Dionaea Honeypot Implementation and Malware Analysis in Cloud Environment

Shahrooz Shahrivartehrani
BET (HONS) in Networking Systems, MIIT UniKL City Campus, 1016 Jln Sultan Ismail, 50520 Kuala Lumpur, Malaysia
shahroozdionaea@yahoo.com

Shadil Akimi Bin Zainal Abidin
BET (HONS) in Networking Systems, MIIT UniKL City Campus, 1016 Jln Sultan Ismail, 50520 Kuala Lumpur, Malaysia
shadil@miit.unikl.edu.my

***Abstract* — The purpose of this paper is analyzing malicious attacks in computer networks and the Internet. Dionaea honeypot is deployed on cloud environment to detect and collect malware attacks by reporting the malwares to external analysis service providers in order to understand the behaviour of malicious attacks, so malwares can be detected before becoming a serious threat in the information technology world.**

Keywords— Dionaea, Honeypot, analyzing malware attack,cloud environment, external analysis service provider

I. INTRODUCTION

In today business oriented and heterogeneous networks protecting the security of applications and network resources are very important. Network vulnerabilities are present in each network, on one hand malicious software known as malwares are increasing rapidly, on the other hand human intervention is too slow; therefore eliminating the vulnerabilities is almost impossible in the computer world. The number and the diversity of attacks increase steadily [1]. Malwares such as computer viruses, worms, spyware, rootkits, trojans, adwares, and botnets are considered as significant menace for computer networks, so collecting malware information is so important for two reasons: (1) Firstly, studying about individual pieces of malware permits better defense system against these and similar artifacts. For instance: Antivirus systems and Intrusion detection systems (IDS) are able to upgrade their signature lists. (2) Secondly, collecting statistical information about attack patterns, attack trends, and attack rates is too hard. Practically, much malware is collected and analyzed by detailed forensic examinations of victim systems. The actual malware should be examined by hand, but with huge improving Intrusion Detection Systems (IDS) such as SurfIDS or Nebula (An Intrusion Signature Generator) [4][5]. Honeypots are able to collect these information also: (1) Signature of Bots for content-based detection (2) Information about C&C servers (3) Unknown security holes that allows the Bots to exploit the network and systems (4) Methods and tools that are used by attackers (5) the motivation of attackers [3]. There are three types of honeypots: (1) High-interaction Honeypots, (2) Low-interaction honeypots, and (3) Mediuminteraction or Hybrid Honeypots [6]. Some examples of Lowinteraction honeypots are Honeyd, Nepenthes, Mwcollect, Dionaea, Amun, and Glastopf [4], and two of these lowinteraction honeypots are famous: Nepenthes and Dionaea that have been successful in information collection about malwares and attacks.

increasing rate of malicious attacks and their complexities this cannot be done for a huge proportion of system compromises by human intervention, so a decoy system is required to tempt the attention of attackers to attack to this computer system in order to protect the critical resources in the network [2] [3]. This decoy system is called as Honeypot. Honeypot has no production value, anything going to or from a Honeypot is likely a probe, attack or compromise [4]. Honeypot is used in order to study of tools and techniques that used by attackers, catching and analyzing malwares, Botnets, and 0-day attacks, slowing down attackers and following incoming attackers improving Intrusion Detection Systems (IDS) such as SurfIDS or Nebula (An Intrusion Signature Generator) [4][5]. Honeypots are able to collect these information also: (1) Signature of Bots for content-based detection (2) Information about C&C servers (3) Unknown security holes that allows the Bots to exploit the network and systems (4) Methods and tools that are used by attackers (5) the motivation of attackers [3]. There are three types of honeypots: (1) High-interaction Honeypots, (2) Low-interaction honeypots, and (3) Mediuminteraction or Hybrid Honeypots [6]. Some examples of Lowinteraction honeypots are Honeyd, Nepenthes, Mwcollect,

Dionaea, Amun, and Glastopf [4], and two of theslowinteraction honeypots are famous: Nepenthes and Dionaea that have been successful in information collection about malwares and attacks.

