Malware Capturing Honeypot: Dionaea

Sarosh Petkar

Under the guidance of Prof. Jonathan Weissman

I. INTRODUCTION

The Internet can be a dark and dangerous, featuring a variety of malware such as worms, viruses and Trojan horses that are aimed at conducting cyber-attacks and as a result stealing personal information such as usernames and passwords. Thus, proliferation of malware poses a significant threat to modern computer technologies. As the size of the Internet is growing, it is becoming increasingly difficult to maintain the security of the Internet especially with the spread of malware using zero-day vulnerabilities.

In such a scenario, security of organization depends upon its ability to keep anti-malware products up-to-date and abreast of current malware developments. However, this tends to be a daunting task as malware has evolved into a powerful instrument for illegal commercial activity. Furthermore, conventional malware sample collection approaches include the extraction of the binary from an infected machine, exchange between Anti-Virus vendors and other similar ways. These traditional methods generally needs human interaction, which can time and resource consuming, hence an automated malware collection mechanism should be used to catch these trends. Thus, the presence of a honeypot can help to detect and obtain metamorphic and unknown malware.

The aim of the project is to collect malware samples using dionaea honeypot in order to understand what is trending in the dark malware underworld. Since in order to deal effectively and efficiently with the threat associated with malware, it is beneficial to obtain a sample of the actual malware in the early stages of propagation. Additionally, as part of the experiment, all the malware communication would be tracked and displayed through the use of a geolocation based visualization module. [8]

Keywords: honeypot, malware, dionaea

II. BACKGROUND

Honeypots are defined as information security resources that are aimed at deceiving malicious users into launching a variety of attacks against servers. These decoy systems are intentionally left vulnerable in order to collect detailed information about the attackers. They are deployed in order to provide some level of protection for organizations or even in some cases as research units to study and analyze the methods employed by attackers to inject malicious software.

Honeypots can be classified as either low-interaction, high-interaction or hybrid. Low interaction honeypots are used to provide limited interaction for an attacker. All the services offered by such a honeypot are emulated, as their sole purpose is to analyze the request and determine its true nature. The advantages of it are its increased speed and low resource consumption.

Whereas, High interaction honeypots are fully functional systems that imitate actual systems and offer the attacker a real system to interact with. This type of honeypot excels at detecting new attacks but at the same time they are more time consuming. Finally, Hybrid honeypots incorporate classification methods used by the other two categories in a cost effective large-scale environment. This system tends to outperform high interaction honeypots. [1]

Interaction	Installation	Deployment	Information gathering	Risk
Low	Easy	Easy	Limited	Low
Medium	Involved	Involved	Variable	Medium
High	Difficult	Difficult	Extensive	High

Table 1: Tradeoffs of honeypots

Related work in this field involves the use of a low interaction honeypot called Nepenthes. It runs on a UNIX server and provides enough emulated vulnerabilities for common Windows services. Nepenthes attempts to download the malicious payload and has the option to submit it automatically to a sandbox. In addition, it provides a report of the characteristics of the given malware.

III. METHODOLOGY

The model consists of a low-interaction python-based honeypot called Dionaea (named after the genus of plants that includes the Venus flytrap) that emulates a vulnerable Windows system in an effort to capture malware samples or interaction from malicious entities. It provides features such as emulating common protocols and deceives the attacker into believing that he is interacting with a real system.

Protocols Dionaea Traps Malware From:

- ❏ Server Message Block (SMB) – SMB is the main protocol offered by Dionaea. SMB is a very popular target for worms.
- ❏ Hypertext Transfer Protocol (HTTP) – Dionaea supports HTTP on port 80 as well as HTTPS. A self-signed SSL certificate is created at startup for HTTPS.
- ❏ File Transfer Protocol (FTP) – Dionaea provides a basic FTP server on port 21. It allows creation of directories, and uploading and downloading of files.

