Network Malware Laboratory Based On Honeypots Technologies

Georges Bell Bitjoka, University of Ydel, Cameroon
Antoine Elang, National Advanced School of Posts and Telecommunications, Cameroon

ABSTRACT

According to studies conducted by researchers across the globe, in recent years there has been an increase in organization and company attacks. Some attacks have been detected, but others, however, were able to bypass the security mechanisms, taking advantage of an unknown vulnerability in security systems. In this context, Honeypots systems aim to collect information on the intruder's activities and learn about threats and attackers' behavior. Honeypots systems are not designed to remedy failures or security errors on the network, but are responsible for providing adequate information on potential attackers before compromising real systems. In this paper, a honeypot system was designed to study the techniques used by attackers. We designed and implemented a malware analysis laboratory based on honeypots technology in a controlled environment to analyze various security incidents. The use of honeypots is based on the idea of simulating applications with vulnerabilities and recording all events produced by attackers, so the network administrator can learn about the different types of attacks to protect organizational systems that are being produced. The results have been very important in terms of the number and types of security incidents recorded by the honeypots. Also, an administration interface for controlling and analyzing the gathered information was designed. This system was not only implemented but also tested for several weeks and data was collected from the attacks was analyzed. This led to some interesting statistics and characteristics about attackers and their goals.

Keywords: Malware; Malware Laboratory; Virtual Machine; Honeypot; Honeynet

1. INTRODUCTION

The purpose of this study is to defend the information infrastructure, detect faults in structures, and correct them proactively. In recent years, attacks generated by individuals with malicious purposes, have increased significantly. This, combined with existing vulnerabilities in all types of operating systems and applications, results in organizations becoming a potential victim. While security is closely linked to knowing that there is no absolute certainty, what we're struggling to do is to cushion the impacts and risks by combining different existing tools. Therefore, in this context, it is of paramount importance to consider new strategies and techniques to generate an extra protection layer. This is where honeypots technology, through which attacks and network vulnerabilities are known in detail play a vital role. Honeypots are used to obtain valuable information on threats, which can be observed, analyzed, and monitored to prevent further attacks and to identify the techniques used by the attackers.

2. OBJECTIVE

The main purpose of this paper is to propose the creation of a test bed that will help network administrators analyze malware and monitor their behavior. The requirement for such an instrument resulted from the disclosure, in the greater part of the examination papers that were read, that scientists never said how they set up their test. The main objectives of the paper are:

- Develop a malware laboratory using virtual machines and honeypots to capture and analyze malware. More precisely, the laboratory should give room to the ability to create an infinite number of network environments combining different operating systems and applications that facilitate the study of malware in the wild. To create these isolated environments, we have used virtualization technology. This technology will give way to the creation of isolated environments, flexible and scalable, where we can run as many experiments as we need.

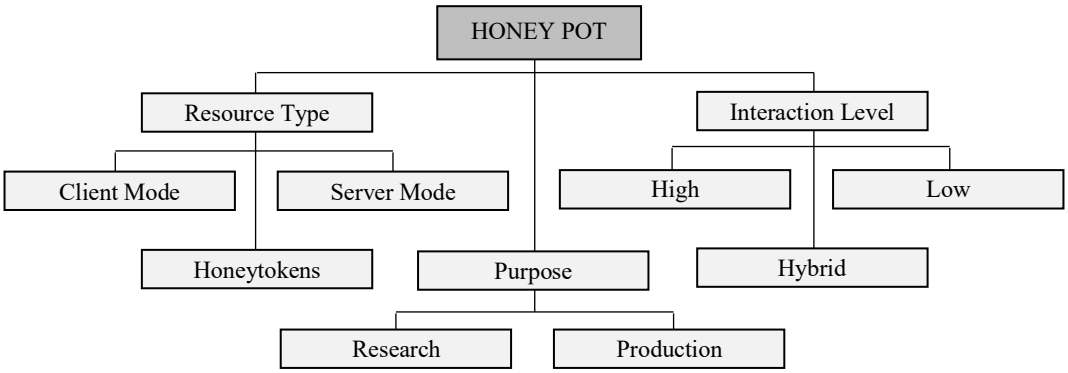
- Design of a platform to manage the infected computers and the analyzed malware itself.
- Development of a main repository to store and analyze malware.

3. LITERATURE REVIEW

3.1 Honeypot

A honeypot is an intentionally exposed computational resource which is aimed at being tested, attacked, compromised, used, or accessed in any unauthorized way. The resource can be a System Service, an Application User or Server, a complete System, or just a piece of information such as records in a database or office documents (ENISA, 2012). In a production environment, any attempt to access or interaction with the honeypot is suspicious activity. All activities between a supposed attacker and the honeypot are monitored and analyzed in order to detect and confirm an unauthorized use. In this way, it is possible to take preventive measures or provide for contingency.