II. DIONAEA

For this project Dionaea is selected as the honeypot because of its installation simplicity and wide capabilities. Dionaea is able to trap malware exploiting vulnerabilities exposed by services offered to a network. The main goal of Dionaea honeypot deployment is gaining a copy of malware which can be a known or an unknown malware attack. Dionaea honeypot emulates some services to attract the attackers to exploit these services. When attack occurs it will interpret the payload and classify it as: (shell or connectback, Url file Download, Command Exec or a Multi-stage payload), then the binary malwares will be stored in a SQLite database and these attacks will be reported to external analysis service providers which are known as sandbox tools.

Dionaea honeypot has several features: (1) Modular architecture is implemented in Dionaea by embedding Python as scripting language to emulate protocols. (2) Most popular protocols in Dionaea are implemented as modules; Protocols which are emulated by Dionaea are SMB, HTTP, FTP, and TFTP. (3) Several modules are available for use over this tool such as MSSQL, MYSQL, and SIP which provides VOIP. (4) Dionaea runs in a restricted environment without administrative privileges. (5) Dionea supports IPV4 and IPV6.(6) [7][8]. (6) Dionea has less services, better logging, and better emulation in comparison with Nepenthes honeypot.

III. IMPLEMENTATION

Dionaea honeypot offers a very flexible design that allows a wide array of possible setups. The most efficient setup is Dionaea sensor setup in cloud environment by virtual private server deployment. It collects information about malicious attacks and stores the information on the local hard disk, and Dionaea sensor reports the malicious traffic to sandbox tools such as CWsandbox, Anubis, Virus Total, and Norman. Figure 3.1 illustrates a possible setup of a Dionaea platform in the cloud environment: Dionaea sensor in cloud environment collects information about suspicious traffic there. This sensor stores the collected information in a local database as SQLite file and other log files, and also forwards the malicious attacks information to sandbox tools.

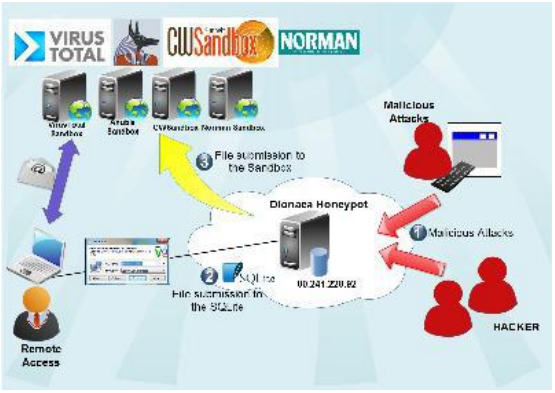

Figure 3.1 Overview of divide and conquer approach

IV.ATTACKED SERVICES

Dionaea sensor started working for 60 days, and the information about the attacked ports are recorded in "dionaea.log", logsql.sqlite file, and "Bistreams" directory. The following monitoring results obtained by observing 'dionaea.log' file and 'Bistreams' directory in Dionaea. Emulated services of Dionaea are attacked 32101 times during 60 days of Dionaea sensor operation. Figure 5.5 shows the number of attacked ports during 60 days Dionaea monitoring. The service with the largest number of attacks is SIP service on port 5060 whereas HTTPS, MYSQL, and SIPTLS services have no attacks. Apart from port 5060, the largest attacked ports are HTTP on port 80, MSSQL on port 1433, SMB on port 445 with number of attacks ranging from about 8599 to 6741 attacks.

