Honeypots Deployment for the Analysis and Visualization of Malware Activity and Malicious Connections

Ioannis Koniaris[1], Georgios Papadimitriou[1], Petros Nicopolitidis[1], Mohammad Obaidat[2], Fellow of IEEE
[1]Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
{ikoniari, gp, petros}@csd.auth.gr

[2] Department of Computer Science and Software Engineering, Monmouth University, W. Long Branch, NJ 07764, USA
obaidat@monmouth.edu

***Abstract*—Honeypots are systems aimed at deceiving threat agents. In most of the cases the latter are cyber attackers with financial motivations, and malicious software with the ability to launch automated attacks. Honeypots are usually deployed as either production systems or as research units to study the methods employed by attackers. In this paper we present the results of two distinct research honeypots. The first acted as a malware collector, a device usually deployed in order to capture self-propagating malware and monitor their activity. The second acted as a decoy server, dropping but logging every malicious connection attempt. Both of these systems have remained online for a lengthy period of time to study the aforementioned malicious activity. During this assessment it was shown that human attackers and malicious software are constantly attacking servers, trying to break into systems or spread across networks. It was also shown that the usage of honeypots for malware monitoring and attack logging can be very effective and provide valuable data. Lastly, we present an open source visualization tool which was developed to help security professionals and researchers during the analysis and conclusion drawing phases, for use with one of the systems fielded in our study.**

Keywords—honeypot; intrusion detection; malware; data visualization; log file analysis

I. INTRODUCTION

In a globally networked environment where most of society's institutions and even human lives depend on the fault-tolerance attributes and robustness of the Internet, the security of our networks is of vital importance. The current state of the Internet is threatened by malicious adversaries who can disrupt its optimal operations. Human and software threat agents launch attacks against the servers or network infrastructure of various organizations, institutions and home users. The motivations behind these attacks can range from financial gains to corporate espionage or state-sponsored cyber warfare.

Attackers and malicious software with auto-propagating capabilities are constantly scanning the Internet for vulnerable targets. These might include servers or personal computers susceptible to a certain type of attack. A human intruder usually tries to detect running services on a system in order to select his or her attack vector. Websites like Exploit-DB [1] archive thousands of different vulnerabilities containing what is commonly referred to as "POC", a proof of concept code that can be run against a vulnerable system and take control of it or act in a unlawful way.

Malicious software, also referred to using the abridgement malware, can scan internal or external network resources in order to find a specific vulnerability to exploit using some form of payload that allows the newly infected system to act in the same way [2]. This type of software that possesses auto-propagating abilities is commonly known as a worm. Worms can spread in various fashions and many studies have been conducted in order to form certain models applicable to their spreading. For example, almost twenty years ago, Kephart and White presented the Epidemiological model to measure computer viruses prevalence [3], [4]. Other authors have presented a modeling methodology using Interactive and Networked Markov Chains called the Influence Model [5], [6]. One other study was based on the existing models and has presented a mathematical model called AAWP which stands for Analytical Active Worm Propagation [7].

Human-driven attacks can be classified as opportunistic or targeted. An opportunistic attack is one that happens out of luck for the attacker or by following a random pattern. For example an attacker can use various scanning tools to automate the process of finding weaknesses in a random network subnet. He or she then proceeds to exploit the specific discovered vulnerabilities; something that might not have happened if the threat agent has chosen a different B or C class IP address, or if the victim has had setup defensive countermeasures. On the contrary, a targeted attack is one that systematically and covertly scans the network infrastructure and/or web applications of a specific target in order to find a way to bypass security mechanisms and gain illegal access. These types of attacks can take much longer time and are usually more catastrophic in the case of successful exploitation. Their targets are usually high-profile companies, organizations, or state-sponsored institutions. One of the most prominent targets of human attackers is the Secure Shell service commonly referred to as SSH. Secure Shell is a remote access service running on network-connected servers. Many times this service can be exploited by malicious users if a weak password has been chosen by the system administrator for authentication purposes. Several studies on this phenomenon have been undertaken in recent years, both as a result of academic work [8], [9] and efforts from information security professionals [10], [11].