Figure 1. Honeypots classification



3.2 Placement

3.2.1 At the External Network

Placing a honeypot in the public address space of an organization, for example, before a BGP (Border Gateway Protocol) router, gives way to a lot of pieces of information targeted from the Internet. This is the most deployed solution in research environments since it enables a researcher to collect a large amount of malware samples and to detect attacks and zero-day vulnerabilities. Locating the honeypot at this location reduces the risk on the internal network in case it was compromised and used as a jumping machine to access or infect other computers in the network (Joshi & Sardana, 2011)

3.2.2 At the Demilitarized Zone (DMZ)

This architecture is the most difficult to implement because the honeypots are exposed to Internet services and internal networks. Therefore, the security level must be critical. A honeypot in the DMZ can collect information and alert about external attacks to those services allowed by the firewall of the DMZ. The honeypot can detect unauthorized actions from the internal network as well (Joshi & Sardana, 2011).

3.2.3 At the Internal Network

In the network there are PCs and backend servers. Within the internal network exists separate subnetworks, according to purpose, Geographic Location or Ownership. Therefore, a network segment without a previously assigned address can be used to deploy one or more honeypots. This separation facilitates network administration, but also enables honeypots to identify internal attacks because the traffic from other internal networks should not interact with the honeypots. Therefore, such an activity would be considered suspicious (Joshi & Sardana, 2011).

3.3 Honeypots vs. Other Technologies

There are several technologies and tools oriented to malware analysis and intrusion detection (Joshi & Sardana, 2011). The one we choose will depend on our needs.

Sandboxes are security mechanisms that enable files to run in isolated environments and get information from actions they take. This technology is used in malware analysis. Generally, they perform a dynamic and real-time analysis of programs or files executed in a virtual operating system. Sandboxes are used to study malware samples once they have been captured by another tool, for example through honeypots. Some examples of sandboxes are Cuckoo and Anubis (Egele, Scholte, Kirda, &Kruegel,2011; Oktavianto&Muhardianto, 2013).

IDS/IPS are technologies based on the detection and mitigation of network attacks. They inspect network packets for suspicious patterns. These tools do not capture malware but may block some activities. Generally, IDS/IPS are used in conjunction with other security tools (firewalls, honeypots, honeynets, etc.). Some examples are Snort and Suricata (China Appala Naidu, & Avadhani, 2013).

Antivirus is another tool used for analyzing and detecting malware. They are a security measure implemented on local computers. They contanalyze the file system of a computer continuously looking for binaries containing code patterns classified as malignant. Online antivirus engines have become a very useful tool for static analysis of files. As an example, the most popular are VirusTotal, Jotti and Metascan (Ligh, Adair, Hartstein & Richard, 2010).

3.4 Honeynets

A honeynet is a network architecture composed of honeypots, network devices, and security tools. Honeypots in a honeynet are real operating systems, that is, they are high-interaction honeypots. When the systems of a honeynet are attacked, the honeynet logs all information about the activities taking place (Spitzner, 2002). For that purpose, there are a number of common components to every honeynet:

- Router: Routes the traffic to the different devices in the honeynet.
- Firewall: Restricts incoming and outgoing traffic to/from the honeynet.
- IDS/IPS: Detection and prevention system which enables to analyze network packets traffic and content in more detail.
- Server Logs: The information collected by honeypots and the rest of devices are sent to a centralized logs Server.

3.5 Virtual and Physical Honeynets

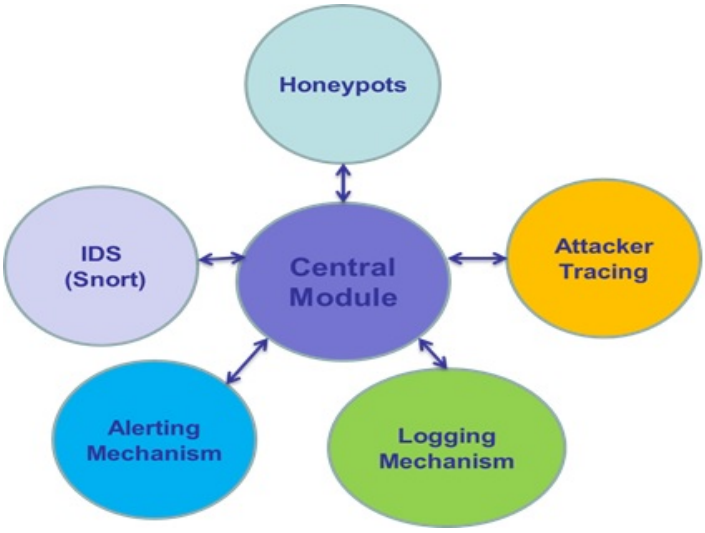
In a physical honeynet, honeypots and other systems are running on separate physical machines. A virtual honeynet deploys honeypots and other systems on virtual machines which run on the same physical machine (*Lu*, *Tavallaee*, *Rammidi* & *Ghorbani*, 2008). There are two types of virtual honeynets:

- Self-Contained Honeynet: All components (honeypots, honeywall, IDS, router, etc.) are implemented on the same physical machine by using virtualization.
- Hybrid Honeynet: Honeypots run on virtual machines within the same physical machine, but the basic devices (honeywall, IDS, router etc.) are deployed on another physical machine. Hybrid honeynets imply a security enhancement which decreases the likelihood of having the full honeynet compromised by an attacker.