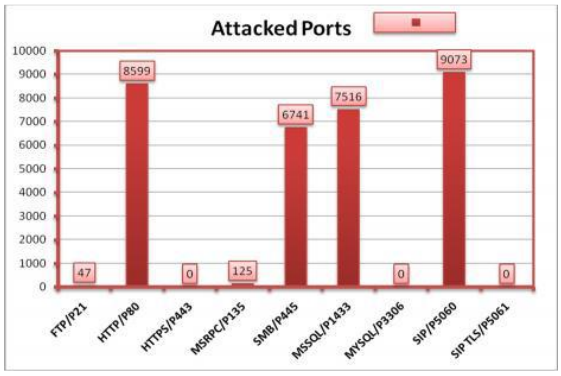

Figure 4.1 Attacked Services

V. DYNAMIC ANALYSIS

After obtaining malicious attacks, sample binary payloads targeted against Dionaea sensor are available; therefore, these binary payloads will be sent to sandbox tools by Dionaea platform for automatic dynamic analysis. Anubis, Norman, and Virus Total are used primarily as sandbox tools. These sandboxes execute and record the behaviour of malicious binaries, and the complete report about the binaries will be generated. The Anubis sample HTML report output can be seen in Figure 5.1.

These Sandbox tools keeps a database of the malware hashes, so it is possible to understand whether this is an old sample, or a possibly new piece of malware or malware variant. Using multiple sandbox tools illustrates different kinds of malware behaviours in order to know the malicious attacks in details.

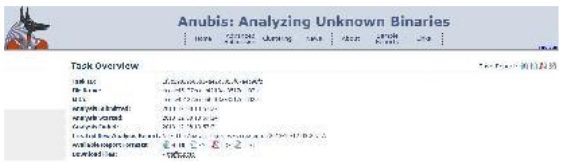

Figure 5.1 Anubis Sample Report

VI. RESULTS

According to the reports from the automated analysis tools, categorizing malware samples is possible, and the approach in order to group the malicious binaries based on vendor detection obtained by Norman and Virus Total sandbox tools. These sandboxes are used for categorizing because threat naming across vendors is often widely varied; therefore, grouping the samples are done manually,

and each of threat group was classified based on their occurrence.

The total number of all captured binary malwares during 60 days is 104 samples including 59 known, 41 unknown, and 4 new binary malwares.

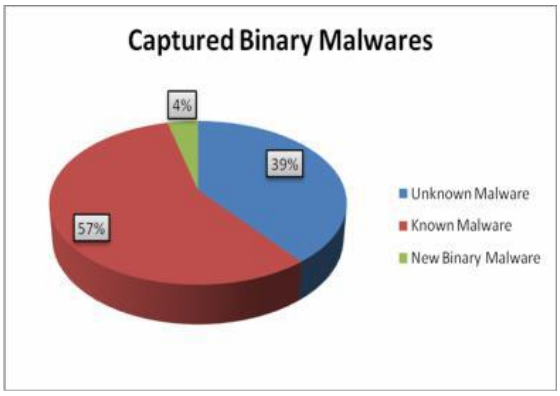

Figure 6.1 Captured Binary Malware

Figure 6.1 shows Known binary malwares are the most frequent malwares in comparison with other malwares. A total of 57 percent of known malwares came in first,unknown malwares came in second with 39 percent, and 4 percent of the captured malwares are recognized as new binary malwares. Moreover,Anubis and Norman sandboxes are used to categorize the known malware binaries into common families of malware, such as Allaple.gen, Downloader, Troj_generic, EmailWorm, Suspecious_Gen2, and Sality. Table 6.1 illustrates the malware families which are categorized by Anubis and Norman sandbox tools, and 6 malware families are selected from 59 samples of known malware binaries. There are 39 samples of the Allaple worm family, 6 samples of Downloader, 6 samples of Troj_Generic, 3 samples of EmailWorm, 3 samples of Sality, and 2 samples of Suspecious_Gen2.

Table 6.1 illustrates the malware families which are categorized by Anubis and Norman sandbox tools, and 6 malware families are selected from 59 samples of known malware binaries. There are 39 samples of the Allaple worm family, 6 samples of Downloader, 6 samples of Troj_Generic, 3 samples of EmailWorm, 3 samples of Sality, and 2 samples of Suspecious_Gen2.

