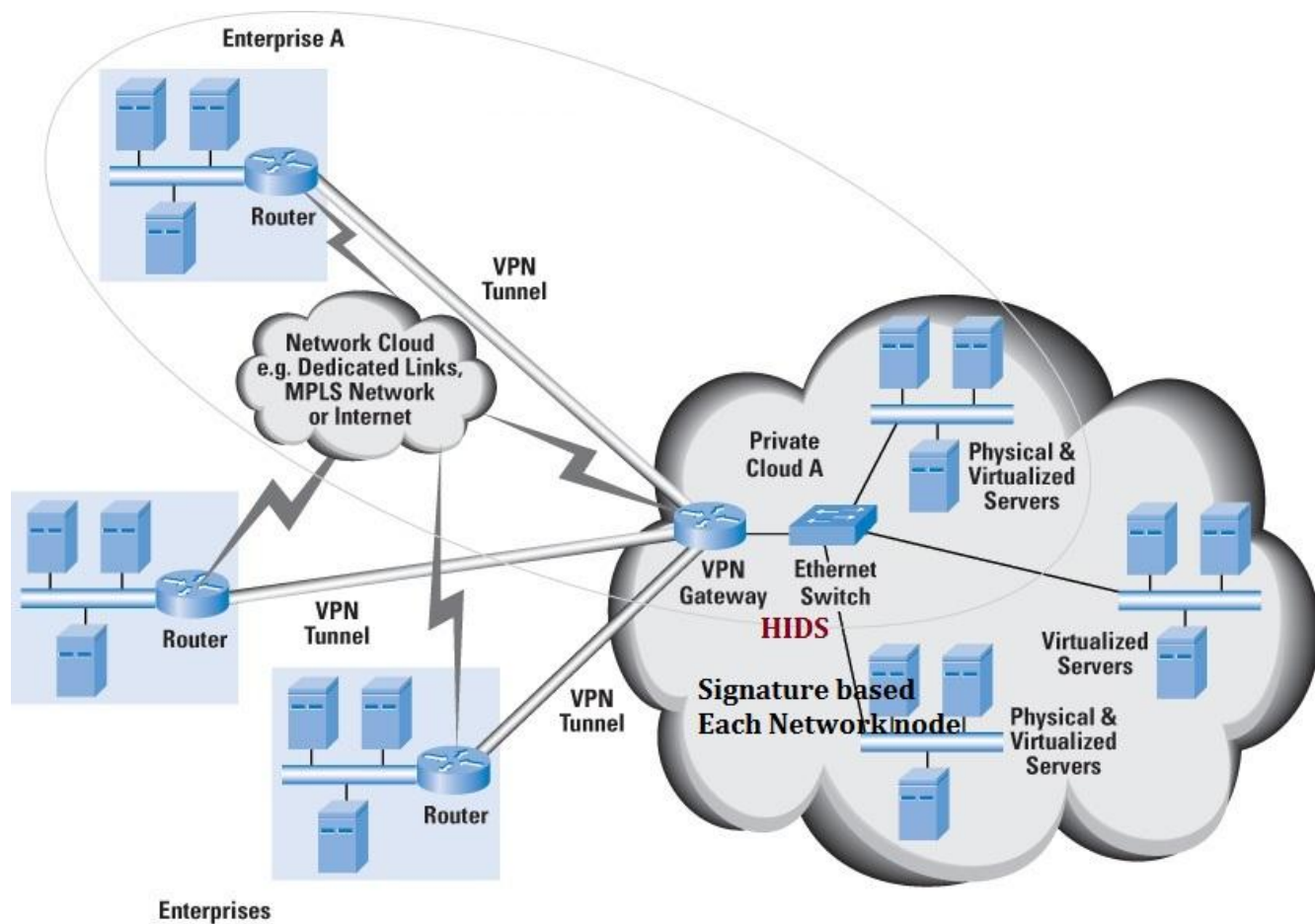


Figure 15 Man-In-The-Middle Cryptographic Attacks

4.2.6 Malicious Insiders

The term malicious here is meant to cover both the real insiders who want to do harm in their free will, but even the category of unintended harm done by users who might do it unintentionally due to a software malfunction or malware in their home workstations. They are not the most common attacks, as attacks usually come from external sources.

Of course, this doesn't mean that it is something of no importance as the graphic below shows.

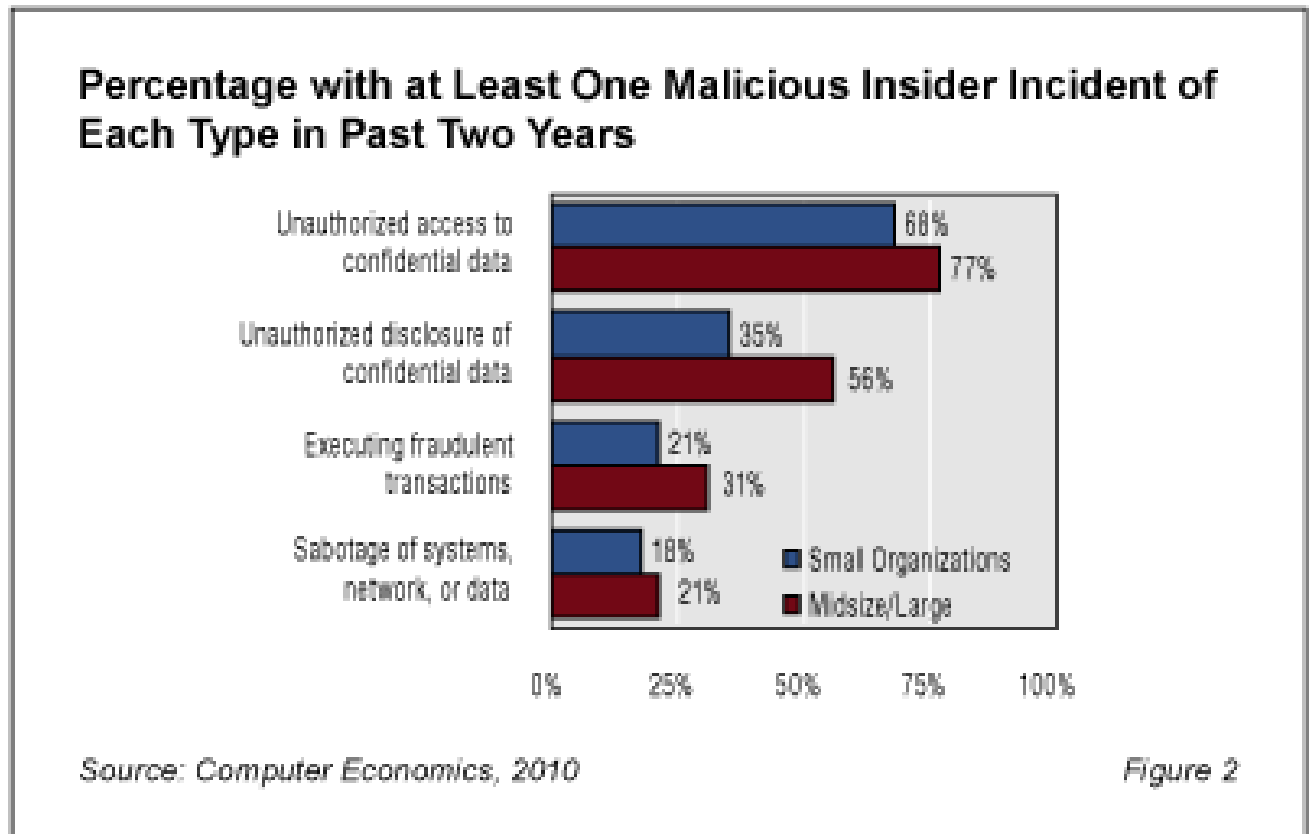


Figure 16 Percentage of Malicious Insider Incidents [55]

Obviously, the IDS need to be inside and not in the perimeter of the cloud eco-system. Configuring to detect insider attacks might be difficult. Specially to define a ruleset for this, and this makes signature based IDS systems not that effective. This happens because of the variations of the attacks. In order to maintain lightweight in computational power of the hosts, an anomaly based system, in various positions of the network can encounter such a behavior.

A great benefit using an IDS catching for attack, is that the administrators will also know the source, and other valuable information regarding this attack as it originated from the system itself, which raises the value of the logs for the whole community as new attacks can be found and examined. [55]

- For IDS type → HIDS/NIDS Hybrid approach, because of the diversity of the issue. We can't be guaranteed that a single technology can cover all the cases.
- Technique used → Anomaly detection (ANN for less false positives).
- Positioning → Internal Network, as the attack will both target and come from inside the system for the NIDS, the HIDS must be on the hosts.

IDS Type	HIDS/NIDS Hybrid approach
Technique used	Anomaly detection (ANN for less false positives) / Signature (hybrid)
Positioning	Each Host for HIDS/Internal Network
Pros	Combinations of technologies levitate the security, as the weak points are covered by the other technology
Cons	Will need more computational power

Table 8 Malicious Insiders

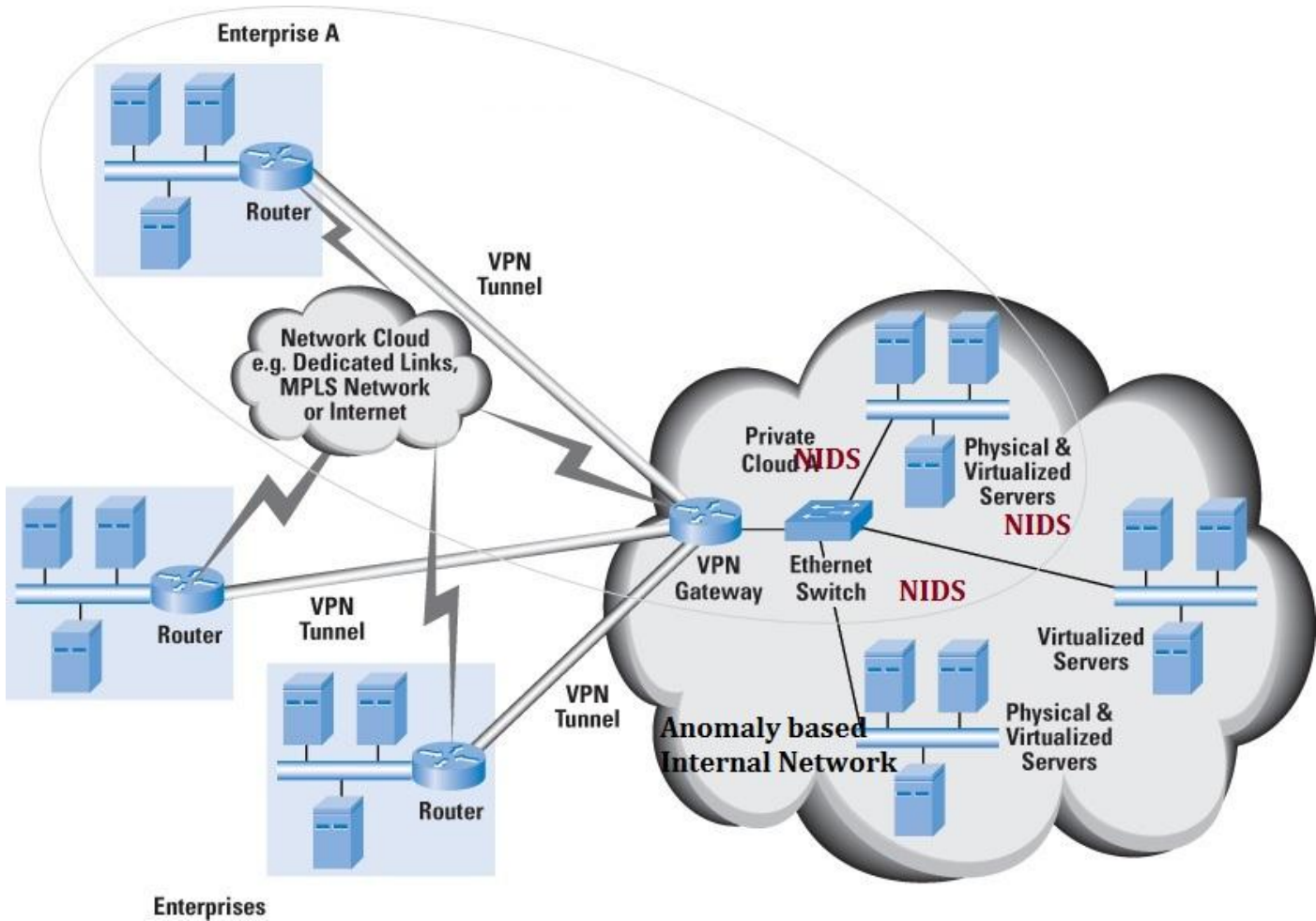


Figure 17 Malicious Insiders

4.2.7 Scanning other tenants

The scanning of other tenants has many similarities with the malicious insiders. These are that both come from inside the cloud system, and they both come from tenants/insiders.

However, as being a slightly more “simple” attack, the solution also gets simpler.

- For IDS type → NIDS, being purely a network issue.
- Technique used → Anomaly Detection, while signature detection works
- Positioning → Every network node, while it can also be put in every host, being in every node, provides a better solution in general, as it can detect a tenant scanning, while “flying under the radar”, multiple tenants. If the IDS were in hosts, this subtle approach could pass unnoticed.

IDS Type	NIDS
Technique used	Anomaly Detection
Positioning	Each network node / Internal network
Pros	Fewer false positives
Cons	Scalability due to the large traffic of the internal cloud network

Table 9 Scanning other tenants

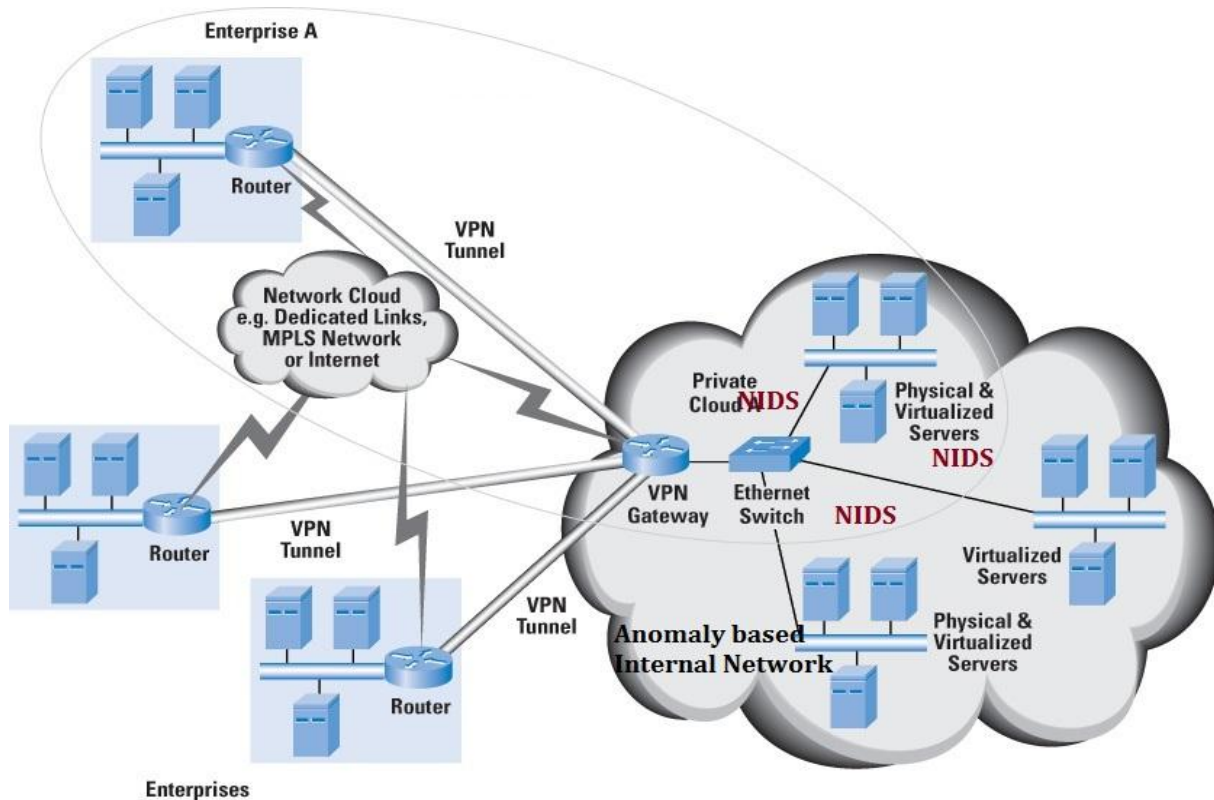


Figure 18 Scanning other tenants

4.2.8 Hypervisor attacks

The Virtual Machine Monitor (VMM) or hypervisor is responsible for virtual machines isolation; therefore, if the VMM is compromised, its virtual machines may potentially be compromised as well. The VMM is a low-level software that controls and monitors its virtual machines, so as any traditional software it entails security flaws [53]. Keeping the VMM as simple and small as possible reduces the risk of security vulnerabilities, since it will be to find and fix any vulnerability.

In virtual environments, the offender will try to take hold of virtual machines by compromising the lower layer hypervisor. New vulnerabilities, (such as zero-day vulnerabilities) when found in virtual machines (VM) can attract an offender to gain access to the hypervisor or different VMs of the cloud. The zero-day vulnerability can be exploited within the application virtualization HyperVM that resulted with the eventual destruction of many websites supported to various the virtual servers. [33]

Having analyzed the Hypervisor based detection systems in chapter 1.7, we are with a no-brainer choice here, as having a specific technology to solve a specific problem is the methodology used, but still it will be show with our own association rules, as we have done to all other risks.

- For IDS type → Hypervisor-based (being specific for this risk, specialized and especially designed for it)
- Technique used → Anomaly detection (ANN for less false positives), as this is how the hypervisor system works.
- Positioning → Internal Network, as there the Hypervisor exists.

IDS Type	Hypervisor-based
Technique used	Anomaly Detection
Positioning	Hypervisor Administration.
Pros	Specialised solution
Cons	Not still widely tested, as it is newer than the other methods.