4. RESEARCH METHODOLOGY

In this work, we used different tools to build a unique framework, to have a centralized console that will be used by scientists to have access to the entire system. This console has a database which stores information about studied bots, but also virtual machines used to analyze them. This console is used to manage the whole system. The system design can be found in Figure 2.

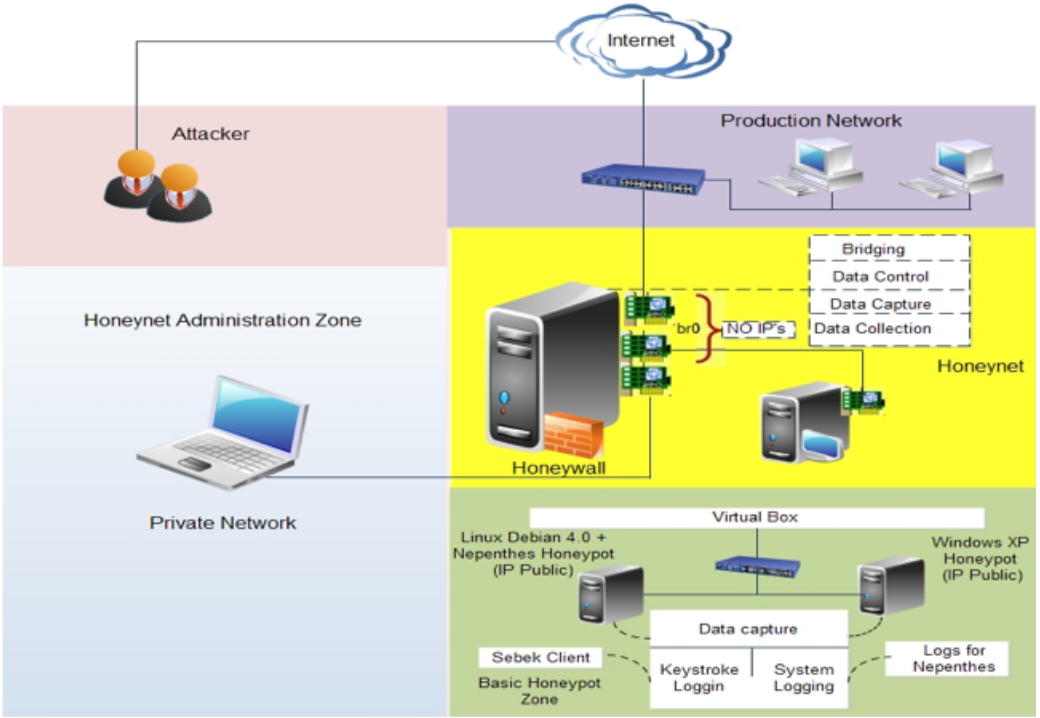
Figure 2. System design



4.1 Architecture

The architecture corresponds to a Virtual Hybrid Honeynet, as shown in Figure 3. For its development we will use two physical machines: (1) the Honeywall and (2) used by virtualization software, raise two virtual machines that correspond to the honeypots. In our experiment, we used the operating systems: Windows XP and the Linux Debian distribution.

Figure 3. Honeynet architecture



4.2 Implementation

4.2.1 Hardware

For the implementation, we will use only one physical machine. We first give the minimum characteristics of the host machine hosting the virtual machines:

- Processor Pentium Dual Core 1.7 GHz
- Memory 1 GB RAM
- Hard Drive 300 GB
- Network Adapter 10/100/1000 Mbps

The implemented virtual machines form a virtual network within the host machine, as shown in Figure 4, and the minimum hardware requirements are listed in the Table 1:

Figure 4. Virtual Honeynet Architecture

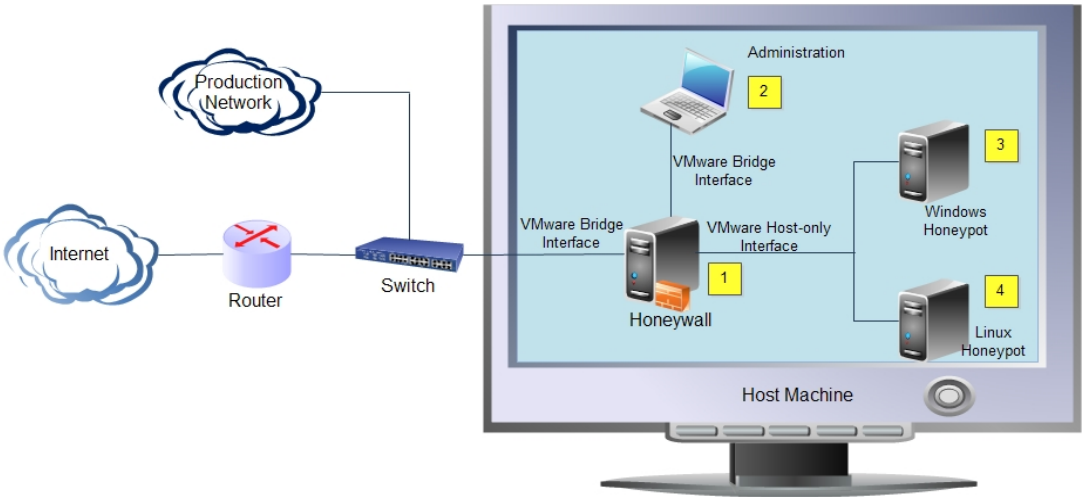


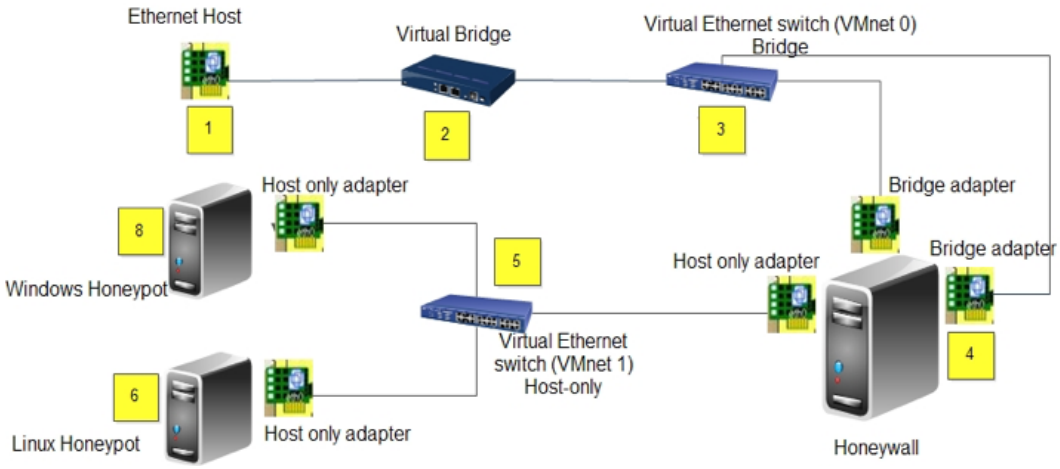
Table 1. Operating Systems

OS	Hard Drive	Memory (RAM)
Linux Debian	30 GB	512 MB
Windows XP	30 GB	512 MB
Honeywall	100 GB	256 MB

4.2.2 Network Setting

The network diagram in Figure 5 shows a view of the Honeynet with physical and virtual components. A single physical machine is connected directly to the switch next to the production network. The host is a Linux Fedora distribution using VMware virtualization software to host the three virtual machines in the Honeywall. The Honeywall [1] uses three virtual network interfaces (one in bridge mode and two in host only mode). Honeypots [3 and 4] use the host mode network interface only. Only in Host mode virtual machine can connect to the hosting machine and other virtual ones with similar configurations, creating an internal private network isolated from the rest of the external network. In the Bridged mode, virtual machines use the same host's network connection, but are connected are different terminals.

Figure 5. Virtual Honeynet Architecture



4.2.3 Configuration of the Honeynet Connectivity

The external and internal networks of honeynet are placed in segments with different IP addressing. This configuration is not ideal since the honeywall should act as a bridge between two segments with the same address. The decision to implement networks with different address for internal and external segments of the honeynet is due to hardware limitations and VMware's characteristics. Given that the entire honeynet and machine attacker is implemented on the same physical computer, connectivity through the VMware virtual Switch does not permit to configure the desired routing. One possible solution to the problem would be to include a physical network device between the honeywall and the attacker machine.

- External honeynet interface:
  - IP: 192.168.30.0/24
  - Interface: eth0.
  - VMware switch: Host-Only (vmnet 5)
- Internal honeynet interface:
  - IP: No IP
  - Interface: eth1.
  - VMware switch: Host-Only (vmnet3)
- Management honeywall interface:
  - IP: 192.168.50.10/24
  - Interface: eth2.
  - VMware switch: Host-Only
  - (vmnet4)
- Bridge interface:
  - HoneywallRoo creates a bridge interface br0 automatically. It detects and associates interfaces eth0 and eth1 in order to form a transparent bridge.

To control outside connections generated by the attacker, the number of connections allowed are limited. This will limit the effectiveness of an attack from the honeypot to third systems. The number of connections has to be adjusted depending on the environment and purpose of the honeynet. Tables 2, 3, and 4 indicate the maximum number of connections allowed.