Additional Features:

- ❏ Uses LibEmu to detect and analyze shellcode.
- ❏ Sends real time notifications using XMPP (Extensible Messaging and Presence Protocol).
- ❏ Supports both IPv6 and Transport Layer Security (TLS).
- ❏ Consists of an SQLite3 database that logs attacker information. [4]

Dionaea is deployed on Ubuntu 14.04 Linux server for a period of 4 weeks, it aims to trap malware exploiting vulnerabilities exposed by services offered over a network, and ultimately obtain a copy of the malware. The basic principle of dionaea is to emulate only the vulnerable parts of a service. This leads to an efficient and effective solution that offers many advantages compared to other honeypot-based solutions. After receiving the shellcode, it is analyzed and then downloaded using either FTP or HTTP. Once a copy of a malware is obtained, there is the option to either store the binaries locally, or submit the file to some external services such as Anubis, VirusTotal etc via an API key for further analysis. [6]

Furthermore, the utility DionaeaFR was installed on the Ubuntu machine to help in the analysis of the honeypots activity. DionaeaFR is written in Python, uses the Django framework and a number of other libraries, mostly client-side JS. This front-end visualization module provides a general overview of the malicious connections as well as maps and graphs related to attacker activities.

IV. RESULTS

Immediately after running Dionaea, it will be frightening to quickly discover how much malware there is floating around on the Internet. After about 4 weeks, the honeypot had collected 55 different samples as distinguished by the MD5 hashes of the binaries. Of these 22 were identified as malware by a particular antivirus service (Virustotal). (Figure 1)

Of the known samples, many were Trojans such as GenericKD, WisdomEyes and CoinMiner. The rest were variants of bot families like Spybot, Polybot and others. The majority of binaries, whether classified, as worms or bots had some kind of IRC backdoor functionality. (Table 2) [7]

Moreover, with the help of DionaeaFR specific details about the attacker were obtained such as the IP address, geolocation and methodology of the attacker along with the time of attack.

URL	MD5	Classification
txxp://46.128.183.60/host.exe	fd1fb45d7ca1eeef06f5d46a3e9a3d2f	Backdoor
txxp://46.128.172.44/host.exe	47b2e95136e660522067221ae405025c	Worm
hxxp://87.20.127.177/Photo.scr	ABA2D86ED17F587EB6D57E6C75F64F05	Trojan
hxxp://158.255.6.208/task3.exe	841867D08E7D92E1FB633C61AB421D53	Trojan
hxxp://hibiscus.com.my/g2n-x688-esp/	F8AD349FF58F24D1DDF39F468822B510	Trojan