Table 10 Hypervisor attacks

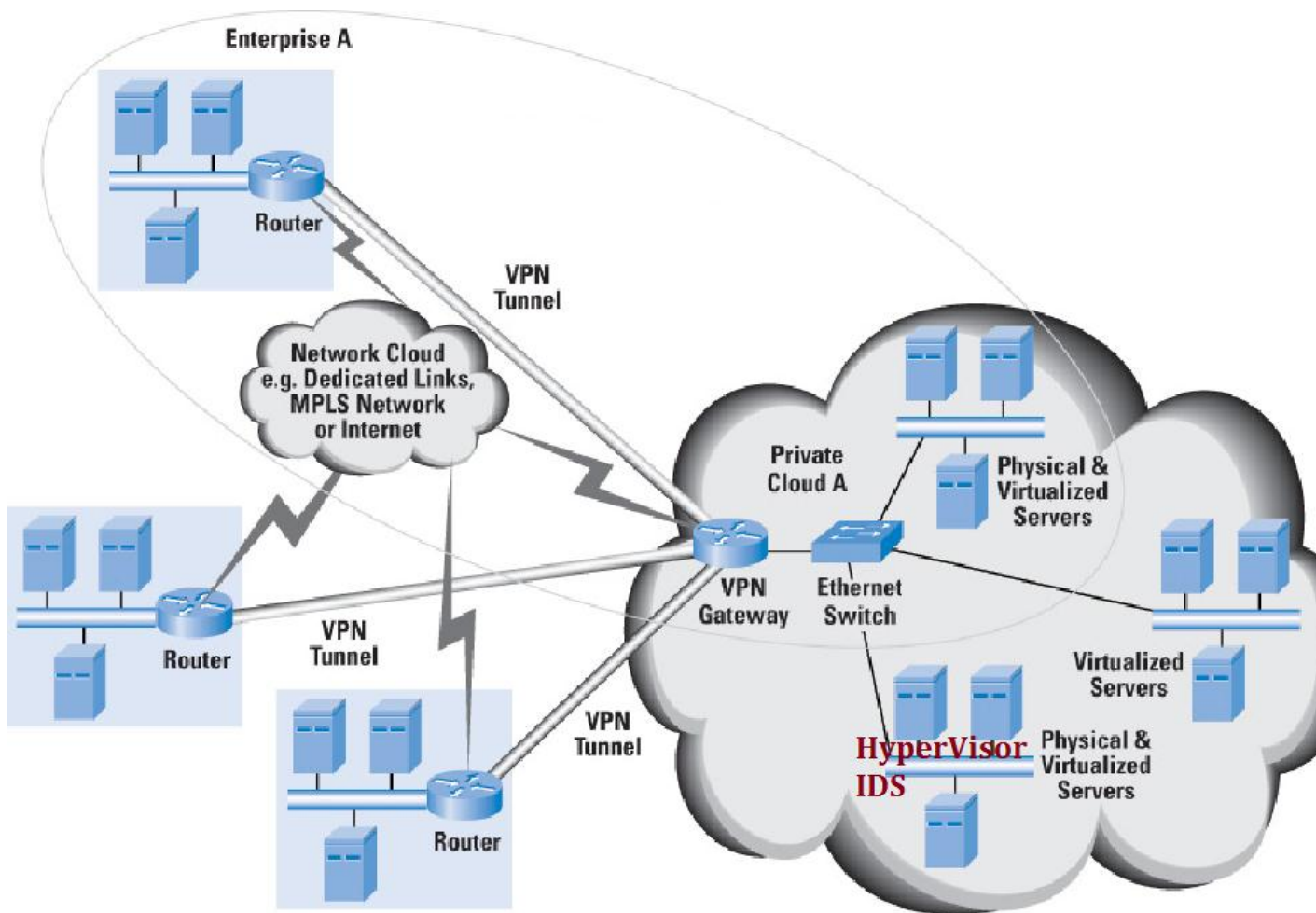


Figure 19 Hypervisor attacks

4.3 Incidence Analysis and Forensics

Here we start with the second selected OWASP category, in which we found vulnerabilities relative to the IDS technology cloud vulnerabilities. These include Malware detection, Intrusion detection response, Honeypot and Logging. In this chapter, we will follow the same methodology in order to find the best IDS solution for each one of these vulnerabilities.

4.3.1 Malware detection

Malware, short name for malicious software, is any software used to disrupt computer or mobile operations, gather sensitive information, gain access to private computer systems, or display unwanted advertising. [60]

Anomaly-based intrusion detection systems were primarily introduced to detect unknown attacks, in part due to the rapid development of malware which by definition guides us to use it. However, the presence of a signature based data base is necessary especially the training period of the IDS. So, combining them is a necessity.

The vast spread of malware, and their various categories, guides us to use a hybrid model. For hosts IDS with signatures, and NIDS with behavioral analysis. The reason is not to overload the network with even more rules, to check the signatures, and leave that to whichever host is targeted, and by this way not delaying the whole infrastructure. Moreover, to cover all cases, the NIDS will have the overview and provide not just a back-up, but a partner to the HIDS.

- For IDS type → Hybrid, due to the nature of the attack, as many types exist and neither system can handle it alone. The combination is needed.
- Technique used → Signature Based / Anomaly Detection. Keeping the duality here, need both to stay versatile.
- Positioning → Every physical machine (Host) / Internal Network, as the target is not known, the HIDS can be in the physical machines and the NIDS in the Internal network.

IDS Type	Hybrid
Technique used	Signature Based / Anomaly Detection
Positioning	Every physical machine (Host) / Internal Network
Pros	Using both gives the needed wider approach to the problem
Cons	Unwanted complexity in the design

Table 11 Malware detection

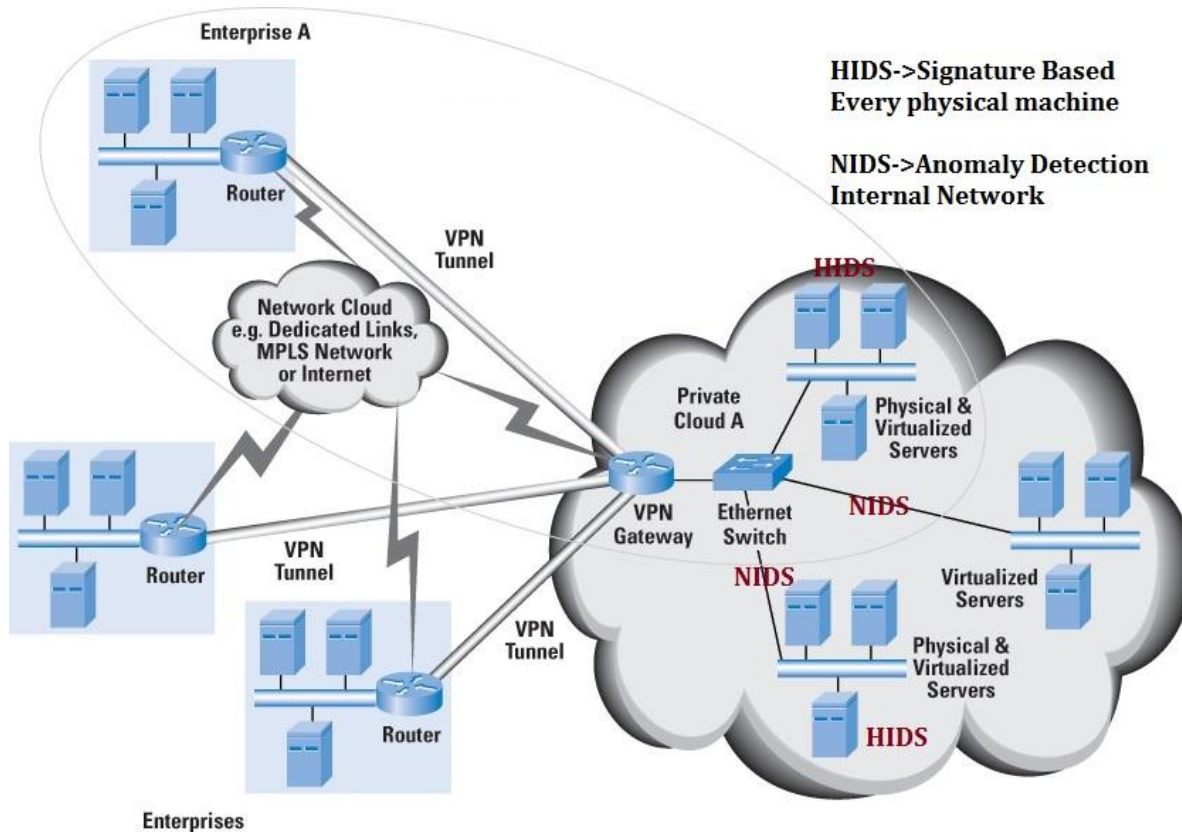


Figure 20 Malware detection

4.3.2 Intrusion detection response

Here we have the implementation of a response plan to the IDS with both automated and human-guided interactions. A direct warning that is sent to the administrator in charge can prove very beneficial. This can lead to direct human actions towards an imminent threat. Even in a system with the best automated procedures, providing the data related to the attack to the human in charge, is most of the times the best solution. This is more important in the case of IDS use, as their sole purpose is to detect and inform and not take massive preventive actions, to mitigate the threat and dangers.

The response policy has to be enforced by the provider, by the management of the organization that owns it, in order to ensure the business continuity, and the wellbeing of the tenants' data and procedures. Having established that, the organization policy, meaning its associated and relevant guidelines, can be inserted as parameters in the IDS to help achieve these desired goals.

Here we don't have a new IDS design, just further rules which effect of course the overall proposed IDS architecture.

4.3.3 Honeypot

A honeypot is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts of unauthorized use of information systems. Generally, a honeypot consists of data (for example, in a network site) that appears to be a legitimate part of the site but it is actually isolated and monitored, and that seems to contain information or a resource of value to attackers, which are then blocked. [56]

This small trap, while not by no means an IDS, can be a part of the proposed implementation, just as an added bonus to help the forensics in case needed. It is not something extremely difficult to implement, various technologies exist such as honeynets. Honeynets are made up form two or more honeypots, in the sense of a network and their use in IDS systems is common. [57]

Honeypots with the addition of the proper logging can provide large datasets of information for further analysis for any forensic investigation. So, the benefits of adding the honeypot to the proposed Intrusion Detection System are many, while it doesn't have a complex implementation.

Here we don't have a new IDS design, just adding the Honeypot which effect of course the overall proposed IDS architecture.

4.3.4 Logging

The proper logging of data will provide valuable information in case of an attack, successful or not. It will provide new knowledge sets to the administrators, the authorities that might be involved and the community in general. While this is not the primary reason for the existence of the IDS, their ability to do so, can prove to be quite a valuable tool for the mentioned reason.

However, some precaution has to be taken. First where the logs are stored, as they can be a target of the attack too. The logs can't be infinite, even in the cloud ecosystem, some rules about the collection process of them have to be made, also about their back-up, and this falls into the organization's internal security policy. Also, the system has to be coordinated, in timely manners, global time presumably, to be able to compare results from different sources. This will give a better understanding of the events.

Here we don't have a new IDS design, just further rules for the logs, where they are kept and how they are transmitted / communicated, which effect of course the overall proposed IDS architecture.

4.4 Infrastructure Security

4.4.1 Internet Dependency

Being an in a sense, Internet technology, the Cloud is depended on the Internet, (in the most implementations), and this dependency from it also creates certain problems. The internet wasn't created having security in mind, and so many of its structural components and protocols, have several security issues. Some from the already covered vulnerabilities were created because of such problems. This is not the issue of this paragraph though, as these issues were comforted in their respective ones.

There has been already stated that the data handled traverse through the internet between end users and cloud data centers. While interception of data in transit should be of concern to every organization, the risk is much greater when organizations utilize a Cloud computing model, where data is transmitted over the Internet. Even if the issue is not the interception, there is always the case of a malicious outsider trying to attack the cloud system.

With the use of cryptography the incoming packets, may fool certain IDS technologies, and pass as secure ones. This makes these technologies inadequate for doing the needed work, or at least it shows that they have been installed improperly or they have inserted incorrectly to the system, or they have been misplaced.

An example is the sole use of an NIDS system, signature based, on the internet nodes of the cloud. Having only this, encrypted malicious packets can traverse through them and enter the eco-system, and allow the attacker to achieve his goals. There won't be a system proposed here as a sole or better solution, or a combination of them, just needed to clarify that some implementation of IDS, just can't work on their own, and vulnerabilities like these point it out. Here we don't have a new IDS design, just making some clarifications for the overall system which effect of course the overall proposed IDS architecture.

4.4.2 Active Unused Ports/Port Scanning

An attack that identifies open, closed and filtered ports on a system in cloud environment [33]

In port scanning, intruders will seize data with the assistance of open ports like services that run on a system, IP and MAC addresses that belong to an association, and/or router, entry and firewall rules. within the Cloud System. The attacker will attack the services obtainable through the scanning of ports (discovering open ports on which can be exploited) [47]

Being a common, widespread attack, all major manufacturers provide rules against it. Especially regarding against the best known and widely spread tools such as N-map. Generally, the attack relies on SYN packets, so the proposed IDS should be able to

detect SYN packets as a very common scan technique. To overcome the problem, of just checking the main attacks from the known programs, either a more generalist approach towards the rule-set can be imported, either an anomaly detection policy.

Being a network attack the NIDS must be in place, in the network nodes. Most attackers have learned to fool the signatures even with the tool Nmap [58], so a behavior-wise method is better, and it will be easy to train, due to the large data presented to process, and the divergence of the normal expected behavior and that of the port scanning.

- For IDS type → NIDS, being a network issue.
- Technique used → Anomaly Detection, while signature detection works, most malicious attempts try to fool the known signatures. However as being a popular attack, there is need for constant updating.
- Positioning → Every network node, in order not to overwhelm the system with placing in in every VM, but still can inspect effectively the network

IDS Type	NIDS
Technique used	Anomaly Detection
Positioning	Each network node
Pros	less number of false positives
Cons	need to be updated frequently

Table 12 Active Unused Ports/Port Scanning

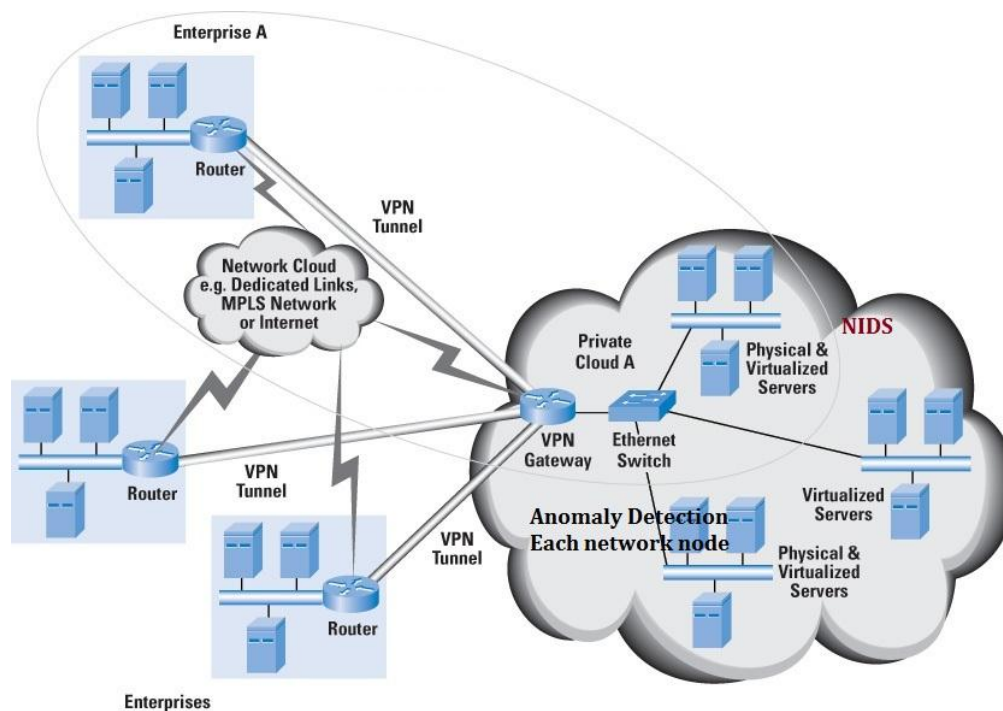


Figure 21 Active Unused Ports/Port Scanning

4.5 The Proposed model

Having the whole case study, it is time to sum up the technologies and get the model in use.

First a table to gather all the info.

Vulnerability	IDS Type	Technique used	Positioning
Denial of Service (DoS) (a)	NIDS	Signature based detection	Every Virtual Machine
Denial of Service (DoS) (b)	NIDS	Signature based detection/ Anomaly Detection (Hybrid approach)	Network perimeter
Cloud Malware Injection Attack	DIDS	Anomaly Detection	Every Virtual Machine
Side Channel Attacks	DIDS	Anomaly Detection	Every Virtual Machine
Authentication Attacks	HIDS	Anomaly detection (ANN for less false positives)	Each network node
Man-In-The-Middle Cryptographic Attacks	HIDS	Signature Based	Each network node
Malicious Insiders (a)	NIDS	Anomaly detection (ANN for less false positives)	Internal Network
Malicious Insiders (b)	HIDS	Signature	Every physical machine (Host)
Scanning other tenants	NIDS	Anomaly detection (ANN for less false positives)	Internal Network
Hypervisor attacks	Hypervisor-based	Anomaly Detection	Hypervisor Administration.
Malware detection (a)	HIDS	Signature Based	Every physical machine (Host)
Malware detection (b)	NIDS	Anomaly Detection	Internal Network
Port Scanning	NIDS	Anomaly Detection	Each network node
ARP spoofing	NIDS	Anomaly Detection	Each network node [95]

Table 13 Full Table Combination

To sum the up, we will first study them by the positioning. So, in the:

- Every Virtual Machine we have 3 hits. 2 with DIDS “Anomaly Detection” and one NIDS “Signature based detection”.
- Network perimeter, we have 1 NIDS Signature based detection/ Anomaly Detection (Hybrid approach)
- Each network node, HIDS (Anomaly detection and signature based) and NIDS Anomaly Detection
- Internal network, NIDS with Anomaly Detection 3 times.
- Hypervisor-based with Anomaly Detection

So, the following table is just a compressed version of the big one, with no innovative changes, just added the ones with the same position.

Positioning	IDS Type	Technique
Every Virtual Machine	DIDS (2)/ NIDS/HIDS	Anomaly Detection/ Signature based
Network Perimeter	NIDS	Hybrid approach
Each Network Node	HIDS (2)/NIDS	Hybrid approach/ Anomaly Detection
Internal Network	NIDS (3)	Anomaly Detection
Hypervisor	Hypervisor-based	Anomaly Detection

Table 14 Simplified Proposal table

The next step, is a further simplification of the model. To find combinations that can serve more variations, but without a trade-off for security.

From what we see, at the moment, is that the Hypervisor can’t be combined with anything else. It is something completely different and will stay to the end. So, one line is guaranteed to be

- Positioning → Hypervisor,
- IDS Type → Hypervisor-based,
- Technique → Anomaly Detection

The “Every Virtual Machine” is heavily populated, with DIDS, NIDS, HIDS. Having explained that the DIDS stands for Distributed Intrusion Detection System, which consists of several IDS (E.g. HIDS, NIDS etc.) all of which communicate with each other, there is no reason for the existence of the NIDS and HIDS, as their needed attributes will be contained in the DIDS. For the technique used, will still have the hybrid approach, the

combination of Anomaly Detection and Signature based, as they are both needed for different detections.

- Positioning → Every Virtual Machine
- IDS Type → DIDS
- Technique → Anomaly Detection/ Signature based (Hybrid approach)

The network perimeter is the first line of defense of the network and there we find a single solution present, which doesn't interfere with the rest of the network. So, it stays as it is.

- Positioning → Network Perimeter
- IDS Type → NIDS
- Technique → Anomaly Detection/ Signature based (Hybrid approach)

In the internal network, sniffers should be placed, to provide coverage for an NIDS that protects us.

- Positioning → Internal Network
- IDS Type → NIDS
- Technique → Anomaly Detection

And at last the Network nodes should be handled carefully, as they play an integral role to the architecture of the cloud, and so the system has to have special care for them. Of course, that derived from our analysis too, as we found them crucial for an IDS architecture.