Table 2. INPUT chain rules

Action	Protocol	Interface In	Source	Destination	Ports
Accept	Any	Loopback	Any	Any	Any
Accept	TCP	Eth2	192.168.50.0/24	Any	24, 443
Drop	Any	Any	Any	Any	Any

Table 3. OUTPUT chain rules

Action	Protocol	Interface In	Source	Destination	Ports
Accept	Any	Loopback	Any	Any	Any
Accept	TCP	eth2	any	Any	20, 21, 22, 25, 80, 443
Accept	UDP	eth2	Any	Any	53, 123, 69
Drop	Any	Any	Any	Any	Any

Table 4. FORWARD chain rules

Action	Protocol	Interface In	Interface Out	Source	Destination	Ports
Accept	Any	Any	Any	Any	192.168.30.255	Any
Accept	Any	Any	Any	Any	255.255.255.255	Any
Accept	Any	eth0	Any	Any	Any	Any
Accept	UDP	eth1	Any	Any	192.168.50.10	65000
Accept	UDP	eth1	Any	Any	255.255.255.255	Src. 68 Dest. 67
Accept	TCP/UDP	eth1	Any	192.168.30.100	Any	53
Accept	Any	eth1	eth1	Any	Any	Any
Accept	TCP	eth1	Any	192.168.30.100	Any	Any-200 connections/hour
Accept	Any	eth1	Any	192.168.300.100	Any	Any- 100 connections/hour

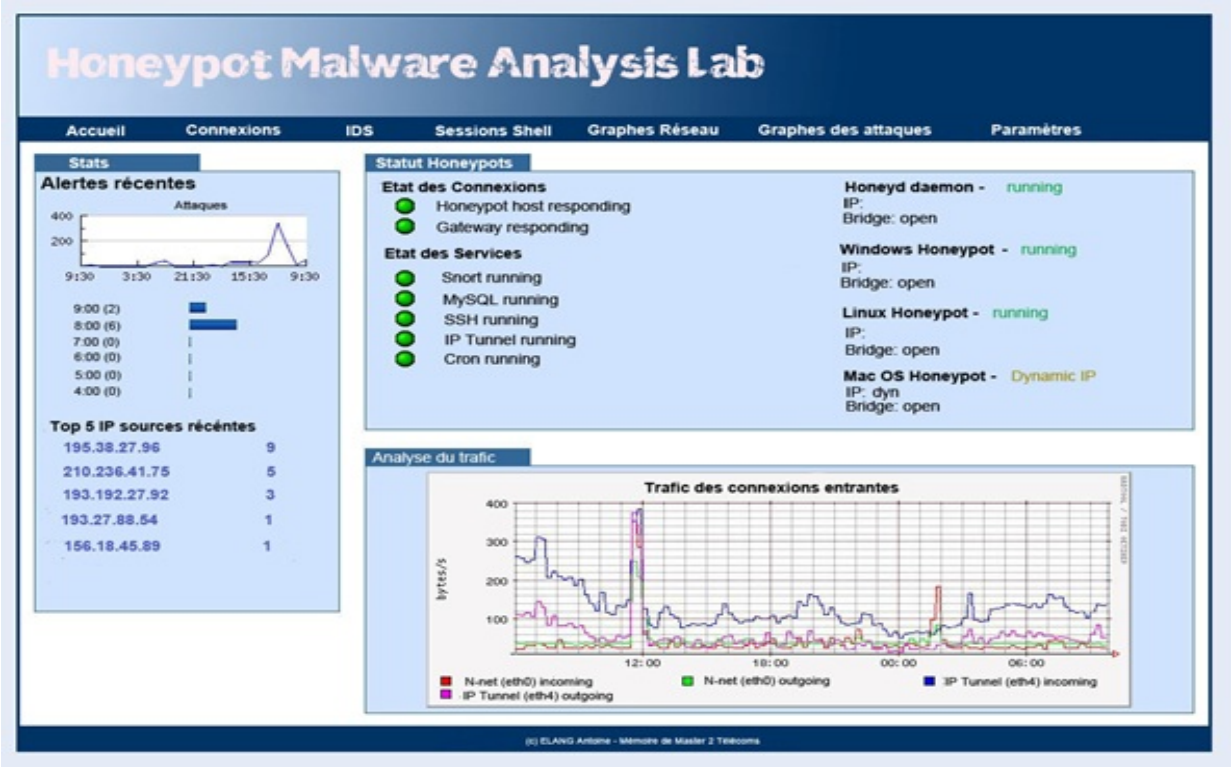
5. RESULTS AND ANALYSIS

5.1 Central module

It is used to manage and control the whole system, by connecting different parts of the system, using interfaces to view and set functions as:

- View the honeypot logs.
- View *Snort* Alerts
- View Complete Packet Logs
- View honeypots connections
- Change settings