Table 2: Logged malware in Dionaea

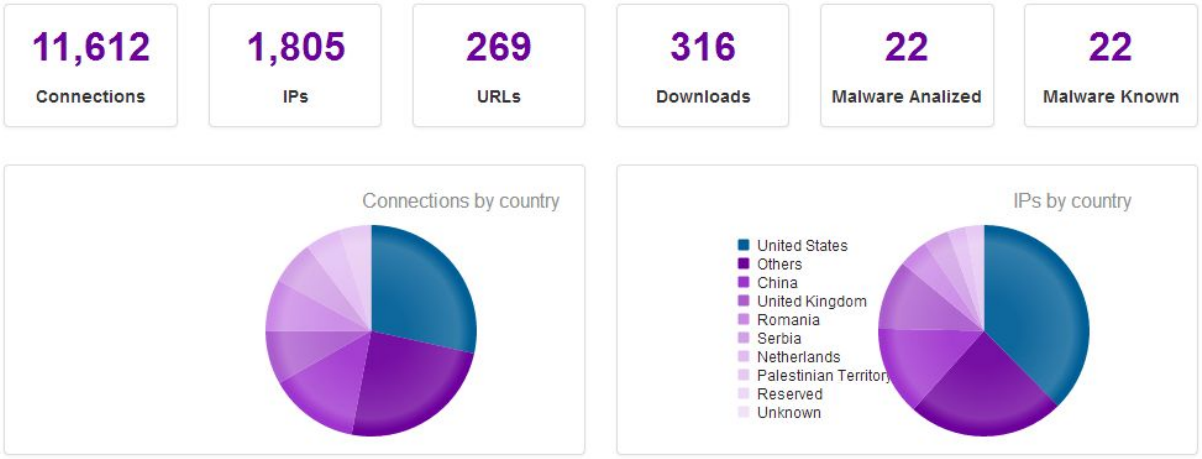
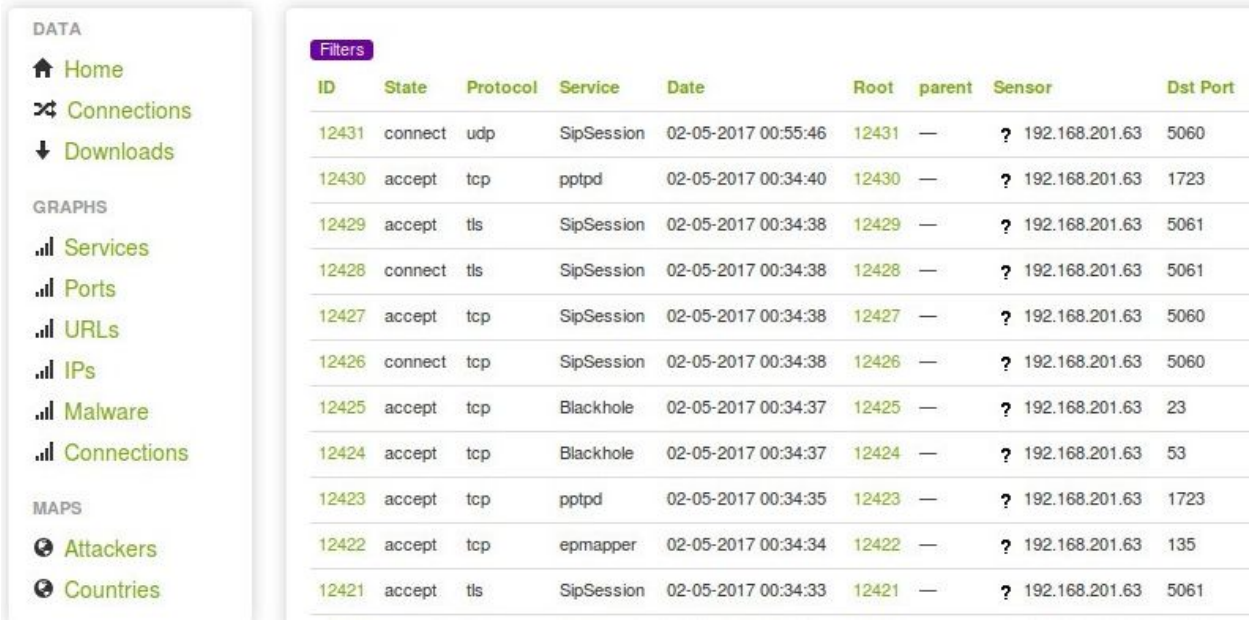


Figure 1: DionaeaFR Dashboard



ID	State	Protocol	Service	Date	Root	parent	Sensor	Dst Port
12431	connect	udp	SipSession	02-05-2017 00:55:46	12431	—	? 192.168.201.63	5060
12430	accept	tcp	pptpd	02-05-2017 00:34:40	12430	—	? 192.168.201.63	1723
12429	accept	tls	SipSession	02-05-2017 00:34:38	12429	—	? 192.168.201.63	5061
12428	connect	tls	SipSession	02-05-2017 00:34:38	12428	—	? 192.168.201.63	5061
12427	accept	tcp	SipSession	02-05-2017 00:34:38	12427	—	? 192.168.201.63	5060
12426	connect	tcp	SipSession	02-05-2017 00:34:38	12426	—	? 192.168.201.63	5060
12425	accept	tcp	Blackhole	02-05-2017 00:34:37	12425	—	? 192.168.201.63	23
12424	accept	tcp	Blackhole	02-05-2017 00:34:37	12424	—	? 192.168.201.63	53
12423	accept	tcp	pptpd	02-05-2017 00:34:35	12423	—	? 192.168.201.63	1723
12422	accept	tcp	epmapper	02-05-2017 00:34:34	12422	—	? 192.168.201.63	135
12421	accept	tls	SipSession	02-05-2017 00:34:33	12421	—	? 192.168.201.63	5061