In order to detect these attacks, get early warnings of new ones and study how human and software threat agents operate, one can use devices known as honeypots. A honeypot in its simplest form is a decoy-based intrusion detection system. A more formal and accurate definition comes from Lance

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

Spitzner who defines a honeypot as an information system resource whose value lies in unauthorized or illicit use of that resource [12]. By design a honeypot should not receive legitimate traffic and any attempt to communicate with it can be considered malicious [13]. Furthermore, a honeypot that tries to communicate with an external network resource can serve as an indicator of an intruder's presence inside the honeypot in particular and the network in general [14]. Honeypots cannot prevent attacks against the network they are attached to, but they can help in the identification and detection phases of unlawful actions if they are combined with other defensive tools such as firewalls and intrusion detection systems. One of the advantages of honeypots is that they often generate a small amount of data of high value [15], but depending on the circumstances the analysis of a captured dataset can present a challenge to analysts if it becomes large over time. In situations like these, security researchers can use visualization tools and techniques in order to get a better understanding of the honeypot's operation.

In this paper, we focus specifically on using honeypots to capture malicious connection attempts targeting various system ports and simulated fake services, and also to assess the current state and monitor malware spreading activities. We present the results of a rather lengthy experiment consisting of more than one phases, and then proceed to the visualization of the captured data using a combination of tools that were created for this purpose.

The remainder of this paper is organized as follows. After presenting a brief overview of Related Work in Section II, Section II describes the experimental setup used in order to test the effectiveness of honeypots. Setion IV overviews the malware activity that was captured by the deployed system, while Section V discusses the malicious connections that were identified. Finally, Section VI discusses ideas for future work and concludes the paper.

II. Related Work

A significant number of studies have been undertaken regarding the visualization of attacks on computer systems and networks. Most of the times the aforementioned projects focus on visualizing attack logs created by Intrusion Detection Systems (IDS) with a combination of NetFlow data. Two significant tools in this field are NFlowVis presented by Fisher et al [16] and VIAssist presented by Goodall and Sowul [17] which serve distinct purposes and can be considered a good basis for new tools and extensions.

More related to honeypots are a number of projects developed by information security professionals or enthusiasts. For example, J. Blasco presented a visualization tool for the Nepenthes honeypot [18]. Nepenthes is a what is known as a malware honeypot; a utility that can simulate vulnerable services targeted by malware in order to safely capture binaries of auto-propagating software such as worms. The aforementioned visualization utility makes use of the AfterGlow and Graphviz visualization libraries in order to create interactive graphs that show the correlations between attack sources, malware samples and geolocation information. Another visualization project is one called carniwwwhore and was built for the Dionaea honeypot [19]. Carniwwwhore is designed as a web interface for Dionaea's Postgres database. Dionaea is considered to be the successor of Nepenthes and the aforementioned visualization tool utilizes its XMPP backend in order to present some statistics of the honeypot's activity. The creator of Dionaea has included some rather basic visualization tools in the software, but these proved to be infective when the dataset grows large in size. Recently a new utility was developed for the same honeypot system, called DionaeaFR [20]. It is written in Python and utilizes the Django web framework. This particular project was evaluated in our experiment during the visualization of the results of our Dionaea instance and was found to be the most effective.

A tool previously created by our team is Kippo-Graph [21], which visualizes the logged MySQL data of a Kippo SSH honeypot [22]. Some other scripts were shared by J. Gimer on the formerly active “Honeypots” mailing list and can be used to visualize logs by Honeyd [23], a well known low-interaction honeypot. Honeyd has a number of other visualization utilities associated with it, but none of them are still maintained or work in an easy way. Some of these are HoneyGraph and HoneyView. Under these circumstances, we decided to create a new tool for this particular honeypot drawing from the experience gained while previously developing Kippo-Graph.

III. Project Overview

A. Experimental Setup

In order to test the effectiveness of honeypots we built the following setup. Firstly, a Dionaea instance was installed on a virtual private server (VPS) with a static public IP address. This particular honeypot played the role of a malware catcher or collector. These types of devices are usually deployed by malware analysts in order to securely store new and unknown pieces of auto-propagating software. These binaries can later be statically and dynamically analyzed in order to study the architectural design and the capabilities that each possesses. Dionaea is a medium-interaction honeypot [24] and is innovative in the sense that it doesn't emulate specific network vulnerabilities, but rather the whole underlying network protocol stack. The main protocol offered for exploitation to threat agents is SMB (Server Message Block) also known as CIFS (Common Internet File System), which uses the well known port 445. Dionaea can also emulate other protocols but doesn't allow much interaction. Some of them are MSSQL (Microsoft SQL Server), MySQL, FTP, HTTP(s) and SIP. The aforementioned system was in operation during three distinct periods: 20 days (February 19 to March 11, 2012), then another 25 days (March 28 to April 23, 2012) and lastly 30 more days (January 21 to February 19, 2013). These periods were selected in order to introduce a small and a big pause in the normal operation of the malware catcher in order to investigate potential significant changes in the patterns. Dionaea logs all the data from its operation in a SQLite database stored locally, which was also frequently backed-up in our lab to avoid an event of data loss or corruption. This database can grow very large in size and essentially can make the analysis of the results a great challenge for the information security researcher. For example, the average size of the three databases was 560MB. This is where data visualization solutions can play a vital role

in helping the defenders get a quick and detailed overview of the honeypot's captured activity.