- Positioning → Each Network Node
- IDS Type → HIDS and NIDS
- Technique → Hybrid Approach

So, what we get is:

Positioning	IDS Type	Technique
Every Virtual Machine	DIDS	Anomaly Detection/ Signature based (Hybrid approach)
Network Perimeter	NIDS	Hybrid approach
Each Network Node	HIDS / NIDS	Hybrid approach
Internal Network	NIDS	Anomaly Detection
Hypervisor	Hypervisor-based	Anomaly Detection

Table 15 Final Proposed System

In a cloud system, this looks like:

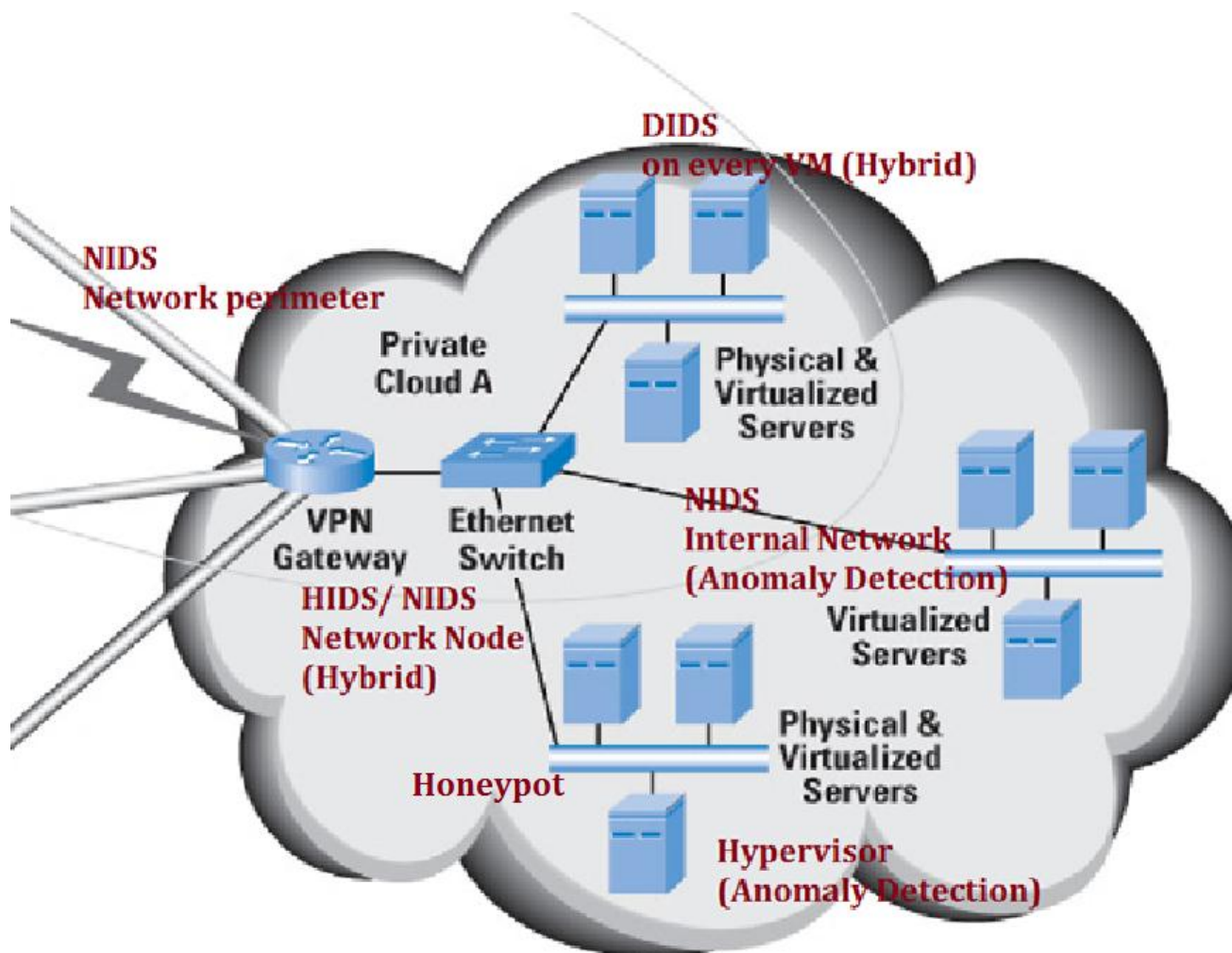


Figure 22 Proposed Model

4.6 Overview and explanation of the Proposed model

We actually can see in the above figure, architecture, where all the native IDS are placed. Even by watching it, we can understand the course of the packets and the inspections happening. Starting from the outer network, a packet must first pass through the network perimeter test, then through the network node. In both areas, a different check is being made. Then it traverses through the internal network, where we have placed also a NIDS. Finally, the VMs are also secured with a Hybrid system. We have also safeguarded the Hypervisor, which is a meaning specific for the cloud environment, as explained earlier.

After studying our proposed system, one might debate if such a solution is viable and achievable. To the non-expert eye, it might seem to be over-populated with small IDS. This is not the case. First of all, we didn't do any trade-offs for security, so this model should achieve the maximum possible efficiency. The presence of more systems, than generally used, doesn't cripple the cloud system. The IDSs are not that computationally demanding to make a difference in such a big system. Especially when we are talking about our network sniffers, which are deemed to be the more lightweight of all. We can say that quite the opposite happens, as with our methodology, we ensured, that the same vulnerability won't be checked by two systems. If two systems checked for it, it would need more computational resources, or would create more network traffic for no reason. With this design, this doesn't happen.

4.7 Benefits of the proposed system

The added partial solutions will make the complete one. A stronger one. One where the weakest link can be easily identified (as it is directly associated with the vulnerability) and thus easily fortified.

In theory, this is as much as perfect system can be, as none of the attack can exploit any vulnerabilities, as the best solutions have been chosen. This is of course, until proven otherwise, but we will see the implementation/ checking practice and methodology in the next chapter. Other prime benefits of the proposed system are:

- Takes advantage of numerous technologies and methods
- Problem Solving Oriented
- Modifiable (can redirect responsibility to another system)
- Upgradeable / Expandable
- Lightweight

And to explain these claims,

Takes advantage of numerous technologies and methods, as the solutions can be implemented with different software. A single bidding selection hasn't been made, at any part of the infrastructure.

Problem Solving Oriented, as this was the model for designing it. Finding the security problems and choosing the right solution. This attitude provides practical solutions and not abstract theoretical models.

Modifiable as we can redirect responsibility from one to another system in case of a problem. Upgradeable as the independent systems can be upgraded, or they can be replaced with better ones. Expandable, as the cloud expands, e.g. with new servers, by placing the right IDS (in VMs and in the internal network for the current example) the model stands, and not major alterations are required to be made.

The modifiable nature also serves other purposes. Not all organizations have the same demands. A Bank, will have many connections, with not that much bandwidth needed for each, but the security will be top concern, while a video streaming service, will need high bandwidth for each connection, and might need slightly less complexity in the security aspect. This is not something we support, as the model goes for the maximum achievable security evaluation, but yet it is understandable in practical commercial implementation.

It is theoretically more lightweight, as the workload is distributed, and no double checking occurs and we have less delay in operation, whether is questionable if the amount of traffic is greater than the average systems. This has to be proven with experiments. A weak link can compromise the system, and so it must be fortified, upgraded, or it's workload to be managed from another IDS of the architecture if possible.

Simplification has been made to the model, both for to physical and logical overlapping of operations, due to the same mechanics, in order for the system to be realistic. To just install everything and everywhere, can happen but it is not realistic as for the costs and the resources. This is not the case here.

4.8 Work Explanation

This was the main part of the present dissertation, and the main contribution, was this chapter. Introducing an innovative work model, with its own work flow, to point out the best individual solutions to the risks vulnerabilities found, to combine them and create a new different system, which doesn't compromise, or makes any trade off, for security.

So apart from the architectural model itself, the way of thought is something to take a close look, as this process can be used efficiently to other aspect of computer technology. With small variations, it can be implemented and find the best solutions, especially in the security aspects as it builds up to the no-tolerance attitude, as it focuses to minimize any threats, by gradually excluding all the vulnerabilities.

The double benefit of the work process is not only the presentation of a complete architecture of a cloud oriented intrusion detection system architecture, which is not reliable to any specific tools, as it can be essentially materialized with many different programs, but a work ethic, flow and mind set.

We came up with a new methodology, and created a new IDS architecture of the cloud systems.

5 Implementation

5.1 Methodology

Until now, the main security problems of the Cloud Environment have been presented. Then the ones that can be solved by an Intrusion Detection System, of any kind, have been pointed out. The appropriate solution for each of these problems, was then brought up, and by combing them a proposed system was presented.

While being a theoretical mode, and the point was to understand the conditions and define the architectural solution, a proof of work must also be given, to observe in by which means this theory is practically applicable. However, the magnitude of a complete IDS built from scratch makes it impossible to concluded in a paper of this kind, and the severity and plethora of attacks and kind, subcategories of them, makes it also impossible to test every one of them.

So, as the building and formatting of the theoretical model was made, the same behavioral evolutionary model will continue to be used here. In order to construct the whole system, small stepping stones, in a sense of building materials will be combined to create the entirety. This means some case examples, the most common attacks, will be examined, and see if the proposed solution stands. An experiment which fails to deliver means that the model can be improved. The successful ones, give the basis, and the evolution of the model to next stage.

Datasets, pack of rules, logs of signatures exist, so an industrialized application of the model would take the benefits of such. But as this is an academic attempt, to realize the model is not the goal in itself, but rather to show the practice of constructing it. As we have already discussed, the model, due to the new attacks, and zero days is an ever evolving one.

Not to generalize any more, the practice to implement the model, and verify it's abilities, it's selecting an attack examined, from the attack chose one of the cases that exist, as it easy to say for example DDOS attack, but this can be accomplished with many methodologies, try to counter, set in motion the part of the IDS responsible at see if it stands. If so the model has succeeded and we proceed to the next one. Being a set of devices, a chain is weak as its weakest link.

An IDS is not something easy to construct and implement as security in general isn't.

So, to view the system testing procedure as a flow chart we find

So, to view the system testing procedure as a flow chart we find

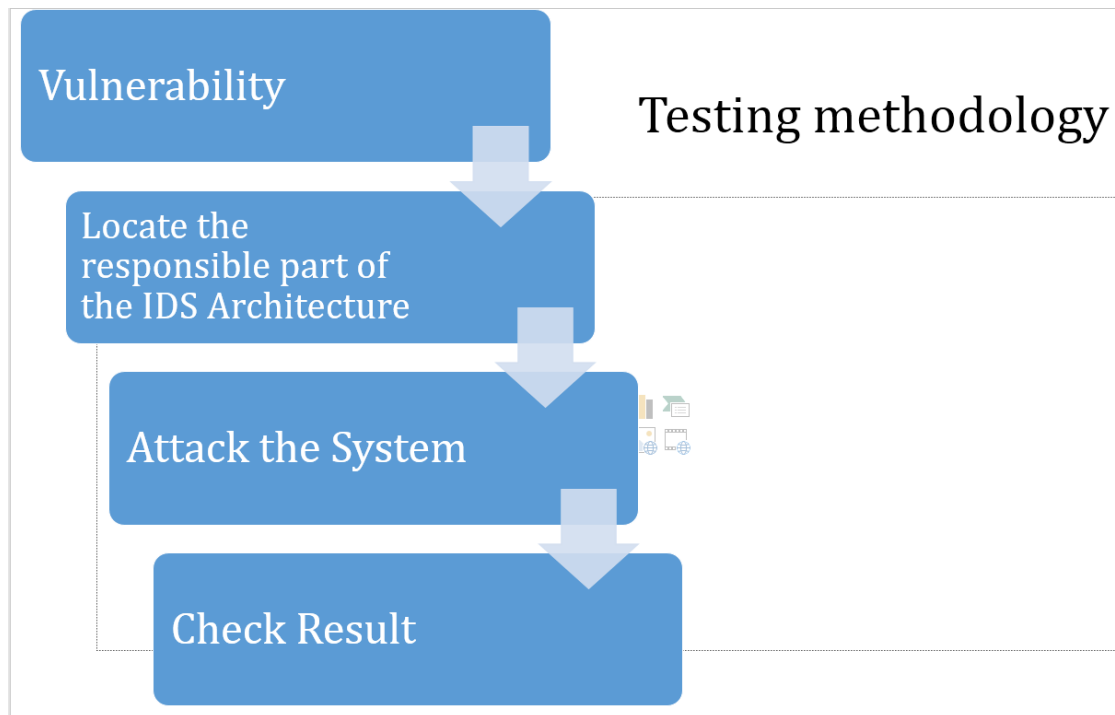


Figure 23 Testing Methodology

5.2 Technologies Selected

Having found the appropriate methods to use for this innovative Intrusion Detection System, next step is to demonstrate that it can actually deliver what is intended too. It's ability to actually respond in the named circumstances. To do so the table of the proposed model has clarified that more than one tools are needed. These are a Host-based IDS, a Network-based, a Hybrid-based and a DIDS. Of course, the whole implementation of the proposed system is not something realizable, as it would need resources far from our reach, neither is the focus and scope of this dissertation. Which is to fabricate such a system, and not it's commercial manufacture. However, the need to demonstrate, even in part it's capabilities is eminent.

For this purpose, certain tools were selected, to conduct use cases, such as those already analyzed. Successful tests will show the viability of the method proposed. A search was made to select open-source and thus free solutions for the realization of the project.

5.2.1 Security Onion

First of all, the system will need a working level, a working station, to make the experiments on. This workstation will also need an operating system. The choice for this is "Security Onion", which is a Linux Distribution specifically designed to aid

users for Network Security Monitoring and Intrusion Detection. It by itself has several tools pre-installed such as Snort, Suricata, Bro, OSSEC, Sguil, Squert, ELSA some of them will put in use. [62]

In order to use it, we haven't directly installed to it to a machine but rather an image of it in a Virtual Box. During the first phase of the installation a snapshot, instance of the installation was captured just in case a roll-back is needed. For the Virtual Machine the oracle virtual box was used, also a freeware software. [63]

5.2.2 Snort

The Security Onion platform already offers the choice of the Snort NIDS. Other options can be used like Suricata, which also is already installed, but Snort being the older one, better known, and probably being the most deployed IDS in the world, makes the choice. [64]

Snort was made and maintained by SourceFire, which now belongs to Cisco, and in its engine, boasts to combine the benefits of all the analyzed technics such as signatures based and anomaly-based, but in its core, it's heavily depended in the signature database it has. Even Security Onion categorizes it as "rule-driven". It performs real-time traffic analysis and packet logging on IP networks, while it also does content searching of the involving packets. [65] It's basic documentation includes "Writing Snort Rules" by Martin Roesch maintained by the Snort Team [66]. Snort entered InfoWorld's Open Source Hall of Fame as one of the "greatest [pieces of] open source software of all time" [67]

5.2.3 OSSEC

OSSEC, which is also included in the Security Onion pack, is a Host-based Intrusion Detection System, which are also needed for the proposed architecture. It is open-source, and it includes an analysis engine, which integrates the logs to do the event analysis, it can conclude file integrity checking, and enforce global rules. It is eligible for rootkit detection while it combines real-time alerting and active response. It has an average of 5.000 downloads per month. [68]

5.2.4 System

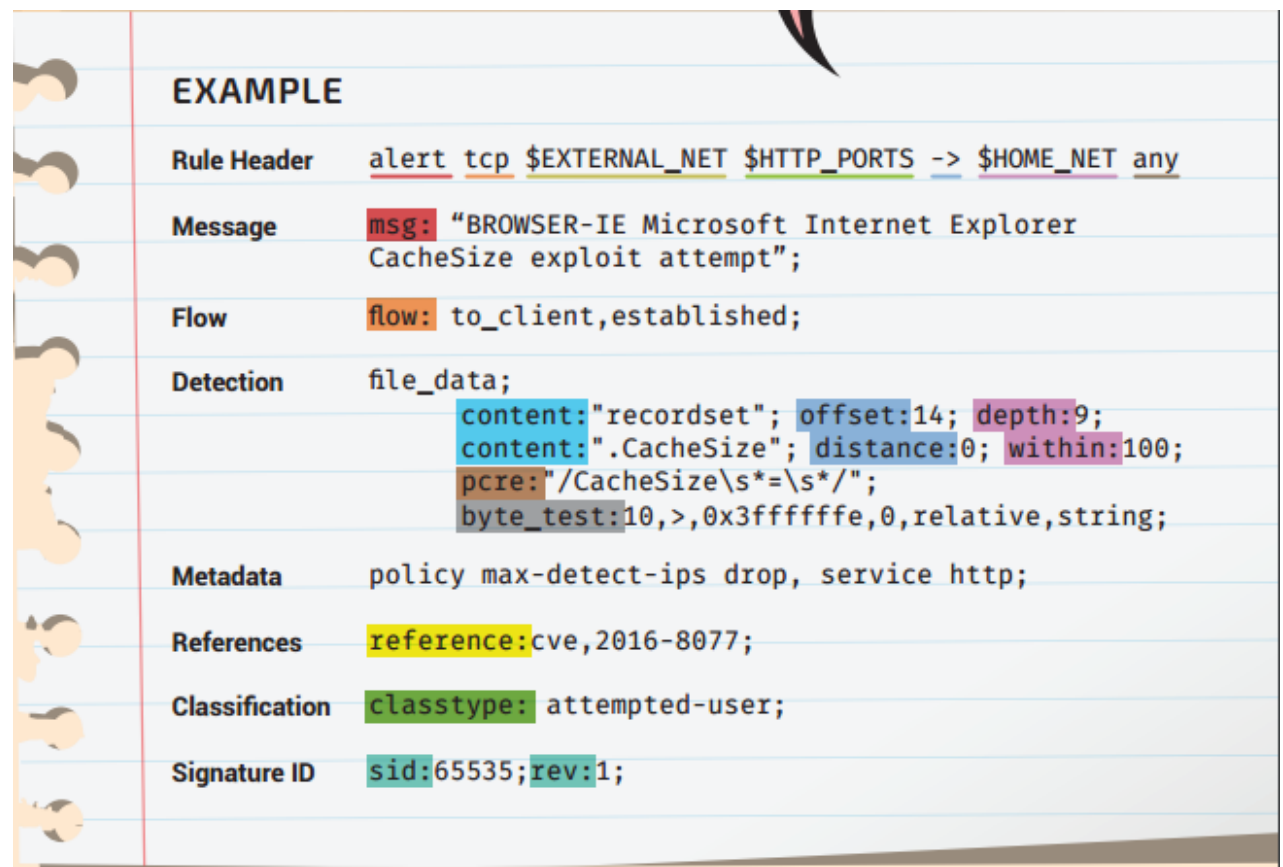
To be more clear of our implementation, we are going to install selected IDSs that suit out model, in the Linux Distribution Security Onion. These will be open source. For the NIDS cases will go with Snort. For HIDS we will go with Ossec. There is no open source hypervisor ID System that can match our needs, so by default there is now way to fulfil the entire system. Will have some examples of the testing of the model, as for it to be implemented and tested for all attacks, by a single person would take several months. Especially taking under consideration the training of the algorithms needed for the entire proposal. These steps will be explained in the future work, as well as the test in a real cloud environment.

5.3 NIDS Issues

In this section, we will experiment and check whether the NIDS issues pointed out in chapter 4 can be successfully be addressed by the NIDS of our choice.