Figure 2: Connections tab in DionaeaFR

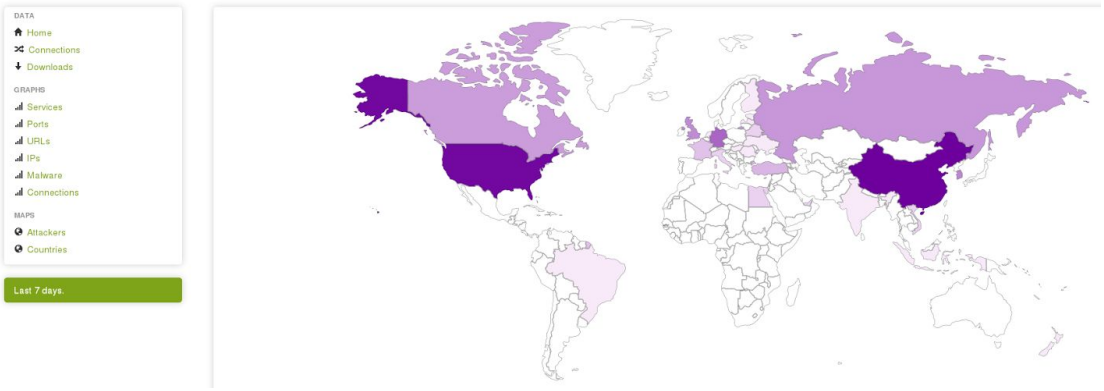


Figure 3: Attacker Map in DionaeaFR

```
student@student-ubuntu:/opt/dionaea/var/dionaea$ cd binaries/
student@student-ubuntu:/opt/dionaea/var/dionaea/binaries$ ls
8d3136ccf8d28d0a8c6ce9077305e681  spoolss-ss_tr_ie.tmp
student@student-ubuntu:/opt/dionaea/var/dionaea/binaries$ ls -la
total 160
drwxr-xr-x 2 dionaea dionaea  4096 May  1 22:04 .
drwxr-xr-x 6 dionaea dionaea  4096 May  1 22:04 ..
-rw------- 2 dionaea dionaea 73802 May  1 22:04 8d3136ccf8d28d0a8c6ce9077305e681
-rw------- 2 dionaea dionaea 73802 May  1 22:04 spoolss-ss_tr_ie.tmp
student@student-ubuntu:/opt/dionaea/var/dionaea/binaries$
```

Figure 4: Sample binary in Dionaea

IP	**Country**
46.128.183.60	Germany
46.128.172.44	Denmark
87.20.127.177	Italy
158.255.6.208	Russia

Table 3: Attacker IP and Country

Using the Dionaea Honeypot, attacks were identified on a number of ports such as UDP, TCP and TLS. Many of the attacks were scanning of typical Microsoft ports with the help of port scanner such as nmap. (Figure 2)

Detailed analysis of the logged information found that the attackers were based over the world, including Germany, Denmark and Italy etc. The major contributors of malware were the USA, China and Russia. (Figure 3)

Overall, the dionaea platform helps to capture thousands of samples of previously unknown and self-replicating malware. These samples can then be transferred over to vendors of host-based IDS/anti-virus systems in order to improve the detection rate of these kinds of threats. (Figure 4).

Dionaea Strengths:

- It is easy to understand the logs that are captured.
- Logs capture the IP address, binary, protocol used along with the time of attack.
- Visualization provided by a front-end plugin.

Dionaea Drawbacks:

- Installation on a specific version of the Ubuntu server can be difficult.
- Requires ample amount of space to store the logs.

V. LESSONS LEARNED & FUTURE WORK

Lessons Learned

In the modern-era malware authors use generators, incorporate libraries, and borrow code from others. Malware also frequently evolves due to rapid modify-and release cycles, creating numerous strains of a common form. The result of this reuse is a tangled derivation of relationships between malicious programs.