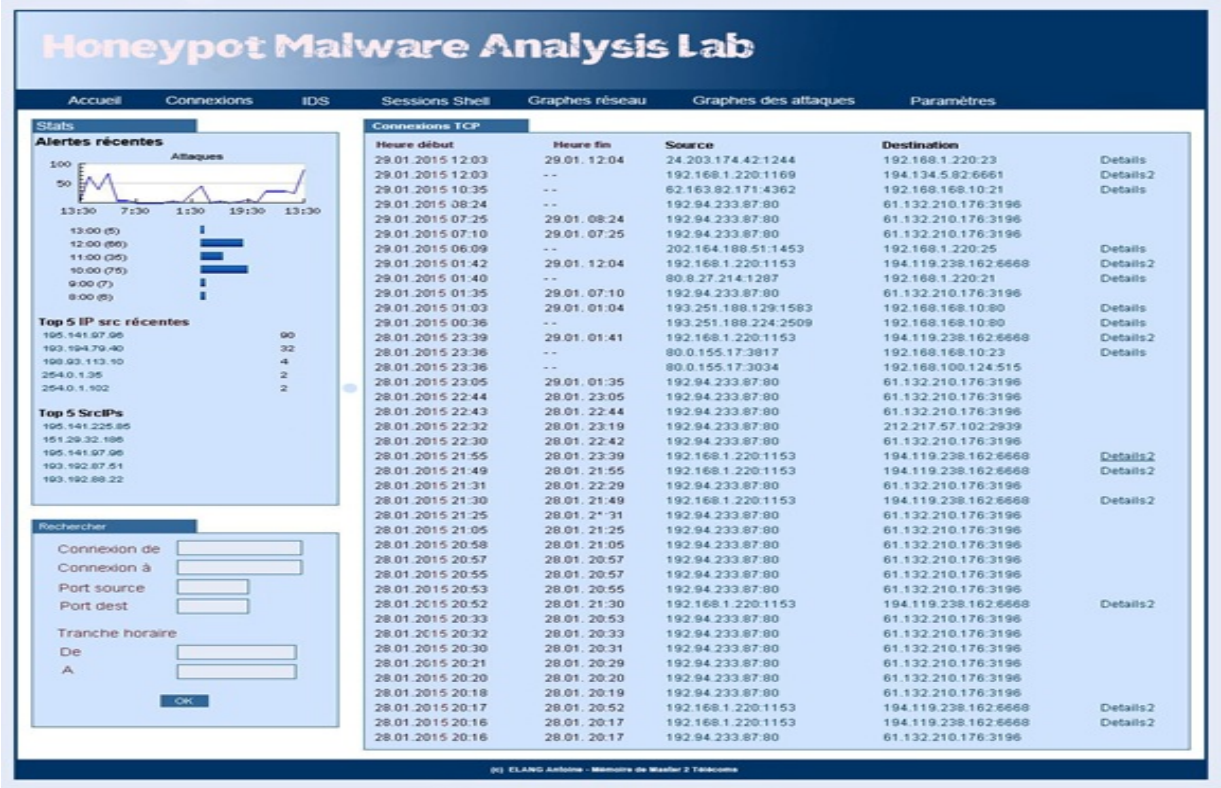
Figure 6. Honeypot Malware Analysis LaboratoryInterface



5.2 Connections

As the system is running, we can list all incoming and outgoing connections of the honeynet in general and of each particular honeypot. We use SNORT to analyze alerts generated and create rules based on the input information to test the entire system.

Figure 7. Honeypot Malware Analysis Laboratory connections



5.3 Statistics

Information about type and quantity of data is important to analyze malware in network traffic. We use NTOP to that purpose.

Figure 8. Attacks on the honeynet over the entire time interval

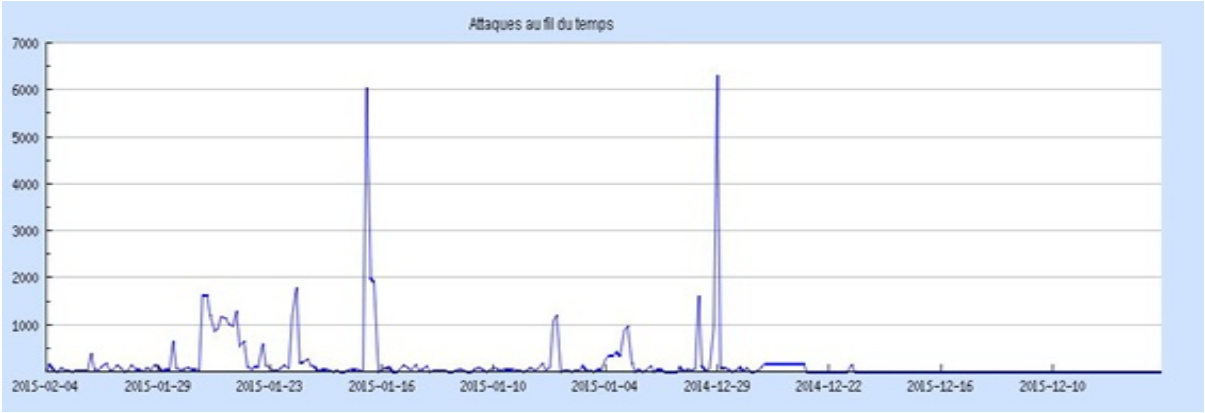


Figure 9 below shows overall honeypot attack activity over the weekdays. Figure 10 shows attacks statistics by level domain.