Table 6.1 Malware Families

no	Malware family	Occurrence
1	Allaple.gen	39
2	Downloade	6
3	Troj_Generic	6
4	EMailWorm	3
5	Sality	3
6	Suspecious_Gen2	2

Figure 6.2 shows Allaple worm is the most frequent attacker malware to Dionaea platform in comparison with other malwares. A total of 66 percent of Allaple worm family consists of 6 types of Allaple worms. Troj_generic and Downloader malwares came in second with 10 percent, EmailWorm and Sality malwares came in third with 5 percent each and 4 percent of malwares stated that Dionaea sensor is attacked by Suspecious_Gen2 malwares.

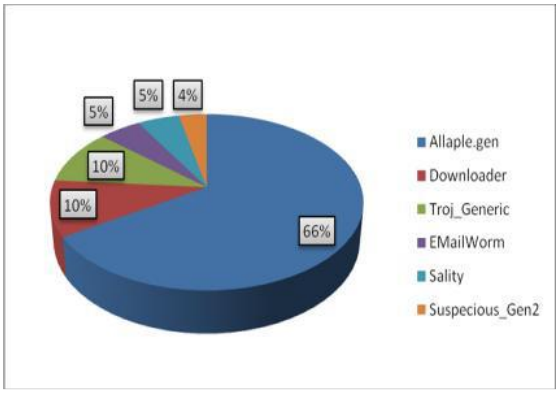

Figure 6.2 Malware Family Frequencies

During 60 days of Dionaea running, 4 new binary malwares are discovered, and submitted by this honeypot platform, and the malware behaviour is categorized as Allaple.gen7 worm which is in Allaple.gen worm category based on Norman and Virus Total sandbox analysis.

Dionaea sensor was able to capture system vulnerability exploits which are deployed by malwares or hackers. Table 6.2 illustrates system exploits which are captured and monitored by Dionaea.

Table 6.2 Exploited Vulnerabilities

no	Exploited System Vulnerability
1	MS03-26
2	MS03-39
3	MS04-11
4	MS04-12
5	MS04-031
6	MS05-17
7	MS05-39
8	MS06-66
9	MS07-65
10	MS08-67

The characteristics that each class of malware can reveal are frequently alike. For instance, a Trojan and a worm may have similar network characteristics. The subsequent part explains the characteristics of malware.

A.Network Activities

Some malwares such as Allaple worm scans and connects to other subnets through port 445 and port 139, and in most scans these malwares identify more potential vulnerable targets. EmailWorm and Sality worms connect to SMTP server on port 25, and the worms spread via email, and scan other hosts. Downloader and Suspicious_Gen2 Trojans connect to a web server via port 80 and download a GIF file as an executable file and open that file to execute it.

B. File Activities

Allaple worm modifies and destructs files which are not temporary. It can keep copies of itself in the Windows directory to stay undetected by users, and this malware is able to change security settings of Internet Explorer, and it can seriously affect safety surfing the World Wide Web due to changing of security settings of Internet Explorer, but Emailworm malware just modifies and destructs files which are not temporary.

Downloader Trojan Creates file C:\WINDOWS\system32\ sysfault.exe which contains malicious codes and downloaded from the website via port 80. Sality worm creates 2 DLL filesin WINSYS folder which are wmimgr32.dl_ and wmimgr32.dll; furthermore, it creates a mutex as "kuku_joker_v3.04" for exclusive access to system resources. It creates a WindowsHook to monitor keyboard activities also.

Suspicious_Gen2 Trojan deletes file DEF~098.TMP, and creates DEF~098.TMP again in C:\WINDOWS\TEMP\ directory.It makes kernlupd.exe and sysfault.exe files in C:\WINDOWS\system32 directory which contains malicious codes. It modifies "WIN.ini"and "KERNELUPD.exe"; furthermore, it creates 6 multiple mutex files for exclusive access to system resources.