With this conundrum, detecting and understanding malware poses significant challenges because of many factors such as:

- ➔ The rate at which new variants are generated from old attacks.
- ➔ The volume of data that must be analyzed to detect or characterize malware.
- ➔ Attempts at obfuscation to thwart the current mechanisms of characterization, detection and attribution. [3]

In the current situation, malware classification is accomplished mostly through manual and semi-automated processes. Generating signatures typically happens either reactively or with

expert knowledge. This mechanism is trivial for malware to evade, and also prone to false classification and attribution. Also, certain AV scanners have been known to identify malware by searching for particular sequences. This motivates malware authors to destroy easily identifiable sequences between releases so that they can avoid detection and has prompted the emergence of polymorphic and metamorphic malware.

In order to solve this problem, phylogeny models must be used to help reconcile naming inconsistencies and assist in the investigation and analysis of new malicious programs. A phylogeny is the evolutionary history or relationships between organisms. Phylogenetic systematics is the study of how organisms relate and can be ordered; using this mechanism, we review sequence comparisons for malware analysis and then separate applications of alternative methods into binary classification and phylogeny generation.

Generating malware phylogeny models using techniques similar to those used in bioinformatics may assist forensic malware analysts by providing clues in terms of understanding how new specimens relate to those previously seen. However, it remains to be seen as to how useful phylogeny models can be built from studying the bodies of malicious programs. The model should be able to account for the types of changes that actually occur in malware evolution such as code rearrangements and instruction or block reordering. [5]

Future work

Future work regarding the dionaea honeypot is to incorporate the ability to automatically infer characteristics from observed malware that are essential for detection and categorization of malware. Such characteristics can be used for signature updates in malware detection tools.

VI. CONCLUSION

One of the key mottos within information security is, “Prevention is ideal, but detection is a must.” We must realize that an organization’s key resources will be attacked, and we have to be ready to detect the attack as early as possible. However, it is imperative to understand that a Honeypot does not replace existing security technologies but works alongside them by monitoring activity occurring on the system. Thus, with the help of a honeypot, collection of the samples as early as possible is the necessary solution for proper treatment of the spreading malware. This way the sample can be analyzed deeply in order to develop accurate detection signatures or a treatment strategy.

VII. RECOMMENDATIONS

It is better to refuse access to intruders than trying to fool them as a honeypot system may be set up by Blackhats to lure regular users to a fake system in order to gain sensitive information such as account number and credit card numbers. Furthermore, there are certain social aspects and legal issues associated with the use of honeypots. Hence, a decoy system should be deployed with great care, consideration and pre-evaluation of all aspects. [2]

VIII. REFERENCES

[1] A. Zarras, "The art of false alarms in the game of deception: Leveraging fake honeypots for enhanced security," *2014 International Carnahan Conference on Security Technology (ICCST)* , Rome, 2014, pp. 1-6.
doi: 10.1109/CCST.2014.6987017

http://ieeexplore.ieee.org.ezproxy.rit.edu/stamp/stamp.jsp?tp=&arnumber=6987017&isnumber=6986962

[2] C. Rong and Geng Yang, "Honeypots in blackhat mode and its implications [computer security]," *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies* , 2003, pp. 185-188.
doi: 10.1109/PDCAT.2003.1236284

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1236284&isnumber=27719

[3] "Classification and Phylogenetics of Malware." (2016): n. pag. Pacific Northwest National Laboratory, Oct. 2016. Web. 14 May 2017.

[4] Donny. "Dionaea – A Malware Collection Honeypot." *Infosec Labs*. N.p., 19 Mar. 2015. Web. 14 May 2017.

[5] Karim, Enamul, Md., Andrew Walenstein, Arun Lakhotia, and Laxmi Parida. "Malware Phylogeny Generation Using Permutations of Code." Springer-Verlag, 1 July 2005. Web. 24 Apr. 2017.

[6] Koniaris, Ioannis. "Analyzing Internet Attacks with Honeypots." (n.d.): n. pag. *Brucon*. Web. 19 Mar. 2017.

[7] Riden, Jamie. "Using Nepenthes Honeypots to Detect Common Malware." Symantec, 07 Nov. 2006. Web. 14 May 2017.

[8] Tan, Emil. "Dionaea – A Malware Capturing Honeypot." N.p., 13 Feb. 2014. Web. 14 May 2017.

[9] https://github.com/DinoTools/dionaea

[10] https://github.com/rubenespadas/DionaeaFR