Regarding the Snort Signatures, we will use, and create we find a mini tutorial in their site. After understanding the syntax, there is the need to apply it, cover the problem in mind.

Taken from the Snort site



EXAMPLE	
Rule Header	<code>alert tcp \$EXTERNAL_NET \$HTTP_PORTS -> \$HOME_NET any</code>
Message	<code>msg: "BROWSER-IE Microsoft Internet Explorer CacheSize exploit attempt";</code>
Flow	<code>flow: to_client,established;</code>
Detection	<code>file_data; content:"recordset"; offset:14; depth:9; content:".CacheSize"; distance:0; within:100; pcre:"/CacheSize\s*=\s*/"; byte_test:10,>,0x3ffffffe,0,relative,string;</code>
Metadata	<code>policy max-detect-ips drop, service http;</code>
References	<code>reference:cve,2016-8077;</code>
Classification	<code>classtype: attempted-user;</code>
Signature ID	<code>sid:65535;rev:1;</code>

Figure 24 Snort Example [69]

We see the typical syntax of the rules to use.

At the start, we have the rule header, every rule starts with it, no matter how short or how long the rule will have always the header and the body, which is the intro to the rule. In this case, we have the “alert” tag which means that in the case of the rule being enabled, an alert will be generated, and it is the most common to find in the rulesets. The alert is directed to the moderator of the system. Other options are log where the event is recorder, added to a file, dynamic and activate. The “tcp” that follows is the protocol to be inspected, while other than tcp we could have udp, ip or icmp. Then the source of the message comes (here the external network, (concerning the http ports)) with the data heading to the home network. There isn’t a need to name

external or internal network, a range of IPs can also be given as a parameter, and the arrows that declare direction can go vice versa. At last the “any” declares the destinations’ ports in concern. They can be declared as “any”, protocol specific (as in http_ports) or by their numeric values e.g. “80”.

Then we go to the body of the rule. First, we have the message, which is the explanation, the title of the concerning event. It’s used to identify the content, the vulnerability in a descriptive way. What the rule is designed to detect is indicated here.

Flow helps especially for performance gain, as it rules out alerts, that would use resources or unneeded checking of the data. In this case, the check is for an established connection, and regards the packet arriving to the client, and the rule will not be applied to similar connections, or flows that don’t exactly match this. The option of flowbits also exists to perform detection within flows.

The part Detection is obviously the most important one. It searches inside the file data (to explain here, searches and inspects with the packet itself) for specific content. It can come either as a binary expressed in hexadecimal, or can be mixed text, the offset in for far deep within the bit set the search for matches should start, as does distance, whereas within/depth specifies how far forward the search should be done. Pcre is an identifier for perl to make more complex identifications, and finally the byte_test allows the rule to test a number of bytes against the specific value (in binary).

The low probability of similarity in consecutive bits is the point these techniques rely on. The more the bits checked the less the possibility for a false positive alarm, however this can cripple the system if there is an exaggeration.

Of course, in content as it is the main function more parameters exist as the negative “!” , as in “find this” and!“ but not this” (as “!” negates the match), no case, lower case options, or “keyword” “argument” where they apply only to the keyword they immediately follow.

Reference directs to an external source, and adds more content to the rule. By default, it supports common resources such as bugtraq and nessus, while they can be customized to our needs. Classification explain what “type” the rule is. Predefined classifications exist examples can be like here “attempted-user” breach, or “malware-cnc”, “web-application-attack” etc. The signature id is a unique identifier, to separate the rule from the other, and rev is the numbers the rule has been revised.

5.3.1 DDOS

In Chapter 4.2.1 the case of the DDOS attacks was pointed out. The HIDS was proposed to solve this problem. The case is now whether Snort can solve this. Of course, DDOS attacks can be made with various ways, just analyzing that in the

needed depth can be a dissertation of its own, so as will happen and later some key examples will be given and checked.

A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic. [70] [71]

SYN Flood Attacks are the most common ones and that's why they have been selected to experiment with.

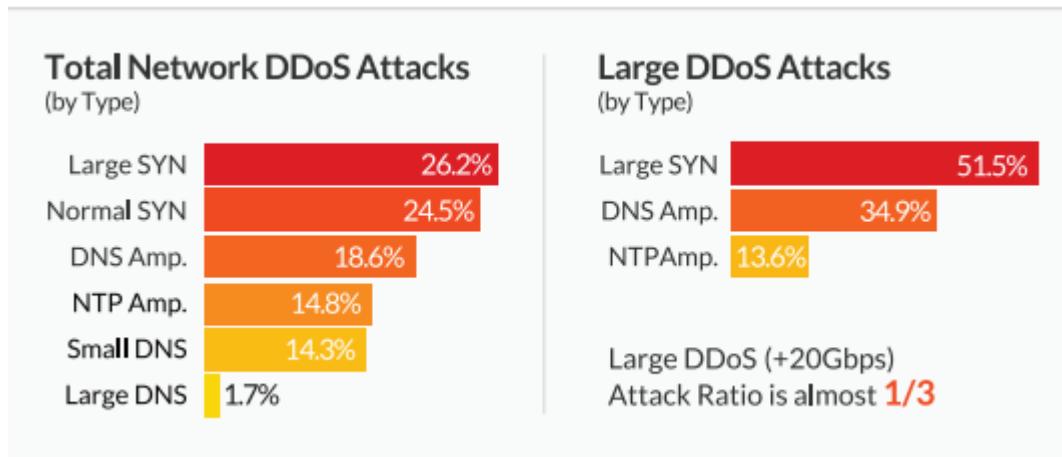


Figure 25 DDoS attacks statistics [76]

SYN being a part of the tcp protocol (as this connection is to be established).

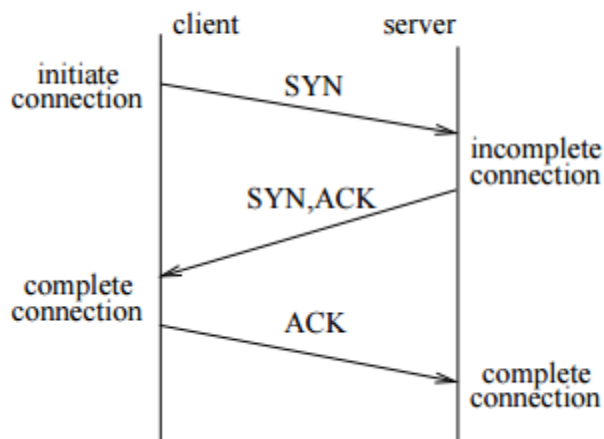


Figure 26 SYN attack [72]

So instead of following this scheme to complete this 3-action verification, the infamous three-way handshake, the client instead of sending the expecting ACK packet, continues with resending SYN (probably after spoofing its' IP address). This

happening in large numbers from a distributed system, a zombie network etc. results to a DDOS attack.

To emulate the SYN flood DDOS attack the tool Hyena is used. [77]

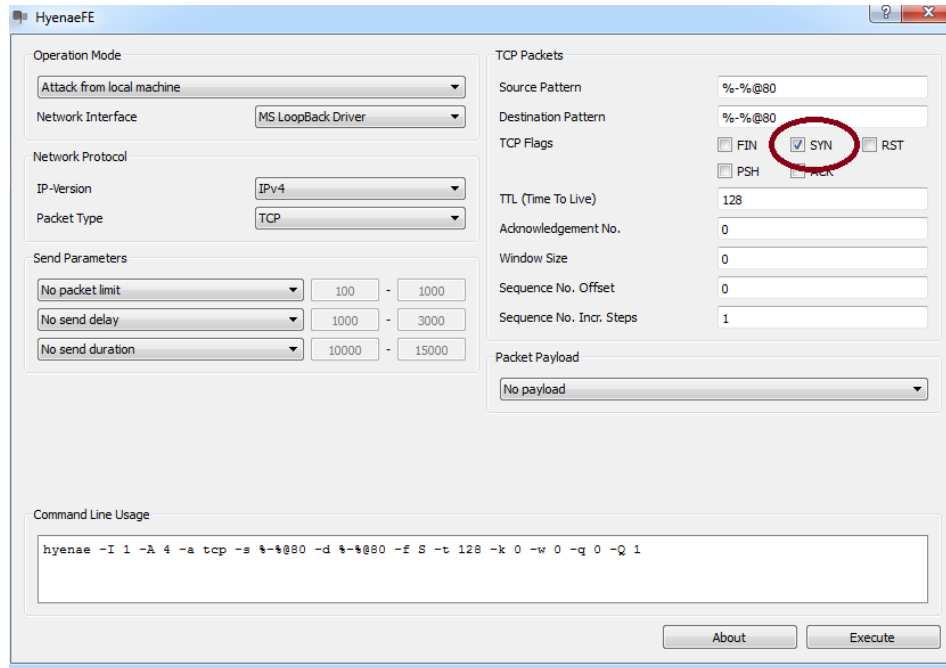


Figure 27 Hyena SYN flood

And after launching the attack for some time,

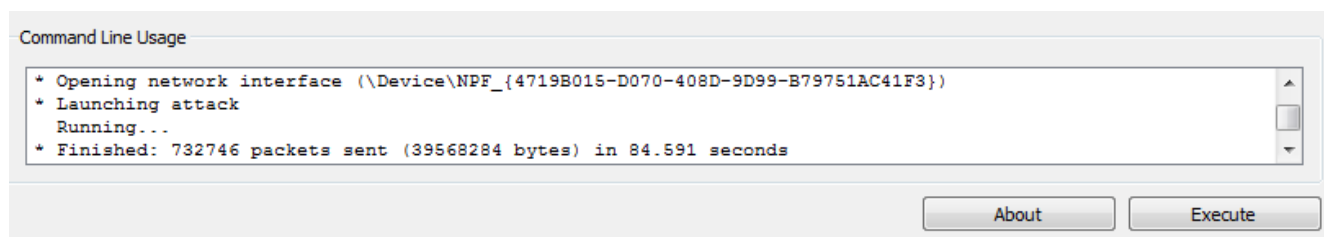


Figure 28 Hyena SYN flood (2)

A rule can be given to send an alert in such a case, which is

Rule

Alert tcp any any -> any any

Msg "Syn Flooding";

Flags:S;

Flow: to_server;

Threshold: type threshold, track by_src, count 10, seconds 1; priority 3;

Sid:9999991; rev1;

[73]

And translates to

An alert given for any tcp connection, either outbound or inbound regardless the port made, will sent the message “Syn Flooding”, and the match options are (the flag which mean TCP : “Syn”) with data flow to server, the threshold line tries also to match, as track by_src means that count is maintained for each unique source IP, count 10 is the number that when reached (10 times SYN respectively from the same source IP) and finally priority 3 to declare the severity of the alert.

A SYN flood attack would seem as the same packet going to the target again and again as shown below

Destination	Protocol	Length	Info
192.168.0.1	TCP	54	10404 → 80 [RST] Seq=1 Win=0 Len=0
192.168.0.1	TCP	54	36076 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	36084 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	12034 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	31889 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	34171 → 80 [RST] Seq=1 Win=0 Len=0
192.168.0.1	TCP	54	36080 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35251 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	3984 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	11971 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	31855 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35182 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	12033 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35181 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35247 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	14463 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	31962 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35259 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35177 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	31961 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35157 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	12020 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35160 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	36102 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35158 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	10644 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35187 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	14409 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	12058 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	12057 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	36098 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	35184 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	14379 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	14399 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	14410 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	14408 → 80 [SYN] Seq=0 Win=16384 Len=0
192.168.0.1	TCP	54	36001 → 80 [SYN] Seq=0 Win=16384 Len=0

Figure 29 Wireshark SYN [83]

What we actually did is to utilize again our work flow as with the input being the DDOS and the SYN flood attack, to see it as a workflow.

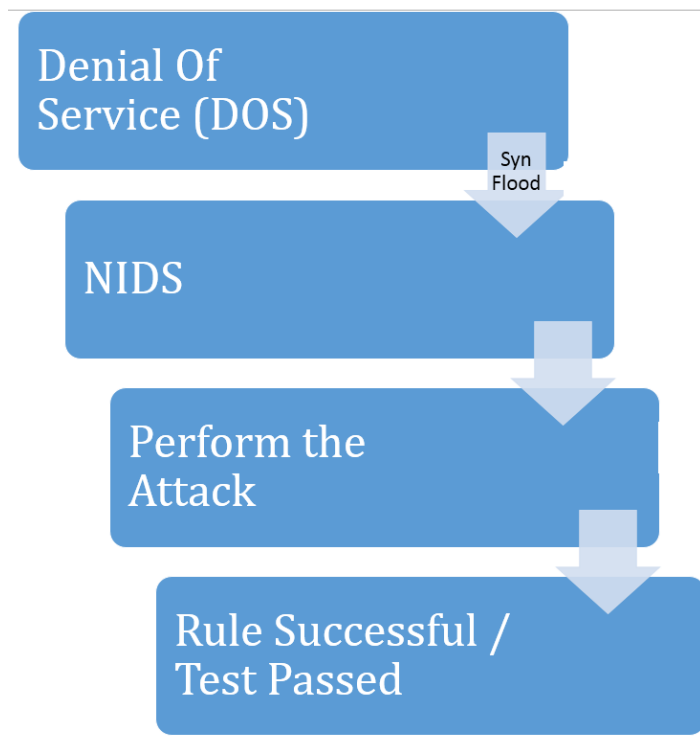


Figure 30 DoS Syn test Workflow

5.3.2 Injection Attack

In the case that an attempt for an SQL injection occurs, by defining the content, in a strict glossary, the NIDS will deliver results if the right trigger is given. For example, when asked to show databases in MySQL

Rule

Alert tcp any any -> any 3306

Msg "MYSQL show databases attempt";

Flow: to_server, established;

Content: "|0F 00 00 00 03|show databases";

Classtype: protocol-command-decode

Sid:9999992; rev1;

[73]

5.3.3 Port Scanning

As we show (in 4.2.1) in order to do a port scan certain tools are needed. The most commonly used one is the Nmap which is open-source freeware. It is used for network discovery and security auditing. It uses raw IP packets to discover available hosts, services, operating systems and many other characteristics. [74] It is widely used, so more probable to be encountered as it holds a top-10 place in the freecode.com [75] repository.

To see the valid of our claims for protection we will perform a port scan on the Virtual machine, through the windows host system.

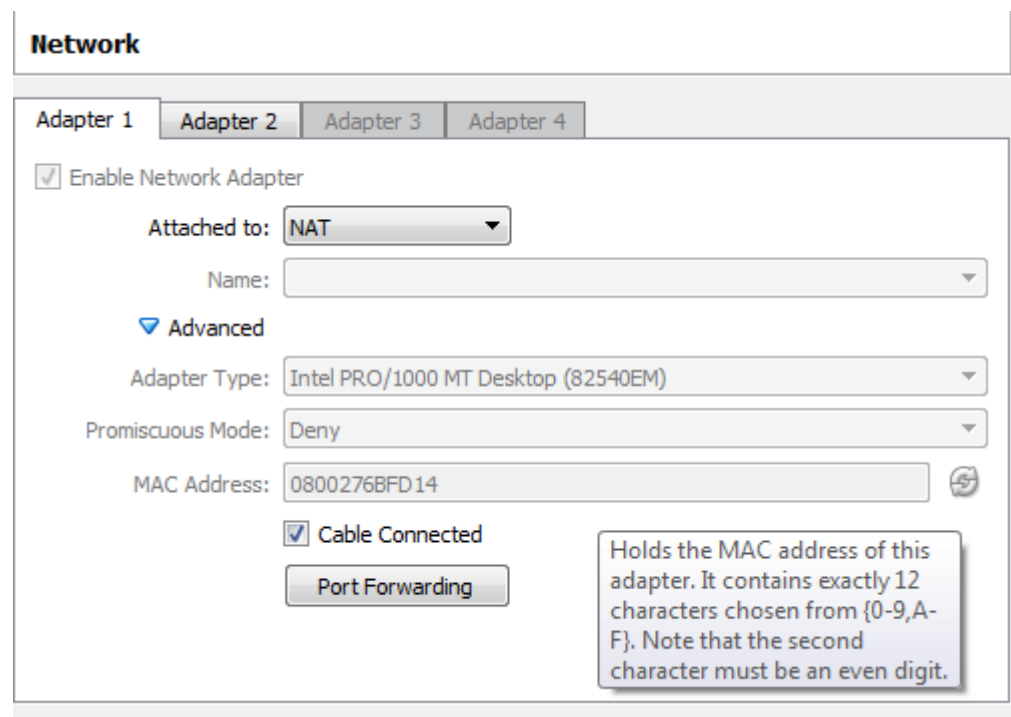


Figure 31 Finding the VM address

We find the “MAC” address of the VM through the virtual box settings,

then correlate it with the arp command, with which we display and modify the Address Resolution Protocol (ARP) cache, and through it can discover the VM’s address, and then conclude the port scan.

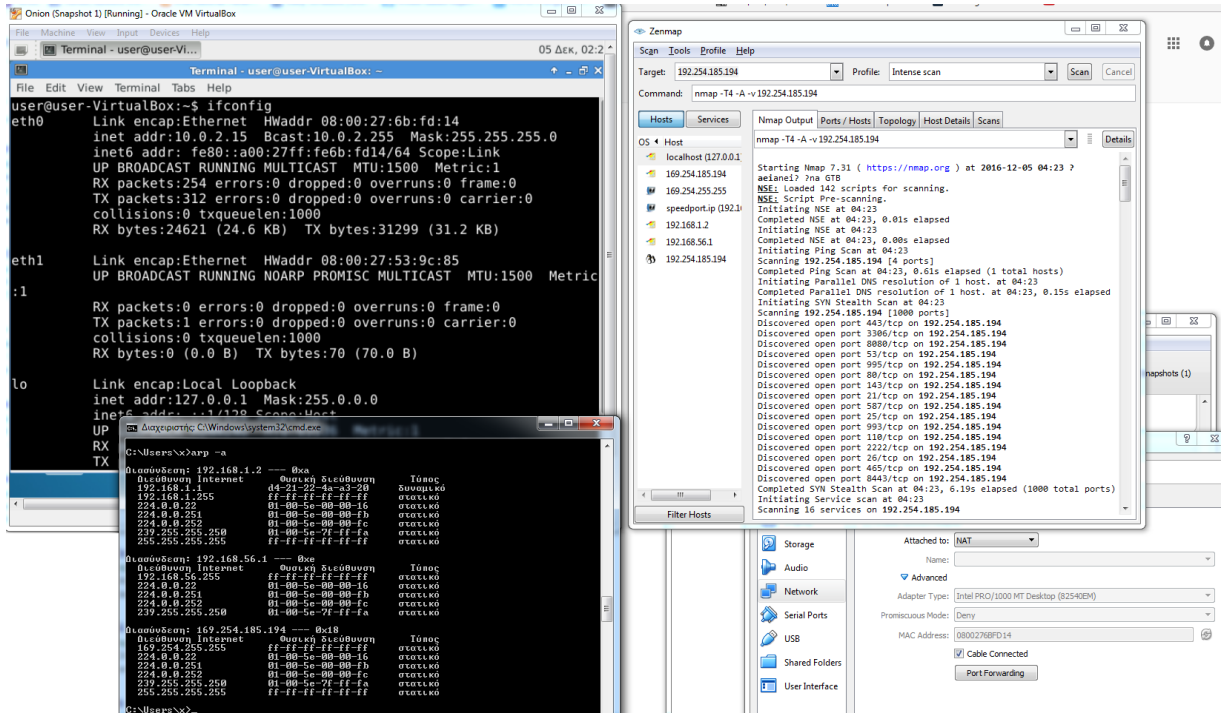


Figure 32 Port scan complete

```

alert tcp $EXTERNAL_NET any -> $HOME_NET any
msg: "SCAN synscan portscan";
id: 39426;
flags: SF;
reference: arachnids,441;
classtype: attempted-recon;
sid:630; rev:1

```

[78]

(a Preprocessor exists to aid , so something like this could stand
Preprocessor flow: stats_interval 0 hash 2, [79] but as we said there many ways to
cover each problem)

```

alert tcp $EXTERNAL_NET any -> $HOME_NET any
msg: "SCAN Nmap ";
flow: stateless;
flags: FPU,12;
reference: arachnids,30;
classtype: attempted-recon;
sid:1228; rev:7;

```

[80]

Here we can use different rules not by whim but because of the fact that Snort specifically has already build-in options regarding the port-scanning incidents, the mentioned preprocessor.