Secondly, a Honeyd low-interaction honeypot instance was installed in a virtual machine residing inside a home-based network behind a DSL line. A router with a static public IP address was configured to forward every connection request back to this system where it was logged for later analysis. Honeyd allows the creation of simple or complicated network topologies, but in our case we configured the system to emulate an old, seemingly unattended Linux server running the following open services: FTP, Telnet, SMTP, HTTP, POP3 and IMAP. This system played the role of a simple malicious attempts detector and logger and operated for a period of 50 days (April 13 to June 1, 2012).

B. *Visualizing the Results*

In our experiment we tested the effectiveness of most of the visualization utilities referenced in the previous section. We found DionaeaFR to provide the best solution currently available for the Dionaea malware honeypot. In this paper, we also present a new tool called Honeyd-Viz [25]. This visualization utility is released under an open source license and is written in PHP. It uses the Libchart chart drawing library and the QGoogleVisualizationAPI library to interact with the Google Maps project. As of this writing Honeyd-Viz can create 20 different charts depicting various statistics of Honeyd's operation. Honeyd-Viz cannot be used on its own, as it is designed to use a MySQL database where it looks for data to visualize. Due to this we also created a simple open source Perl script called Honeyd2MySQL [26]. This tool parses Honeyd's text-based log files and imports all the captured malicious events in a MySQL database. For this reason a standalone MySQL database server was setup in our lab in order to store the final data.

IV. Overview of Malware Activity

A. *First Period of Operation (February - March 2012)*

During the first implementation period of our malware catcher (Feb. 19 to Mar. 11 2012), we directly observed activity that can be characterized as intense. In total there were almost half a million distinct connection attempts, and more precisely 510,368. From those, 401,636 targeted the SMB/CIFS service, coming as no surprise since this protocol has a history of vulnerabilities and is the main way Dionaea gathers results. A significant number of connections were also observed for MSSQL (Microsoft SQL Server) that was targeted 486 times. A very low number of others were observed for MySQL, FTP and EPMAP services. Interesting findings were 12 connections arriving at port 9988, which is usually being used for illegal remote control of compromised machines by the Rbot/Spybot malware family. Some of the results of the honeypot's operation regarding the number of connections are presented in Table I.

The aforementioned connections were originated from compromised systems across the Internet, where auto-propagating malware were trying to spread. Each of these connections attempted to send a copy of the worm to our honeypot instance, many times with successful outcome. Dionaea stores these binaries in a special directory for further analysis. In total, 191,566 malicious binaries were offered to our system for download. From these, it was possible to retrieve successfully 172,564. Failure to get the binary is mostly due to the use of NAT (Network Address Translation) on the attacking computer's end, which is also a victim of previous compromise.

TABLE I. NUMBER OF CONNECTIONS BY PORT (1ST PERIOD)

Service	Port	Number of connections
SMB	445	401,636
MSSQL	1433	486
MYSQL	3306	23
FTP	21	13
EPMAP	135	13
Rbot/Spybot	9988	12

TABLE II. "TOP 5" ATTACK SOURCES (1ST PERIOD)

IP address	Geolocation	Number of connections
60.42.182.151	Toyota, Japan	29,827
79.140.162.227	Moldova	25,091
178.140.113.207	Moscow City, Russia	20,834
122.125.52.100	Nantou, Taiwan	19,667
95.24.108.249	Moscow City, Russia	12,872

The average number of offers during the 20 days of the first period of operation was 8,628 per day. Getting all of these malware binaries is not very practical when it happens to be the same sample many times over. Dionaea can calculate the unique hashes of the files and store only distinct samples. From the total 172,564 offers, our system downloaded a mere 508 distinct and unique malware binaries; totaling 141,8MB in size.