Figure 9. Weekdays attacks statistics

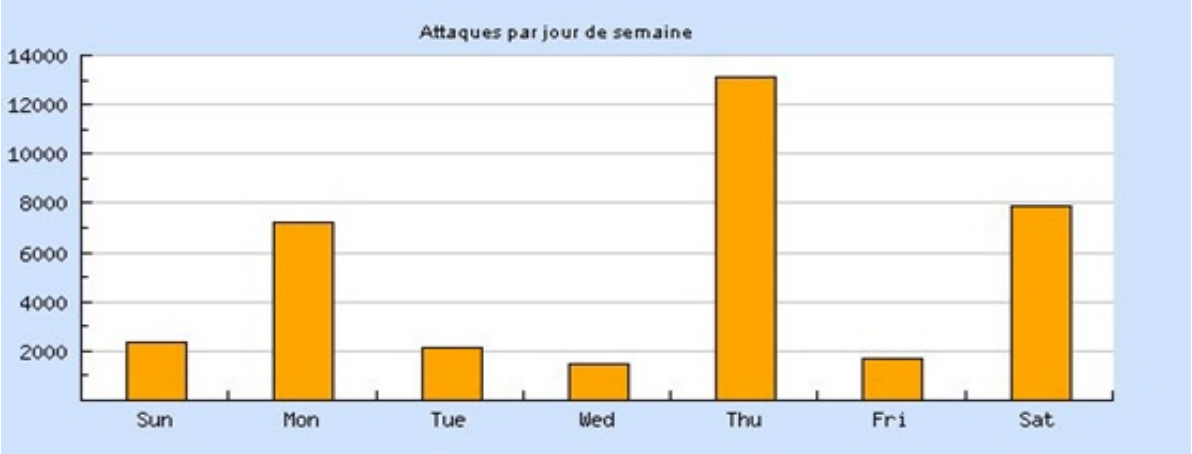
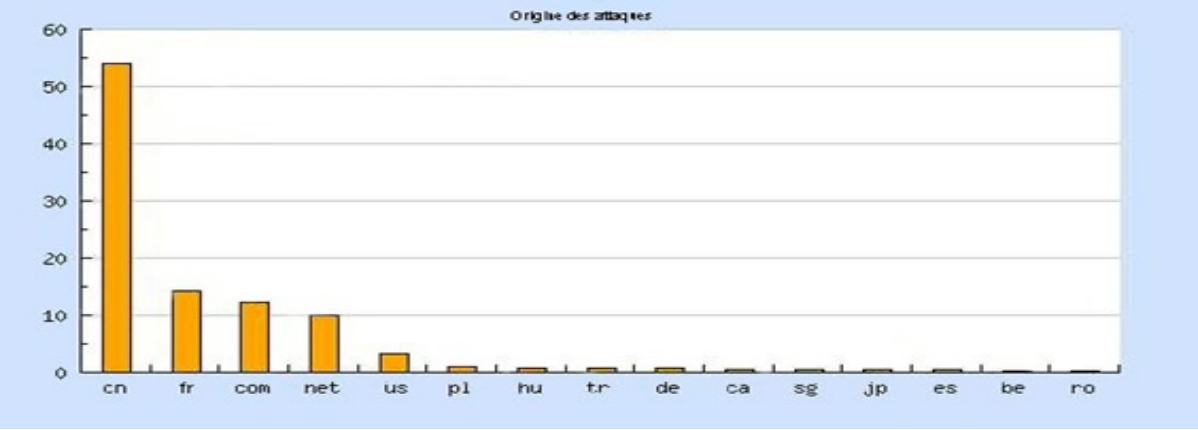
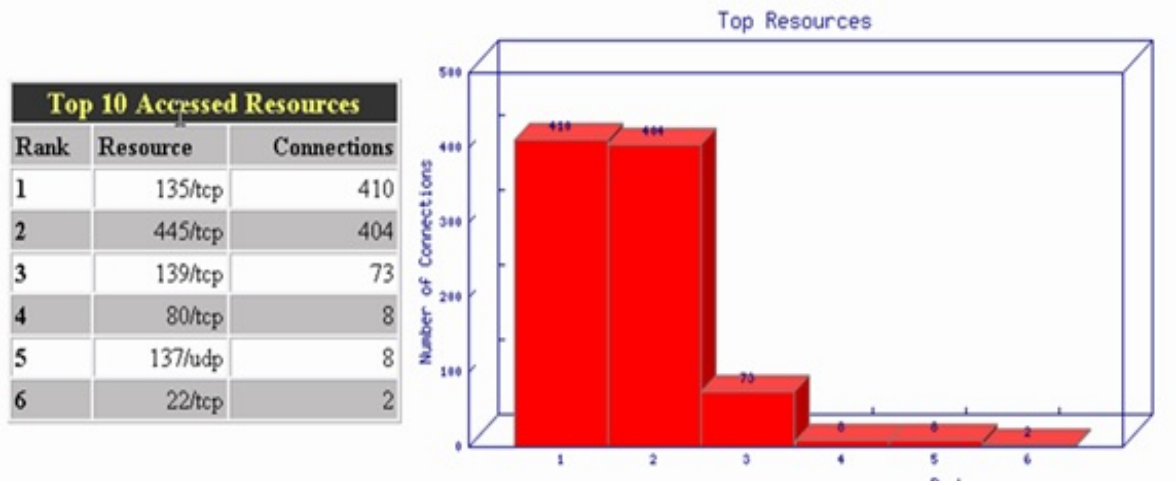


Figure 10. Level domains attacks statistics



We finally provide some other interesting statistics in Figure 11.