5.4 HIDS Issues

5.4.1 OSSEC

We can see the OSSEC reports through Sguil

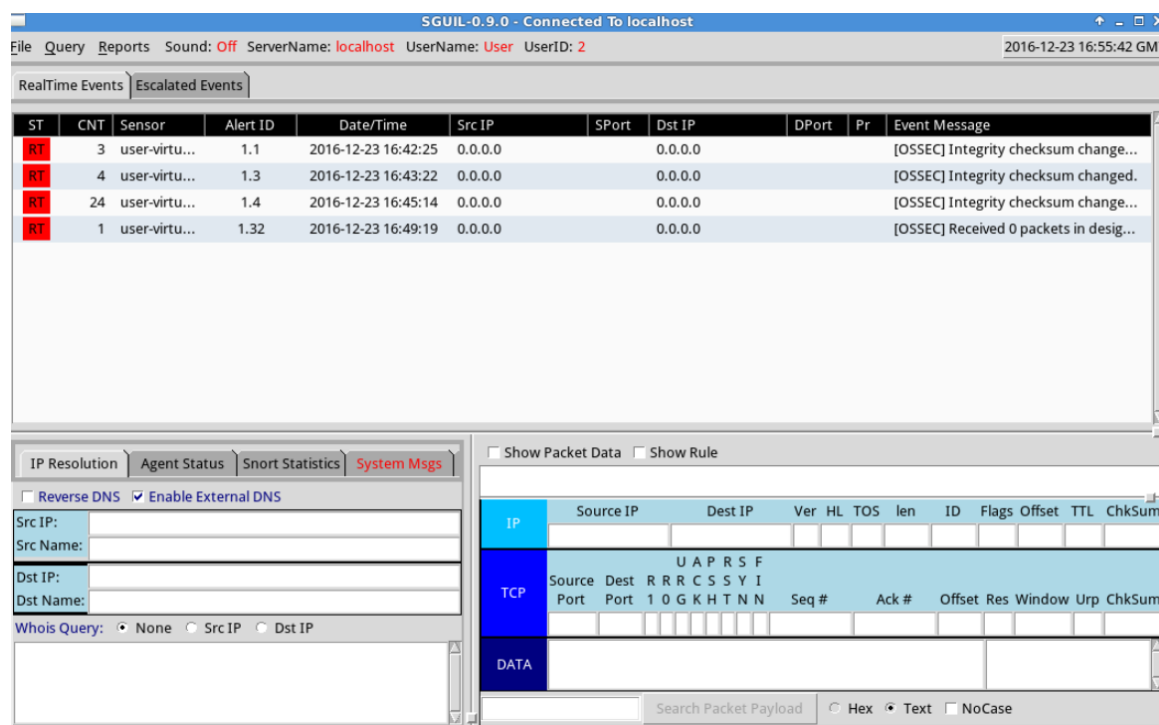


Figure 33 OSSEC Alert

OSSEC being a HIDS monitors the host (our Security Onion instance), we don't have it, as it can't, monitor the network, the previous attacks will go unnoticed. It has alerted the system of integrity checksum change.

Checksum is a technique used to guarantee the integrity of the files. It stores with a predefined method a hash according to the file size. (more advanced techniques of getting the hash exist). As it finds it unexpectedly changed, it generates an alert, with a specific ID. We can monitor these alerts through the Sguil interface. So, tempered files, especially system files from a e.g. malware will be detected this way from the HIDS.

5.5 Future Work

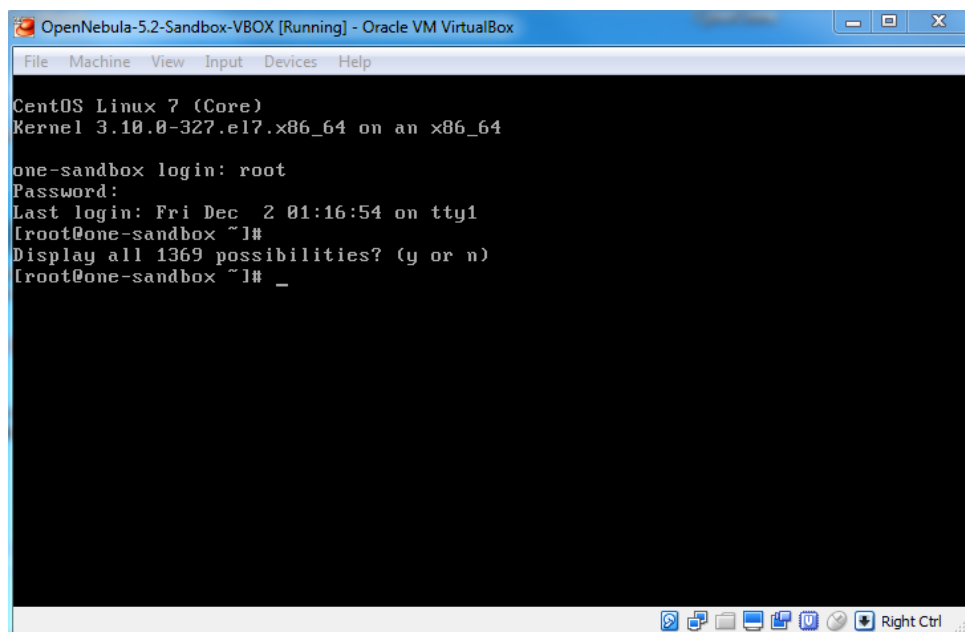
A project this big isn't possible to be implemented in the course of a dissertation process. However, the methodology and techniques are fully demonstrated. By following these steps the whole project can be realized. Instead of making the whole ruleset, common sense dictates the use of the existing libraries. Instead of using datasets, recycled pcap files, or small scale attacks on single hosts another method is proposed for future work and development of the existing idea.

5.5.1 Open Nebula

Open Nebula is a free open source platform, which can manage data center infrastructures. So, at its core is a pure representative of an IaaS model. The user, having the role of the host, administrator and owner of the mentioned infrastructure can provide and build private, public and hybrid implementations of the infrastructure as a service. It is not a simulating program, to replicate the behavior of a cloud environment it is a fully manageable cloud. [81]

It can be installed in the Virtual Box used, and accessed via the browser. There, we can create and manage our own VMs in our cloud, and for instance, dedicate some of them as being used by specific operational systems, having the role of an IDS. A dedicated host, which sole purpose of VM is to have an NIDS installed, can assume the role of and Network NIDS. Then the infrastructure can be assaulted with the attacks we have already studied and see the actual cloud behavior and response.

When the machine is, running nothing is visible in it:

A screenshot of a Virtual Machine terminal window titled "OpenNebula-5.2-Sandbox-VBOX [Running] - Oracle VM VirtualBox". The terminal displays the following text:

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.el7.x86_64 on an x86_64

one-sandbox login: root
Password:
Last login: Fri Dec 2 01:16:54 on tty1
[root@one-sandbox ~]#
Display all 1369 possibilities? (y or n)
[root@one-sandbox ~]# _
```

The terminal window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". At the bottom, there is a taskbar with various icons and a "Right Ctrl" button.

Figure 34 OpenNebula Cloud running

But all the interaction is done through the browser.



Figure 35 Logging in the Cloud (1)



Figure 36 Logging in the Cloud (2)

Where we gain access to the whole platform

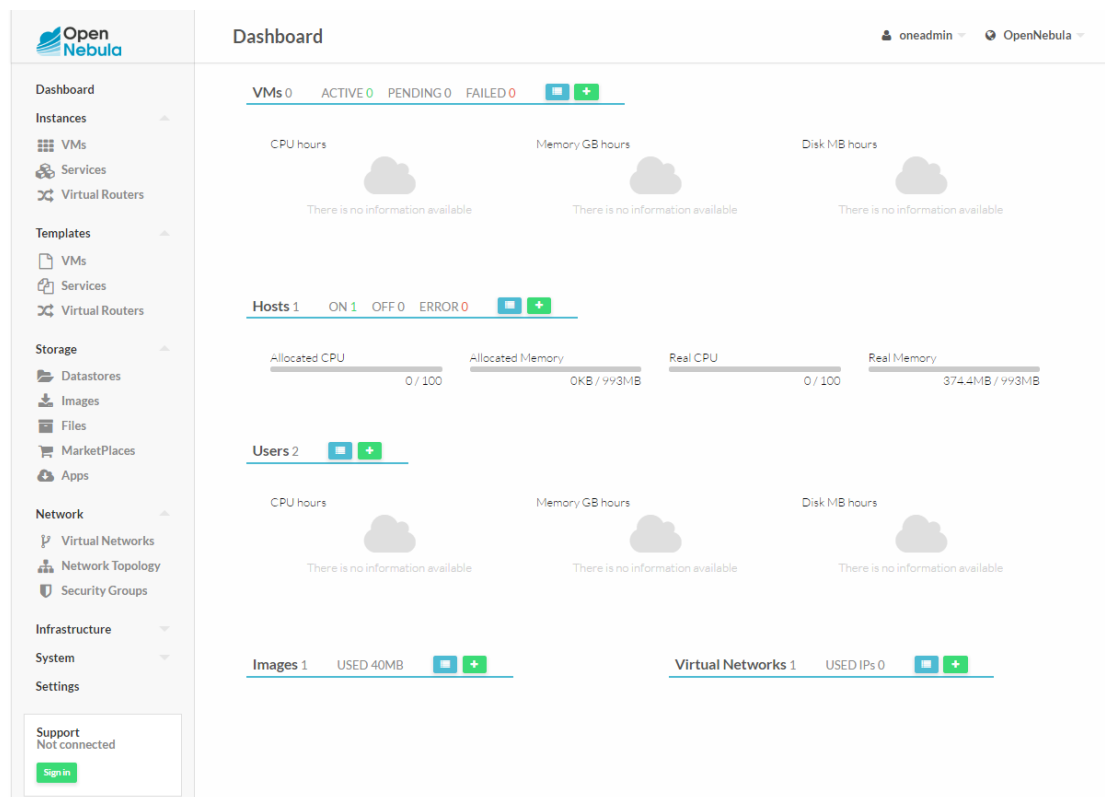


Figure 37 Cloud Control Interface

in order to emulate a cloud environment, as we need traffic generated from hosts, traffic directed to them, and some being the actual targets, the application has a type of mini-Linux. This system is called “ttylinux“ and we can populate the infrastructure with these. Apart from those dummy VMs we can upload, from a predefined installation we made in a Virtual machine, our own operating systems, which can play the role of IDS.

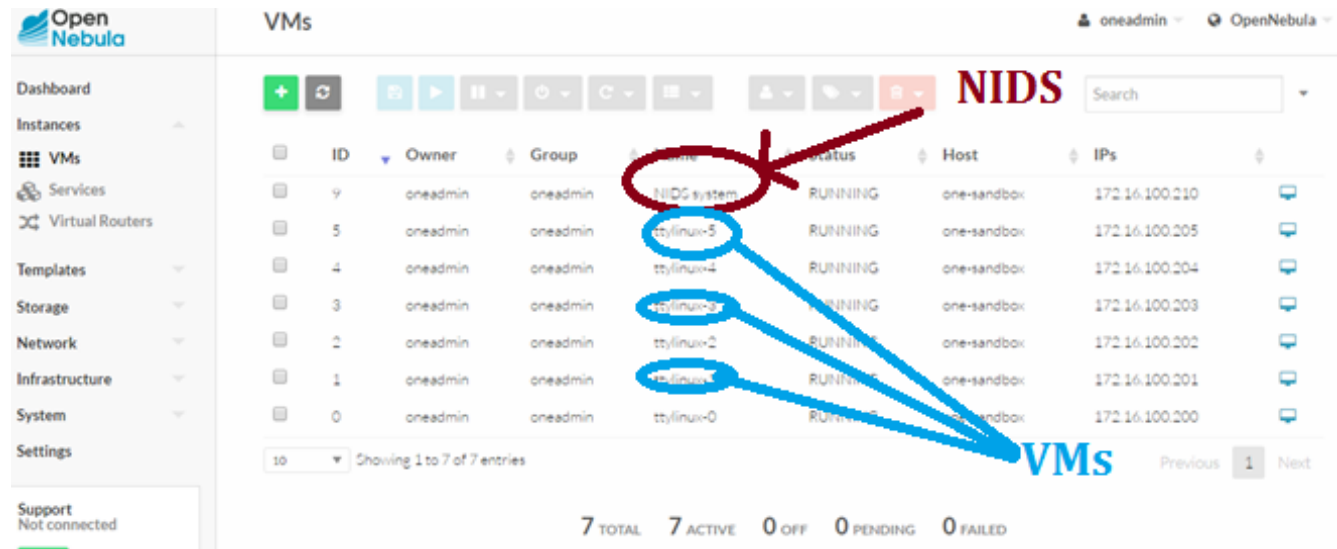


Figure 38 Cloud Contents (VMs)

Here we see 7 hosts, 6 of them being simple workstations (with the ttylinux) and one having the role of a NIDS system. They are all up and running with their won respective IPs.

Or we create more VMs and a new inner network as the picture below, whatever our needs the possibilities are endless, and this platform can provide amazing tools to do so.

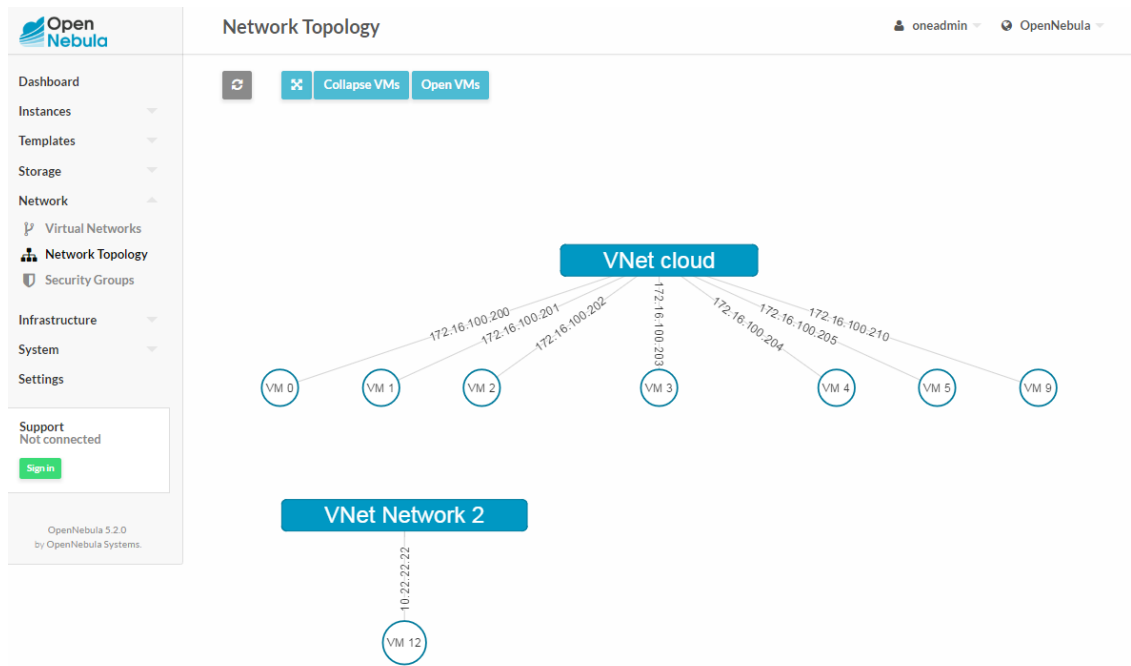


Figure 39 Cloud Internal Network

The natural evolution of this project, and so labeled as future work, would be the implementation of the fully working proposed system in the Open Nebula environment, and in there start to study its behavior by attacking it, being a free and always available platform. It can provide valuable feedback, to improve and find weak links.

However, the Cybersecurity issue is ever changing with new challenges to arise, and in order to maintain top-notch, the administrator of our supposed system should stay ever-vigilant and evolve the model. That being the new signatures update, or even the use of new technologies. This is another benefit of this system. Having small building materials, the tools used, an emerging technology can be easily added, and be compatible. It is not a self-centered environment, but an adaptive system.

Of course, in order to be complete the whole system must be implemented, not just the HIDS and NIDS as we did here. But with the lack of resources, open-source tools and time, we only showed the first step of this procedure. In future work, we would integrate the NIDS and HIDS of the hosts as a distributed system, with the other network NIDSs.

6 Conclusion

6.1 What we learned

With our involvement with this dissertation we gained valuable knowledge and skills. First of all, we gained complete and thorough knowledge of the cloud environment, as we needed to have a complete understanding of it, of how it works, what are its needs, how it is build and designed.

Secondly, we had to study the intrusion detection systems, how they work, what are the benefits of each one, when and where exactly they are applied. Then we studied how they are implemented in the cloud environment, so we have a holistic approach towards the theory regarding these systems.

Then there was the time to study the security issues of the cloud, the risks that affect this technology. We learned to associate these risks with vulnerabilities, and gained vast knowledge regarding the possible attacks.

We proposed our own method, to create the model. A special workflow, to find the best architectural solution. So, we learned to create new problem solving processes, to create new models, a procedure that easily can be altered and used in other cybersecurity problems. Through this work-flow, we created our own IDS architecture for the cloud environment.

Finally, we learned to use popular stand-alone applications, which are used as IDS, created and reviewed rules, learned how to imitate the attacks, and attacked our own system.

So we acquired both theoretical and practical knowledge, about the cloud, the IDSs, the risks, vulnerabilities, attacks and tools.

6.2 What we proposed

We actually did two separate proposals with this dissertation.

The workflow, a procedure to deliver results, guided by the problem-solving mindset which proposes that there is not a global best solution, to mitigate the problem by default. But instead each problem dictates the solution. With the combination (and of course simplification) of all these part-solutions, we got the model we wanted.

By using this proposed work model, we came to our proposal. By using academically acclaimed sources, and our own work, we did the proper combinations and logical assumptions and created a new innovative model for intrusion detection in the cloud environment.

This model, is our own work, created by us, in order to mitigate the weaknesses, the cloud intrusion detection models share. That of overreliance in certain technologies, whereas we welcomed variety.

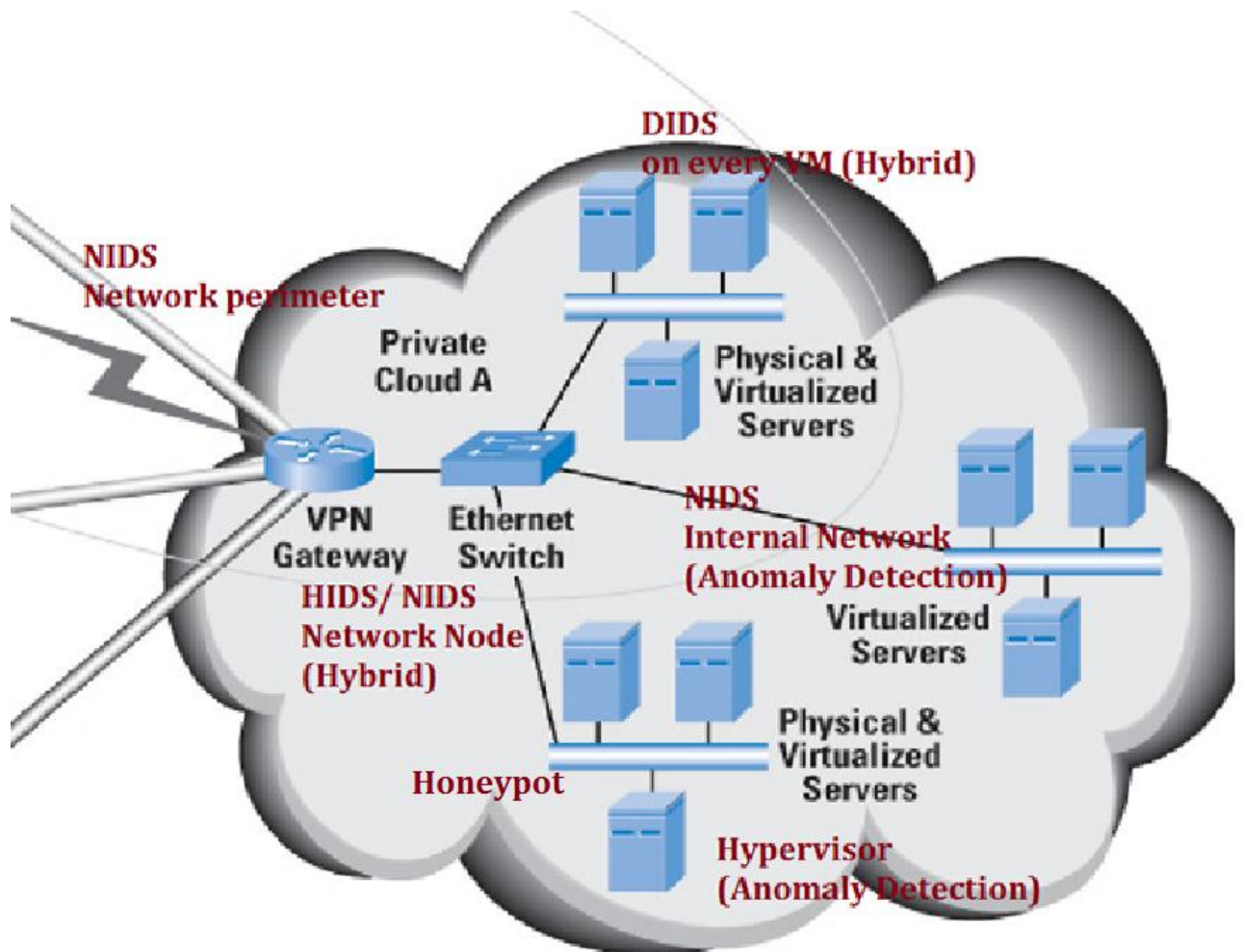


Figure 40 Proposed Cloud IDS Architecture

6.3 Does it work?

In order to experiment with our model, and see if our proposal stands, we emulated certain attacks, and observed whether specific, designated parts of our model stand or not. If they detected the attack it means, that our proposal works, if not we would have a critical failure and would need to re-arrange and re-design certain things. Of course, the project is too vast, and the number of possible attacks is too big for a single person to manage in such short period of time, and we took some examples, mainly to show how the testing of this model must proceed. Perhaps this could be the point of another assignment.

But this is the use of **Corroborating Evidence**, which are evidence that tend to support a proposition that is already supported by some initial evidence, therefore confirming the proposition [82]. We can have countless experiments that show that our model is correct and stands, but one is enough to disprove it. This one hasn't been found yet. But even if it happens, or a new vulnerability and attack method is found, due to the modifiable nature of our model, a solution can be found, by doing alterations to the model.

6.4 The Future

The cloud technology evolves. New vulnerabilities and attacks are constantly found. An article by Cisco (12/2016) made the question “Is Government Regulation the Answer to IoT Security?” [83]. In the article, several arguments showed that this might be indeed the case, as many drawbacks and faults can be corrected. In this manner, I would like to rephrase it as “Is Government Regulation the Answer to Cloud Security?”. Even better, by replacing “Government” with “International regulations”.

The attackers evolve; thus, the attacks evolve, new technologies are found. In this merciless chase, innovative ways of problem solving and thinking as this we presented with this dissertation can provide the cutting edge and be a game changer.

7 References

- [1] The NIST Definition of Cloud Computing
- [2] Roundup of cloud computing forecasts and market estimates – “Forbes.com”
2016 March 2016
- [3] Ben Kepes,
Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS
Rackspace.com white paper
- [4] <http://researchhubs.com/post/computing/cloud-computing/what-is-cloud-computing.html>
- [5] Arshiya Sultana,
Cloud Computing- How much you know of it
Sysfore.com April 28, 2014
- [6] Erick Simpson,
The Cloud Computing Services Primer
SPC International, October 15, 2014
- [7] Abdullah A. Mohamed,
Design Intrusion Detection System Based on Image Block Matching,
International Journal of Computer and Communication Engineering,
IACSIT Press, Vol. 2, No. 5, September 2013.
- [8] Denning, Dorothy E.,
An Intrusion Detection Model
Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986,
pages 119–131
- [9] John R. Vacca
Managing Information Security.
Syngress. p. 137. ISBN 978-1-59749-533-2. Retrieved 29 June 2010.
- [10] Symantec, News Release Symantec Strengthens Security Leadership With
Acquisition of AXENT
<https://www.symantec.com/region/can/eng/press/2000/n000727.html>
- [11] Cisco Catalyst 6500 Series Intrusion Detection System (IDSM-2) Module,
Cisco <http://www.cisco.com/c/en/us/products/interfaces-modules/catalyst-6500-series-intrusion-detection-system-idsm-2-services-module/index.html>
- [12] www.cybersafe.com

[13] www.iss.net

[14]

[15] Wireless Intrusion Detection System

International Journal of Computer Applications (0975 – 8887) Volume 5– No.8,
August 2010

[16] Tiwari Nitin, Solanki Rajdeep Singh, Pandya Gajaraj,
Intrusion Detection and Prevention System (IDPS)

Technology- Network Behavior Analysis System (NBAS)

International Science Congress Association

ISCA Journal of Engineering Sciences ISCA J. Engineering Sci. Vol. 1(1), 51-56,
July (2012)

[17] Newman, Robert C.

Computer Security: Protecting Digital Resources.

Jones & Bartlett Learning. (2009) ISBN 0-7637-5994-5.

[18] Debar, Hervé; Dacier, Marc; Wespi, Andreas

Towards a taxonomy of intrusion-detection systems

Computer Networks. (23 April 1999).31 (8): 805–822.

doi:10.1016/S1389-1286(98)00017-6

[19] David Wagner,

Mimicry Attacks on Host-Based Intrusion Detection Systems

University of California, Berkeley

[20] P. Julia Grace N. Raghya Priya,

A Study on Network Intrusion Detection – Counter Measure using VM Profiling

International Journal of Advanced Research in Computer Science and Software

Engineering, Volume 6, Issue 10, October 2016 ISSN: 2277 128X

[21] Chirag Modi, Dhiren Patel, Hiren Patel, Bhavesh Borisaniya, Avi Patel,

Muttukrishnan Rajarajan,

A survey of intrusion detection techniques in Cloud

Rajarajan Centre for Cyber Security Sciences, City University London

[22] Huseby, Sverre.

Innocent Code: A Security Wake-Up Call for Web Programmers.

Wiley. p. 203. ISBN 0470857447.

[23] https://www.owasp.org/index.php/About_OWASP

[24]

[https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Proje
ct](https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project)

- [25] Federal Standard 1037C: Glossary of Telecommunications Terms
- [26] Tillapart,
Intelligent Handover Decision Based on Fuzzy Logic
- [27] Shankar Babu Chebrolu, Vinay Bansal, Pankaj Telang,
Top 10 Cloud Risks That Will Keep You Awake at Night
- [28] United States Computer Emergency Readiness Team
- [29] Cisco, A Cisco Guide to Defending Against Distributed Denial of Service Attacks
- [30] Ryo Kaizaki, Osamu Nakamura, Jun Murai,
Characteristics of Denial of Service Attacks on Internet Using AGURI,
Volume 2662 of the series Lecture Notes in Computer Science pp 849-857
- [31] An Overview of DDoS Attacks in Cloud Environment,
International Journal of Advanced Networking Applications (IJANA)
ISSN No.: 0975-0290 124
- [32] A.Bakshi, and B. Yogesh,
Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine,
Second International Conference on Communication Software and Networks, 2010,
pp. 260- 264
- [33] Flora S. Tsai, Network Intrusion Detection Using Association Rules,
International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009
Nanyang Technological University
- [34] <https://www.radware.com>
- [35] A Cisco Guide to Defending Against Distributed Denial of Service Attacks
cisco.com
- [36] A.S.Syed Navaz, V.Sangeetha, C.Prabhadevi,
Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud
International Journal of Computer Applications (0975 – 8887) Volume 62– No.15,
January 2013
- [37] John E. Dickerson, Julie A. Dickerson,
Fuzzy Network Profiling for Intrusion Detection
Electrical and Computer Engineering Department Iowa State University
- [38] P. P. Ramgonda and R. R. Mudholkar,
Cloud Market Cogitation and Techniques to Averting SQL Injection for University
Cloud,

International Journal of Computer Technology and Applications, Vol. 3, No. 3, pp. 1217-1224, January, 2012.

[39] Te-Shun Chou, Security threats on cloud computing vulnerabilities, International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, No 3, June 2013

[40] Chu-Hsing Lin, Chen-Yu Lee, Shin-Pin Lai and Wei-Shen Lai, A Semantic Rule-based Detection Scheme against Flooding Attacks on Cloud Environment International Journal of Security and Its Applications Vol. 6, No. 2, April, 2012.

[41] Swapnil Kharche , Jagdish Patil, Kanchan Gohad , Bharti Ambetkar, Preventing sql injection attack using pattern matching algorithm Department of Computer Engineering, Maharashtra institute of technology

[42] N. Gruschka and M. Jensen, Attack surfaces: A taxonomy for attacks on cloud services Proc. - 2010 IEEE 3rd Int. Conf. Cloud Comput. CLOUD 2010, pp. 276–279, 2010.

[43] A. M. Lonea, D. E. Popescu, and H. Tianfield Detecting ddos attacks in cloud computing environment International Journal of Computers, Communications & Control, vol. 8, no. 1, 2013.

[44] I. Khalil, A. Khreishah, and M. Azeem Cloud Computing Security: A Survey Computers, vol. 3, no. 1, pp. 1–35, Feb. 2014

[45] B. Sevak Security against Side Channel Attack in Cloud Computing Int. J. Eng. Adv. Technol., vol. 2, no. 2, pp. 183–186, 2012.