Troj_Generic only modifies 2 DLL files which are"WS2HELP.dll" and "WS2_32.dll"

C. Process Activities

Allaple malware creates an event called as VeVT, andcreates some sections with full access permission to everyone.It creates a mutex for exclusive access to system resources.This malware registers processes to be executed at system start, and this could result in unwanted actions to be performed automatically .

Emailworm malware and Suspicious_Gen2 Trojan createsa process as "sysfault.exe" during the execution and it will automatically restart after system boot.

Sality worm creates 3 mutex as "kuku_joker_v3.04", "KUKU300a" and "KUKU301a" for exclusive access to system resources.

VII. SUMMARY

Dionaea low-interaction honeypot is implemented successfully, and this leads to an efficient and effective solution in order to capture the malwares, and with only one operating honeypot thousands of malicious attempts are listened, and 63 known malware, 41 unknown malware, and 4 new samples of binaries are collected, then the captured malwares are analyzed by dynamic analysis. Based on the malware analysis results, the highest percentage malware infection was performed by Allaple.gen worm category.

ACKNOWLEDGMENT

Praise to Allah for giving me the strength to complete this final year project. The project delivered in this dissertation could not have been accomplished without the help of many individuals. I would like to acknowledge the contributions of the following individuals and groups to complete this project successfully.

First, I would like to express my deepest gratitude to my final year project supervisor, Mr. Shadil Akimi Bin Zainal Abidin, for his brilliant guidance, patience, caring, and providing me an excellent atmosphere in order to complete final year project .

I would like to thank the FYP coordinator of Networking systems, Mdm. Arfah Binti Baharudin for the guidance and supports. In addition, I would like to thank the following lecturers at University Kuala Lumpur for the kind guidance and supports: Mdm. Shahrizad Binti Mohd Sharifuddin, and Mdm. Roziyani Rawi.

I would also like to thank my parents, and my sister. They were always supporting me and encouraging me with their best wishes.

Finally, I would like to thank to all my friends at UNIKL MIIT for their wonderful supports.

References

[1] Kumar, S., Sehgal, R., & Singh, P. (2012). Nepenthes honeypots based botnet detection. *JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY,, 3*(4), 215. doi: doi:10.4304/jait.3.4.215-221

[2] Baecher, P., Koetter, M., Holz, T., & Freiling, F. (2006). The nepenthes platform: An efficient approach to collect malware. *Lecture Notes in Computer Science*, *4219*, 165-184. Doi: 10.1007/11856214_9

[3] Raghava, N. S.; Sahgal, D.; Chandna, S., "Classification of Botnet Detection Based on Botnet Architechture," *Communication Systems and Network Technologies (CSNT), 2012 International Conference on* , vol., no., pp.569,572, 11-13 May 2012 doi: 10.1109/CSNT.2012.128

[4] Karimi, A. (2010, November 14). *Use honeypots to know your enemies*. Retrieved from http://www.irhoneynet.org/docs/Use-Honeypots-to-KYE.pdf

[5] Karimi, A. (2013, March). *Honeypots: Challenges and future directions*. Presentation delivered at the 18th national csi computer conference (workshop section), Sharif University of Technology, Tehran ,Iran. Retrieved From http://www.irhoneynet.org/docs/CSI2013_AKarimi_Honeypotworkshop.
pdf

[6] Tiwari, R., & Jain, A. (2012). Improving network security and design using honeypots. *CUBE '12 Proceedings of the CUBE International Information Technology Conference*, 847-852. doi: 10.1145/2381716.2381875

[7] Grudziecki, T., Jacewicz, P., Juszczyk, L., Kijewski, P., & Pawliński, P. (2012, November 20). *Proactive detection of security incidents - honeypots*. Retrieved from http://www.enisa.europa.eu/activities/cert/support/proactivedetection/proactive-detection-of-security-incidents-II-honeypots

[8] *Dionaea*. (2010, September 13). Retrieved from http://carnivore.it/
2009/10/27/introducting_dionaea