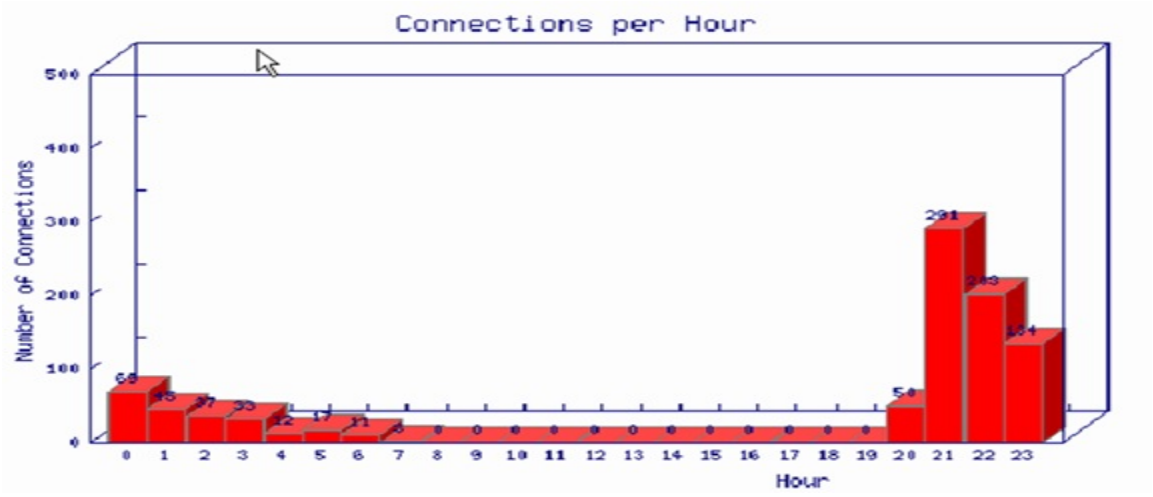
Figure 11. Accessed resources statistics



Top 10 Accessed Resources

Rank	Resource	Connections
1	135/tcp	410
2	445/tcp	404
3	139/tcp	73
4	80/tcp	8
5	137/udp	8
6	22/tcp	2

Figure 12. Hourly attacks statistics



6. CONCLUSION

The logical evolution of systems is not attack-free against malicious software. On this score, their study, analysis and early detection are the most important elements to prevent them. Honeynets enable us to collect malware samples and attack vectors for a later study with the ultimate goal for developing protection techniques. From our experiment and results gathered, we have shown that honeypots are powerful tools that enable us to study and analyze the type of malware and attacks and also serve as early warning systems against security incidents and first line of defense against attacks. We have implemented a self-contained virtual honeynet along with a high interaction honeypot. The data collected by our honeynet enables us to obtain information that would not have been possible otherwise. Systems can be used by any scientist working on malware by adapting tools to his one specific question of research.

AUTHOR AUTOBIOGRAPHY

Bell Bitjoka Georges Network Intrusion Detection System and National Advanced School of Engineering of the University of Yde1, Cameroon; Email: georges@bellbitjoka.com; Tel: (+237) 694031476

Elang Antoine Network Intrusion Detection System and National Advanced School of Posts and Telecommunications; Yaoundé – Cameroon; Email: antoine.elang@gmail.com; Tél: (+237) 677707449

REFERENCES

Egele, M., Scholte, T., Kirda, E., &Kruegel, C.(2011). A survey on automated dynamic malware-analysis techniques and tools.*ACM Computing Surveys*.

European Network and Information Security Agency "*ENISA*" (*2012)*. report.

Joshi, R., & Sardana, A.(2011). *Honeypots: A new paradigm to information security.*Nishant Doshi MEFGI, Gauridad Campus, India.

Ligh, M., Adair, S., Hartstein, B., Richard, M. (2010). Malware analyst's cookbook and DVD.

Oktavianto, D., & Muhardianto, I. (2013). *Analyze malware using Cuckoo Sandbox.*

Lu, W., Tavallaee, M., Rammidi, G., Ghorbani, A. (2008). BotCop: An online botnets traffic classifier. GLOBECOM 2008.

Spitzner, L. (2002). Honeypots tracking hackers. IEEE security and privacy.

China Appala Naidu, M., Avadhani, P. (2013). A comparison of two intrusion detection systems. *International Journal of Computer Science and Technology, 4*(1), 316-319.