[46] A. Singh and M. Shrivastava Overview of Attacks on Cloud Computing Int. J. Eng. Innov. Technol., vol. 1, no. 4, pp. 321–323, 2012

[47] Damien Riquet, Gilles Grimaud and Michaël Hauspie Study of the impact of the attacks and distributed multi-path on network security solutions MajecSTIC, 2012.

[48] Kazi Zunnurhain and Susan V. Vrbsky, Security Attacks and Solutions in Clouds

[49] Deepa.B, Security attack issues and mitigation techniques in cloud computing environments, International Journal of UbiComp (IJU), Vol.7, No.1, January 2016

- [50] Priyanka Chouhan, Rajendra Singh,
Security Attacks on Cloud Computing with Possible Solution,
International Journal of Advanced Research in Computer Science and Software
Engineering Research Paper, Volume 6, Issue 1, January 2016 ISSN: 2277 128X
- [51] Subashini S, Kavitha V
A survey on Security issues in service delivery models of Cloud Computing.
Netw Comput Appl 34(1):1,2011
- [52] Joint Universities Computer Centre Limited,
Man-in-the-Middle Attack Security and Privacy Concerns.
Information Technology Services, The University of Hong Kong
- [53] Reuben JS,
A survey on virtual machine Security. Seminar on Network Security.
Technical report, Helsinki University of Technology, October 2007
- [54] Nathan Einwechter,
Preventing and Detecting Insider Attacks Using IDS SecurityFocus
- [55] Computer Economics survey
- [56] Naveen, Sharanya.
Honeypot. Retrieved 1 June 2016.
- [57] Ryan Talabis,
Honeynets a Honeynet Definition
PhilippineHoneynet.org
- [58] Cynthia Bailey Lee Chris Roedel Elena Silenok,
Detection and Characterization of Port Scan Attack,
Department of Computer Science & Engineering University of California, San Diego
- [59] Mitchell, Melanie
An Introduction to Genetic Algorithms.
Cambridge MIT Press. ISBN 9780585030944 (1996).
- [60] Malware definition. techterms.com.
- [61] Dempster, A. P
"Upper and lower probabilities induced by a multivalued mapping".
The Annals of Mathematical Statistics. (1967). . 38 (2): 325–339
- [62] <https://securityonion.net/>
- [63] <https://www.virtualbox.org/>
- [64] <https://www.aldeid.com/wiki/Suricata-vs-snort>

- [65] <https://www.snort.org/>
- [66] Snort manual
<http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node2.html>
- [67] Doug Dineley; High Mobley
The Greatest Open Source Software of All Time
- [68] <http://ossec.github.io/about.html>
- [69] Snort rule infographic official from <https://www.snort.org/>
- [70] 1996 Advisory.
TCP SYN Flooding and IP Spoofing Attacks
Software Engineering Institute, Carnegie-Mellon University
- [71] New York Times,
New York's Panix Service Is Crippled by Hacker Attack
September 14, 1996
- [72] Jonathan Lemon,
Resisting SYN flood DoS attacks with a SYN cache,
FreeBSD Project
- [73] Jinsegn Xu, Jinghua Zhang, Triveni Gadipalli. Xiaohong Yuan, Huming Yu,
Learning Snort Rules by Capturing Intrusions in Live Network Traffic Replay
- [74] <https://nmap.org>
- [75] <http://freecode.com/>
- [76] Imperva White paper
The Top 10 DDoS Attack Trends
- [77] <https://sourceforge.net/projects/jhyena/>.
- [78] <http://www.linuxquestions.org/questions/linux-security-4/how-to-detect-nmap-syn-scan-w-snort-331817/>
- [79] Snort Cookbook. O'Reilly.
ISBN: 0-596-00791-4
- [80] <https://github.com/eldondev/Snort/blob/master/rules/scan.rules>
- [81] <https://opennebula.org/>
- [82] Douglas Walton,
Evaluating Corroborative Evidence,
Department of Philosophy, University of Dundee, UK, University of Winnipeg.

- [83] Owen Lystrup,
Is Government Regulation the Answer to IoT Security?
Cisco official. - December 13, 2016
- [84] Mohammed Alsafi , Wafaa Mustafa Abdulllah and Al-Sakib Khan Pathan,
IDPS: An Integrated Intrusion Handling Model for Cloud Computing Environment,
Department of Computer Science Faculty of Information and Communication
Technology International Islamic University Malaysia (IIUM), Malaysia
- [85] Xin W, HTing-lei,and LXiao-yu,
Research on the Intrusion detection mechanism based on cloudcomputing,
In the proceedings of International Conference on Intelligent Computing and
Integrated Systems(ICISS), Guilin,pp.125–8,2010
- [86] Ms. Parag K. Shelke, Ms. Sneha Sontakke, Dr. A. D. Gawande
Intrusion Detection System for Cloud computing
International Journal of Scientific & Technology Research Volume 1, Issue 4, May
2012
- [87] Amirreza Zarrabi, Alireza Zarrab
Internet Intrusion Detection System Service in a Cloud
IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2,
September 2012
- [88] Patel A,
An intrusion detection and prevention system in Cloud computing: A systematic
review
Journal of Network and Computer Applications (2012)
- [89] Vieira K, Schulter A, Westphall C
Intrusion detection for grid and Cloud computing
IEEE Journal IT Professional, pg no:38– 43, July 2010.
- [90] Danger Theory Based Hybrid Intrusion Detection Systems for Cloud Computing,
International Journal of Computer and Communication Engineering, Vol. 2, No. 6,
November 2013
- [91] Kumar Rajendrana*, B.Muthukumarb, G.Nagarajanc,
Hybrid Intrusion Detection System for Private Cloud: A Systematic Approach
(2015)
- [92] Arshad J, Townend P, XuJ.
An abstract model for integrated intrusion detection and severity analysis for clouds.
International Journal of Cloud Applications and Computing 2011;1(1):1–17.
- [93] Lee, J-H, Park M-W, Eorn J-H, ChungT-M.
Multi-level Intrusion detection system and log management in cloud computing.

13th International conference on advanced communication technology (ICACT);2011, pp.552–5.

[94] Kwon H, Kim, T, Yu, SJ, Kim HK.
Self-similarity based lightweight intrusion detection method for cloud computing.
Proceedings of the third international conference on intelligent information and database systems—Volume Part II; 2011: pp.353–62.

[95] Hemairy MA, Amin S, Trabelsi Z.
Towards more sophisticated ARP Spoofing detection/prevention systems in LAN networks.
International conference on the current trends In information technology (CTIT); 2009: pp.1–6.

[96] Roschke S, Feng C, Meinel C.
An extensible and virtualization compatible IDS management architecture.
5th international conference on information assurance and security, 2; 2009: pp.130–4.

[97] Bakshi A, Yogesh, B.
Securing cloud from DDOS attacks using intrusion detection system in virtual machine.
2nd international conference on communication, software and networks; 2010: pp.260–4.

[98] Mazzariello C, Bifulco R, Canonoco R.
Integrating a network IDS into an open source cloud computing.
6th international conference on information assurance and security (IAS); 2010; pp.265–70

[99] Vrushali D. Mane, S.N. Pawar,
Anomaly based IDS using Backpropagation Neural Network
International Journal of Computer Applications (0975 – 8887)
Volume 136 – No.10, February 2016

[100] Dinesh Singh, Dhiren Patel , Bhavesh Borisaniya , and Chirag Modi,
Collaborative IDS Framework for Cloud
International Journal of Network Security, Vol.18, No.4, PP.699-709, July 2016

[101] Salah Talha Babiker, Ayman Ali Abdalla Ali.
Design a Hybrid Algorithm for Cloud Computing Security.
American Journal of Computer Science and Engineering.
Vol. 2, No. 5, 2015, pp. 38-41.

[102] Gulshan Ansari
Framework for Hybrid Network Intrusion Detection and Prevention System
International Journal of Computer Technology & Applications, Vol 7(4), 502-507
ISSN:2229-6093

[103] A Primer, Cloud Computing
The Internet Protocol Journal, Volume 12, No.4, Cisco System

8 Appendixes

8.1 Cisco Signature table for Cloud DDoS attacks

Cisco Signature table for Cloud DDoS attacks.

Example of how Cisco keeps track of the signatures.

CVE ID	Signature Release	Signature ID	Signature Name	Enabled	Severity	Fidelity*	Notes
NA	S672	1493/0	Distributed Denial of Service on Financial Institutions	Yes	High	90	—
NA	S593	2152/0	ICMP Flood	No	Medium	100	Retired
NA	S572	4002/0	UDP Host Flood	No	Low	75	Retired
NA	S520	4004/0	DNS Flood Attack	No	Medium	85	Retired
NA	S593	6009/0	SYN Flood DoS	No	Medium	85	Retired
NA	S573	6901/0	Net Flood ICMP Reply	No	Informational	100	Retired
NA	S573	6902/0	Net Flood ICMP Request	No	Informational	100	Retired
NA	S573	6903/0	Net Flood ICMP Any	No	Informational	100	Retired
NA	S573	6910/0	Net Flood UDP	No	Informational	100	Retired
NA	S573	6920/0	Net Flood TCP	No	Informational	100	Retired

Figure 41 Cisco Signature Track

(Fidelity is also referred to as Signature Fidelity Rating (SFR) and is the relative measure of the accuracy of the signature (predefined). The value ranges from 0 through 100 and is set by Cisco Systems, Inc)

8.2 Images Used

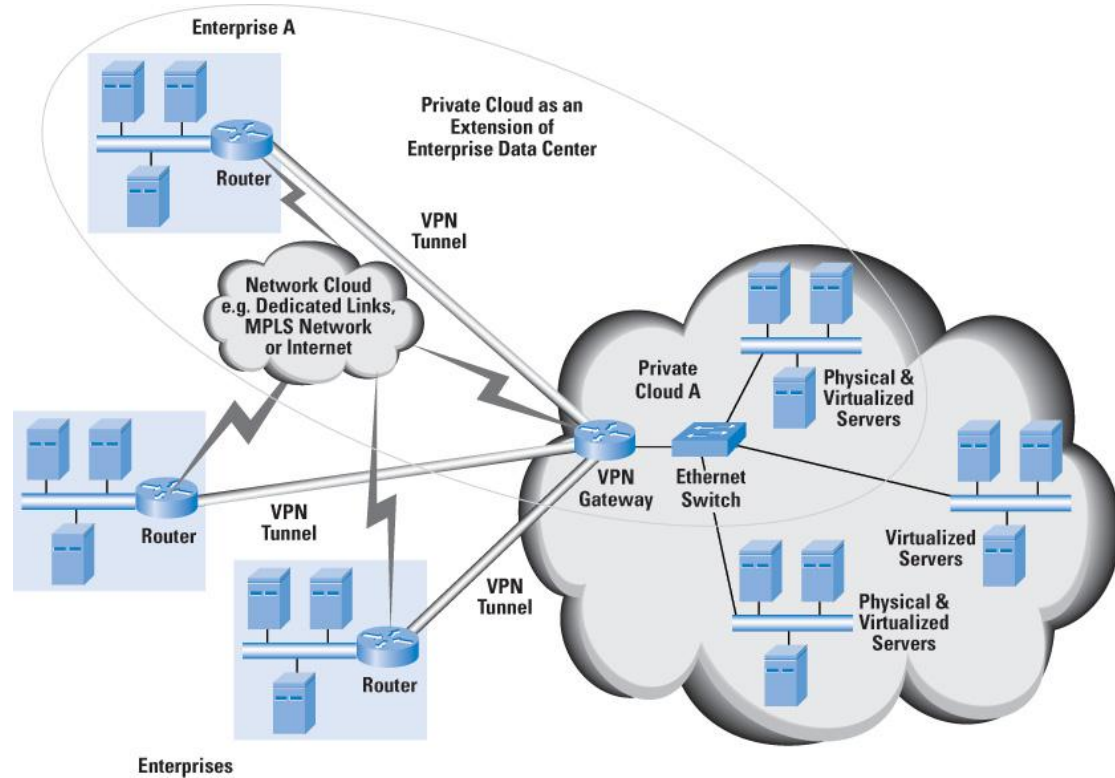


Figure 42Raw Cloud Architecture image

The image above was used as a resource and was modified through the whole dissertation.

It is from “Cloud Computing - A Primer”, from “The Internet Protocol Journal, Volume 12, No.4” (Cisco System) [103]

8.3 Installations / Use of tools selected

8.3.1 Security Onion

The Version used was “securityonion-14.04.5.1”

It is loaded in Oracle’s VM Virtual Box.

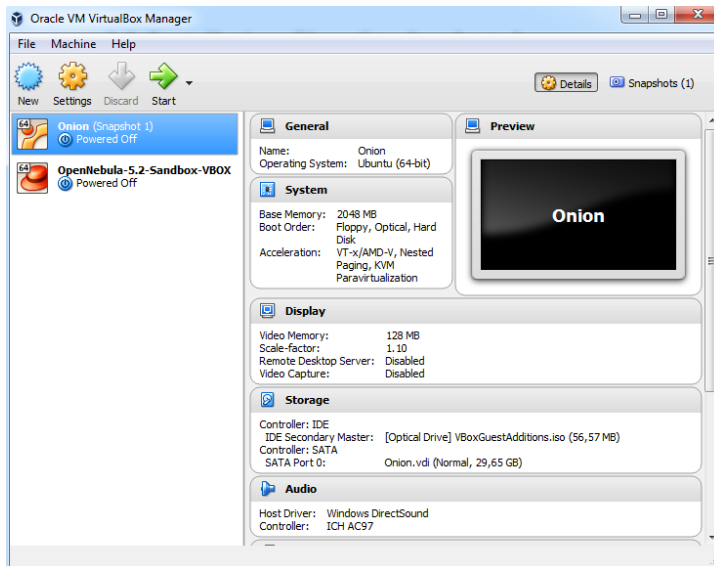


Figure 43 Security Onion in Virtual Box

Inside the interface

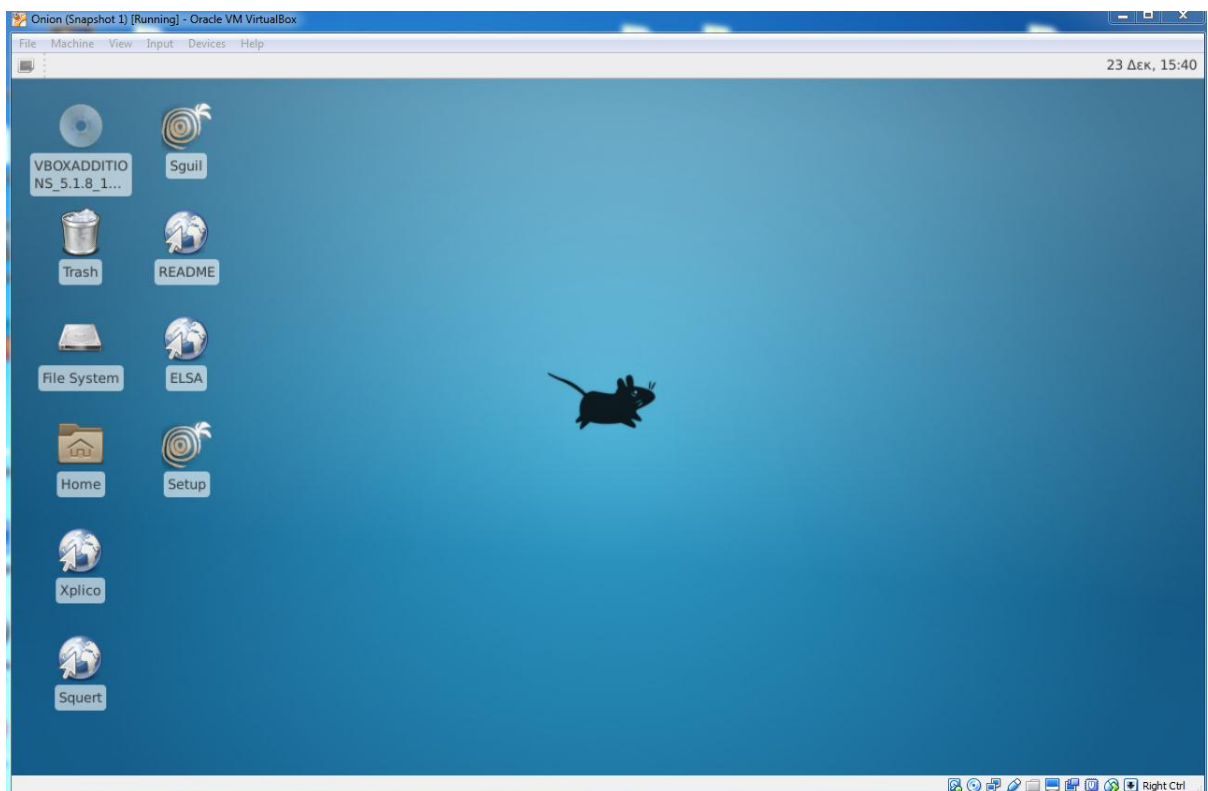


Figure 44 Security Onion Interface

Then we had to setup the system to match our needs. By selecting the “Setup” we get many choices. We will make a run-through.

First, we had to configure the network interface.

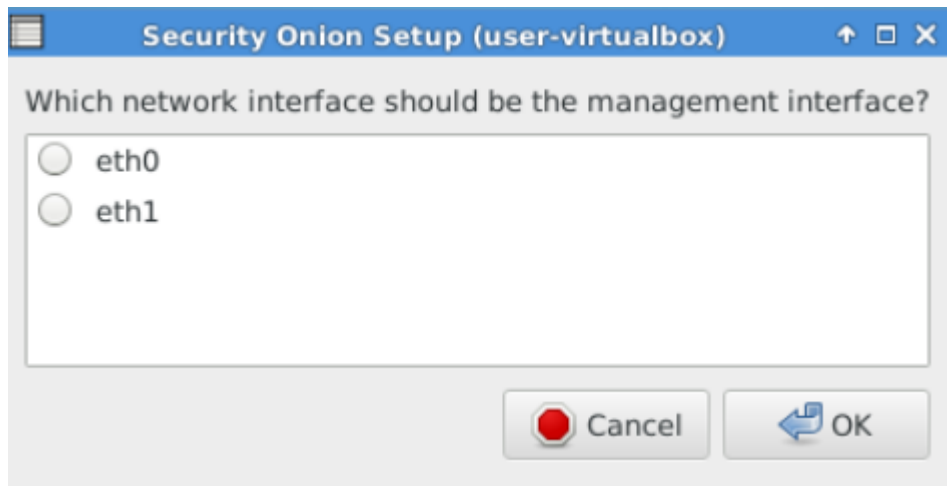


Figure 45 Network interface selection

In the selection, Static or DHCP we go for DHCP, being a virtual box installation, we can't have a static address.



Figure 46 Sensor installation

We are not building a server but a Standalone installation.

And by having the primary network interface as eth0, chose the sniffing to be eth1

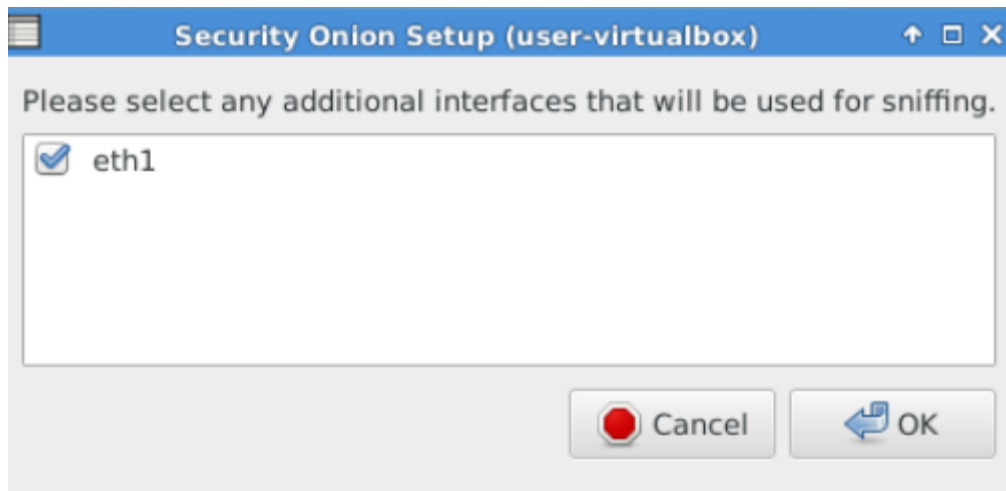


Figure 47 Sniffing interface

Back in the setup again to make our programs choices.

We go with production choice to have total control over the choices made for our system.

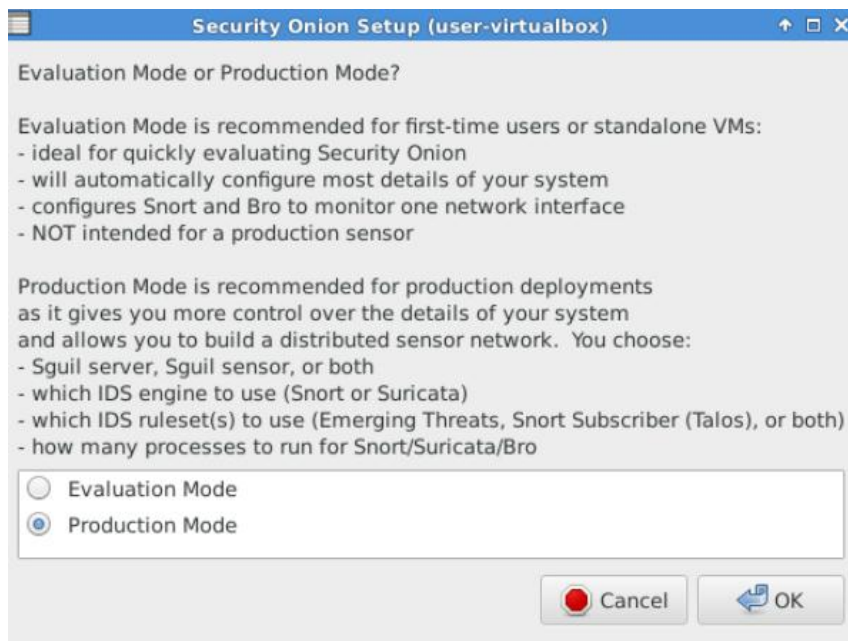


Figure 48 Production Mode

We select Standalone, to configure both server and sensor components.

In future work, when will try to create a DIDS with multiple VM instances, one can have the role of the server.

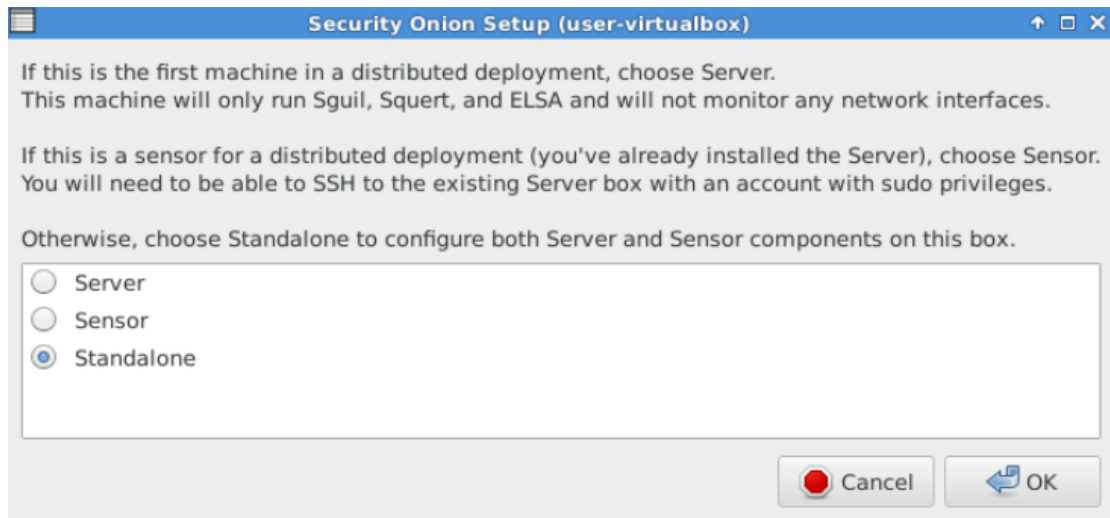


Figure 49 Standalone selection

A pop-up box enquires us, if we want to use the default setting, or continue with custom configuration. We continue with our customization.

We are prompted to select user name and password for Sguil, Squert and Elsa, will go by with "user".

We chose to have the logs stored for 30 days in the database of Sguil and to have a 7-day back-up.

We make the choice to use as NIDS, Snort with the reasoning we explained earlier.

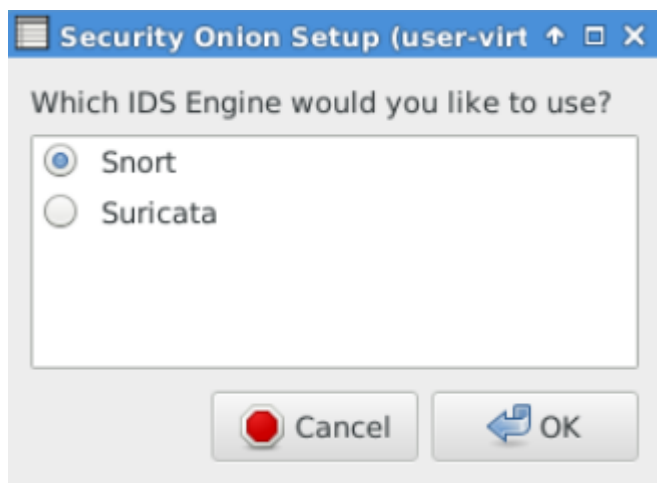


Figure 50 Snort Selection

We know by now, the significance of the ruleset, the signature database for the IDS. Here we have the selection of which of the available databases to use. We select the free one, as the others need codes (which mean subscriptions) to acquire.



Figure 51 Ruleset Selection

More technical questions come next and we go with

- 4096 for PF_RING min_num_slots (this must be higher for busy networks, but will go with the minimum here)
- As we said earlier eth1 will be the monitored network interface

And we enable the IDS



Figure 52 IDS enabled

We leave the default Home_Net configuration:
192.168.0.0/ 16.10.10.0.0/ 8.172.16.0.0/12

And we select to enable Bro. (Bro monitors the network traffic, and helps for analysis-driven network intrusion detection. So, the data collected can be used for behavioral/anomaly detection algorithms)

We select the Bro logs to be stored in Sguil's database.

We won't enable Argus, as Bro does the same job. The same with Prads, as Bro has us covered for its uses also, and we would only end with duplicate reports.

We enable full packet capture, in expense of hard disk space, for the forensic benefits it provides (remembering that one of the OWASP's risks had to do with forensic analysis).



Figure 53 Full packet capture

- We leave the default size pcap file size at 150 MB (pcap stands for "packet capture" and it is a file format for captured network traffic)
- We enable netsniff-ng to use pcap file i/o
- We leave the pcap ring buffer at the default 64 MB
- And again default 90% of disc usage, to begin purging the old logs, as storing space is not infinite
- With the Salt enables we get more help to manage our sensor deployment, with the automatic update of IDS rulesets etc., but we will have it disabled, for it not to interfere with our experiments.
- But will enable ELSA which is also a centralized syslog framework, to check with it our logs. ELSA is even more useful in distributed environments (as our proposal is) as it will collect logs and data from the sensors, and acts in the master server.
- We allocate disk space for ELSA to store logs

And we are done with our choices.



Figure 54 Security Onion Configured

And we are prompted about our first actions, to begin using the system.

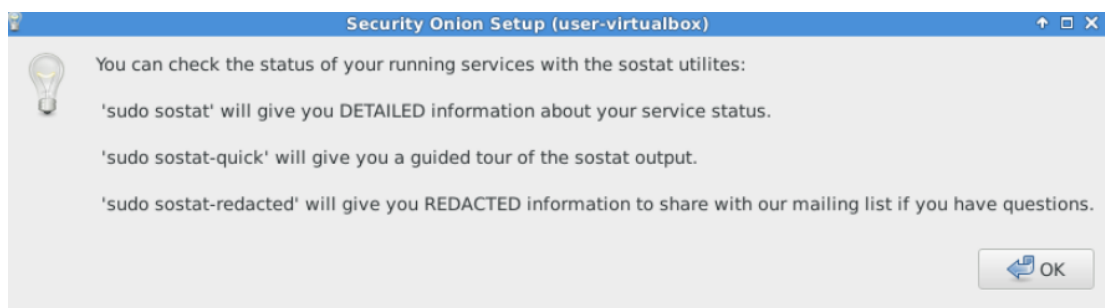


Figure 55 First actions

Local rules are stored in `/etc/nsm/rules/local.rules`

And sensors can be modified in `/etc/nsm/NAME-OF-SENSOR`

8.3.2 Snort Verifications

How the rules are actually stored in the logs