This is not to say that this number is insignificant, on the contrary it appears to be a rather successful outcome. The well-known Conficker worm [27] dominated the top list of logged malware submissions with many of its variants present. It appears that Conficker is still very active and the number of attacks it launches is very significant for the security of the Internet as a whole. Dionaea can work alongside the well known VirusTotal web service [28] and automatically submit downloaded malware samples to it for automated analysis. The results are returned and logged inside the SQLite database for each distinct download. In total, VirusTotal was able to successfully identify 489 out of the 508 files, after conducting a number of 20,931 checks using various anti-virus engines. It is interesting to observe that different companies classify the same dataset in malware categories with big deviations. One anti-virus engine identified 105 distinct samples while others identified 82 and 11.

By summarizing the results, almost all the downloaded files had some connection to the Conficker worm, with around 10 different additional malware samples like the Allapple worm, Rbot and Jorik IRC bot. The aforementioned malwares were offered from 2,251 distinct IP addresses. A significant finding is that a great geographical dispersion was observed for the

origins of these addresses. A "top 5" list along with geolocation information and the respective number of logged connections is presented in Table II. Regarding the underlying operating system running on those compromised machines, Dionaea can work alongside the well known p0f tool [29] that passively fingerprints operating systems. Out of the total 510,368 connections, p0f logged results for 510,022 with Windows OS ending up at 90.1%.

B. Second Period of Operation (March – April 2012)

During the second implementation period of our malware catcher (Mar. 28 to Apr. 23 2012), even more intense activity was observed against the honeypot. In total there were 771,919 logged connections and 616,182 of those targeted the SMB service. MSSQL received 1,105 connections, more than double than the first period. Some of the rest services and port pairs are presented in Table III. In total, 265,573 malware samples were offered to the system, and 238,273 of those were successfully downloaded. The average number of offers distributed along the duration of this second time period was 9,530 files per day. As described in the previous section, Dionaea only stores unique files, and so we ended up with 677 distinct malware samples totaling in 210,1MB in size. VirusTotal was able to successfully identify 509 out of 677 files after conducting 21,776 checks using various anti-virus engines.

Once more Conficker dominated the top list of malicious files, with around 10 different additional malware samples such as Allapple and Deborm worms, Virut and Rbot. The aforementioned malware offers came from 3,340 distinct IP addresses, a number that is higher by 1,090 from the previous one. This is a significant discovery considering the fact that the second period lasted only 5 days longer than the first one. Geographical disparity is still high and there are attackers from many places around the globe. A "top 5" list of attacking IP addresses is presented in Table IV. Regarding the underlying operating system running on those compromised machines, p0f was able to identify 771,029 out of 771,919 connections with Windows OS present in 92.3% of the remote hosts.

C. Third Period of Operation (January – February 2013)

During the third implementation period of our malware catcher (Jan. 21 to Feb. 19 2013) 506,308 distinct connections were observed against our honeypot, which is lower from the previous period and lower even from the first period. From those, 367,662 targeted the SMB service. MySQL received 1,337 connections and MSSQL received 1,025. Some results concerning the connection types are presented in Table V. In total, 175,066 offers of malware binaries were submitted to the system and from these it was possible to download 119,804. The average "offers per day" value was 5,535 which is significantly lower than the two previous periods of operation.

As described above we are only interested in distinct samples, and as a result Dionaea stored 573 unique binaries, totaling 189,05MB in size. Coming as no surprise, Conficker dominated the results for a third time while some other insignificant malwares were still captured like in the previous operational periods, such as Allapple, Virut and Rxbot. All the aforementioned malware offers came from 4,469 distinct IP addresses, an interesting discovery since it is a number higher by 1,129 compared to the previous one, despite the overall less intensive activity. A "top 5" list of attacking IP addresses are presented in Table VI. Regarding the underlying operating system of the attacking hosts, we must stress that p0f was able to identify only 192,531 out of the 506,308 connections, a rather low performance result for the third period of operation. Strong conclusions cannot be drawn from this percentage, but a prevalence of the Windows operating system cannot be overlooked.

TABLE III. NUMBER OF CONNECTIONS BY PORT (2ND PERIOD)

Service	Port	Number of connections
SMB	445	616,182
MSSQL	1433	1,105
Rbot/Spybot	9988	27
EPMAP	135	26
WINS	42	19
MYSQL	3306	17

TABLE IV. "TOP 5" ATTACK SOURCES (2ND PERIOD)

IP address	Geolocation	Number of connections
88.158.151.88	Romania	15,622
42.71.158.222	Taipei, Taiwan	14,459
84.51.95.14	Odintsovo, Russia	13,307
190.106.4.163	Managua, Nicaragua	12,303
78.153.27.84	Yekaterinburg, Russia	11,268