```
File Edit View Text Document Navigation Help
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT CVE-2013-3893 IE Memory Corruption Vulnerability"; flow:establishe
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT CVE-2013-3893 IE Memory Corruption Vulnerability"; flow:establishe
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible Microsoft Internet Explorer Use-After-Free CVE-2013-3897"
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT SUSPICIOUS JS Multiple Debug Math.atan2 calls with CollectGarbage"
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible IE 0day CVE-2013-3918 1"; flow:established,from_server; f
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible IE 0day CVE-2013-3918 2"; flow:established,from_server; f
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible IE 0day CVE-2013-3918 3"; flow:established,from_server; f
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible IE 0day CVE-2013-3918 4"; flow:established,from_server; f
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET WEB_CLIENT BeEF Cookie Outbound"; flow:to_server,established; content:"Cookie
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET WEB_CLIENT Possible BeEF Default SSL Cert"; flow:established,from_server; content:"[0
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible BeEF Module in use"; flow:established,from_server; file_d
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible IE10 Use After Free CVE-2014-0322"; flow:established,to_c
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT EMET Detection Via XMLDOM"; flow:established,from_server; file_dat
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Generic HeapSpray Construct"; flow:established,from_server; file_d
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Possible CVE-2014-1761 HTTP"; flow:from_server,established; file_d
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Microsoft Rich Text File .RTF File download with invalid listoverr
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"ET WEB_CLIENT SUSPICIOUS Possible automated connectivity check (www.google.com)"; flow:es
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"ET WEB_CLIENT SUSPICIOUS Possible automated connectivity check (www.msn.com)"; flow:estab
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"ET WEB_CLIENT SUSPICIOUS Possible automated connectivity check (www.bing.com)"; flow:esta
alert tcp $HOME_NET any -> $EXTERNAL_NET 80 (msg:"ET WEB_CLIENT SUSPICIOUS Possible automated connectivity check (www.yahoo.com)"; flow:est
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET WEB_CLIENT Microsoft Application Crash Report Indicates Potential VGX Memory
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET WEB_CLIENT Microsoft Application Crash Report Indicates Potential VGX Memory
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Base64 Encoded Java Value"; flow:established,to_client; file_data
#alert tcp $EXTERNAL_NET {!21,!22,!23,!2100,!3535} -> $HOME_NET 1024:65535 (msg:"ET WEB_CLIENT Possible GnuTLS Client ServerHello SessionID
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Adobe Flash Player Rosetta Flash compressed CWS"; flow:established
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Adobe Flash Player Rosetta Flash compressed FWS"; flow:established
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Adobe Flash Player Rosetta Flash compressed ZWS"; flow:established
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET WEB_CLIENT DRIVEBY Social Engineering Toolkit JAR Download"; flow:established
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT DRIVEBY Social Engineering Toolkit JAR filename detected"; flow:es
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT DRIVEBY Social Engineering Toolkit Web Clone code detected"; flow:
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Malicious iframe guessing router password 1"; flow:established,fro
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET WEB_CLIENT Malicious iframe guessing router password 2"; flow:established,fro
```

Figure 56 Snort Rules default databases

Here we find tenths of thousands of entries.

We create our own in local ruleset

8.3.3 SGUIL

Sguil is the interface to monitor and evaluate the data gathered by the sensors.

Here we see the OSSEC (our HIDS) and events it collects. OSSEC has reported the Integrity Checksum change. Being an HIDS, checking the checksum is a common practice to inspect the integrity of files.

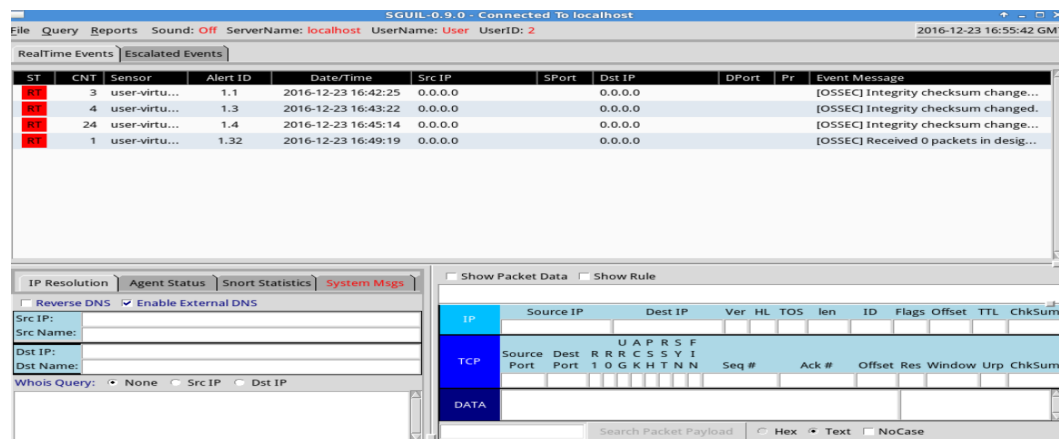


Figure 57 Sguil and OSSEC

Our enabled agents (Snort NIDS, OSSEC HIDS), as Sguil can collect data, from all the sensors, for further checking, evaluation and analysis.

Sid	Net	Hostname	Type	Last
1	user-virtual...	user-virtual...	ossec	2016-12-23 1
2	user-virtual...	user-virtual...	pcap	2016-12-23 1
3	user-virtual...	user-virtual...	snort	N/A
4	user-virtual...	user-virtual...	http	N/A

Update Interval (secs): 175 NOW

Figure 58 Snort and OSSEC in Sguil

8.3.4 Nmap

In order to do a port scan in our machine, we installed the windows version of Nmap

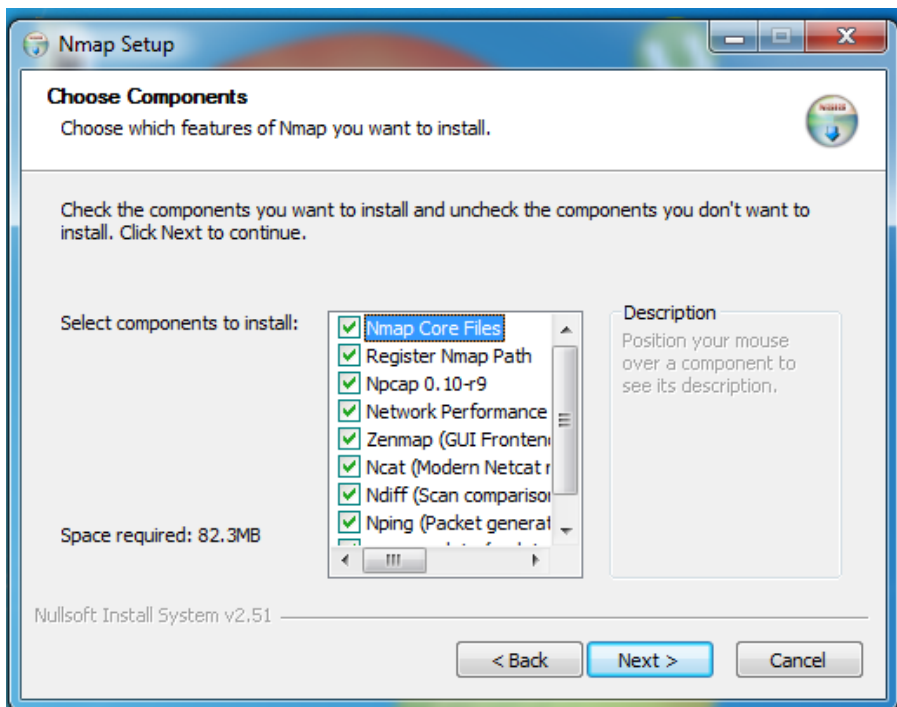


Figure 59 Nmap setup

This came along with various other needed programs also installed within the same package.

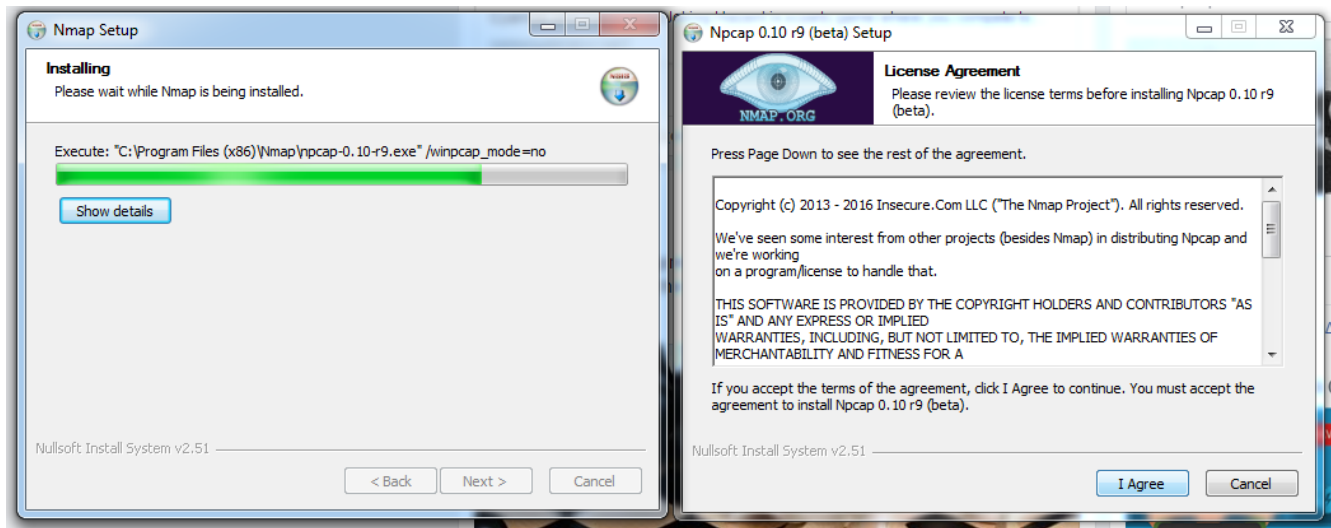


Figure 60 Npcap setup

It provides automatic options for the Scanning such as intense scan, ping scan, udp scan, all tcp ports scan. It need an IP as input, to be the target of the attack.

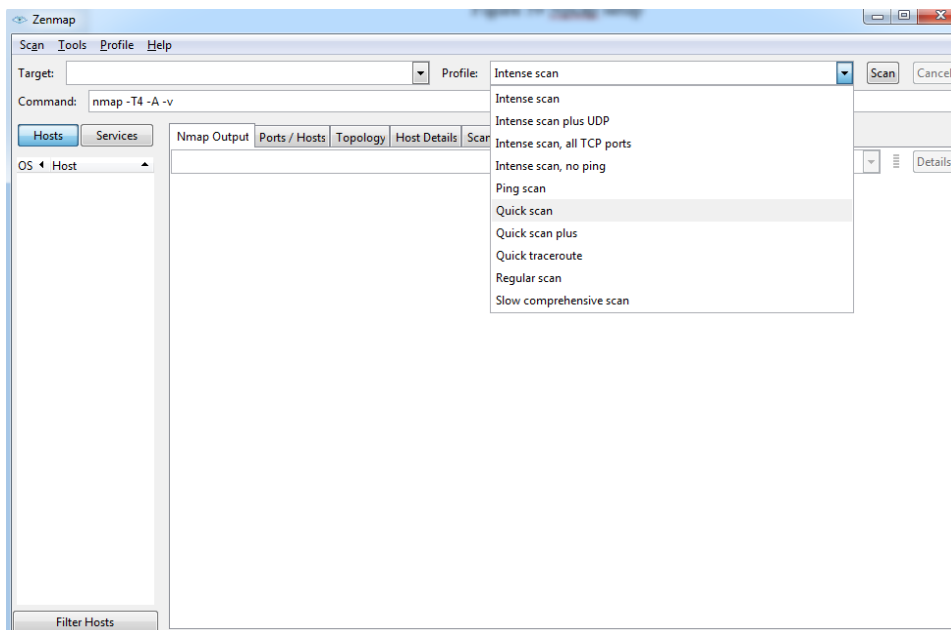


Figure 61 Nmap Interface

While experimenting with the tool, we used an IP outside our local network. The tool also provides a map, with the route of the packets it sends.

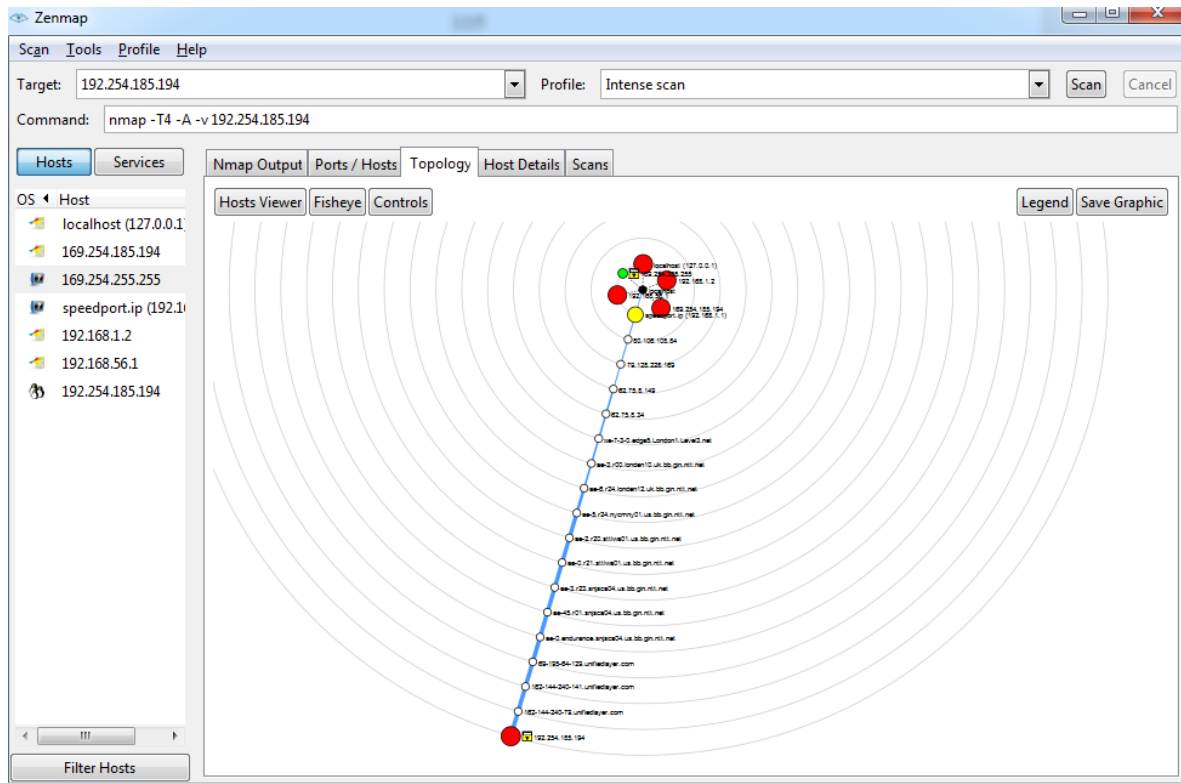


Figure 62 Nmap Map

Here we see the main target of our local network we attacked for testing, plus a remote target, and with all the switches passed through being recorded.

8.3.5 Hyenae

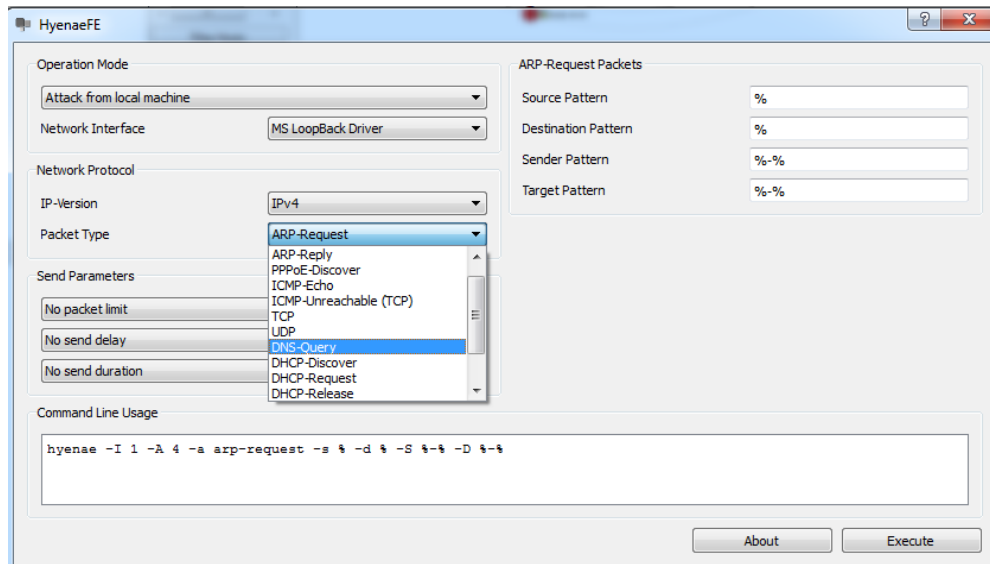


Figure 63 Hyenae

This tool enables us to perform various DDoS attacks at target IPs. It is capable for the SYN flood we tested, ICMP, DNS query, DHCP Discover/Request/Release and many more.