TABLE V. NUMBER OF CONNECTIONS BY PORT (3RD PERIOD)

Service	Port	Number of connections
SMB	445	367,662
MYSQL	3306	1,337
MSSQL	1433	1,025
EPMAP	135	52
FTP	21	27
TFTP	69	6

TABLE VI. "TOP 5" ATTACK SOURCES (3RD PERIOD)

IP address	Geolocation	Number of connections
61.221.55.46	Taichung, Taiwan	234,000
5.152.210.197	United Kingdom	27,220
208.126.16.216	Jefferson, IA, USA	25,207
46.233.24.218	Sofia, Bulgaria	23,261
89.137.172.153	Constanta, Romania	18,620

V. Overview of Malicious Connections

Malicious connections other than malware activity were logged by the residential honeypot system based on Honeyd, which remained online for a period of 50 days; April 13 to June 1, 2012. During this experiment the honeypot logged 20,768 distinct events. Each of these connections used one of the TCP, UDP or ICMP protocol, and more precisely their breakdown is: 11,435 TCP connections, 9,253 UDP and a mere 80 for ICMP. These results are depicted in Fig. 1 and Fig. 2 as generated by the visualization tool Honeyd-Viz. From the aforementioned connections it was observed that all UDP packets originated from devices belonging to the internal network. Hence, we can conclude that no malicious UDP attacks took place. Activity and connections against the honeypot system were not evenly balanced during the period of its operation. It was observed that five dates had a significantly larger number of connections than the average daily activity. Overall, most of the connections received by the honeypot took place during the 10 last days of May 2012. A "top 20" chart displaying the dates with the greatest number of connections is presented in Fig. 3, while Fig. 4 and Fig. 5 present the daily and weekly activity charts for the whole duration.

TCP connections were by far the most interesting since they can be all classified as malicious. From the various services that our system emulated (FTP, Telnet, SMTP, HTTP, POP3 and IMAP) the most popular in terms of malicious connections destined to it was the POP3 email service. This service was listening on port 110 and received 5,355 connections in total. Telnet was the second best with a significant number as well; 4,717 against its port 23. HTTP is following in the list with 880 connections.

A "top 10" chart with the busiest attackers is presented graphically in Fig. 6 along with the two letter country code of their origin. Regarding these IP addresses, if one were to perform a reverse DNS operation on them, a task supported by Honeyd-Viz, he or she could find that most of them belong to net blocks owned by relatively known web hosting companies. Some of these IP addresses were in use serving legitimate websites such as an e-shop and blogs. Based on this we can draw the conclusion that most of the attacking systems are victims themselves, which have been previously exploited by malicious users. Lastly, Fig. 7 depicts some of the advanced features of Honeyd-Viz where it points the locations of the top 10 IP addresses on a Google Map along with their information.

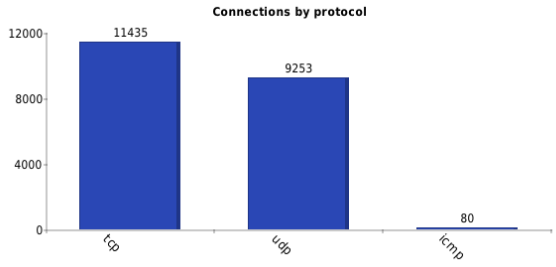

Fig. 1. Number of connections per protocol observed during our honeypot's operation.

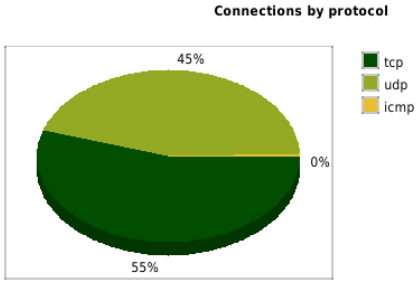

Fig. 2. Number of connections per protocol observed during our honeypot's operation, as a pie.

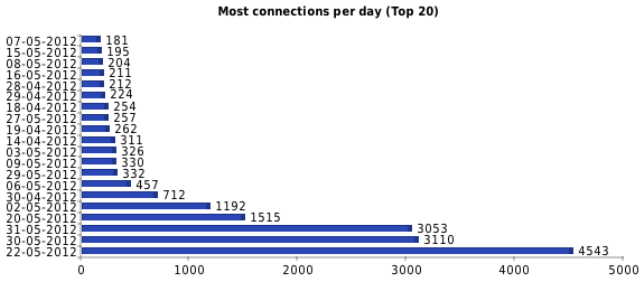

Fig. 3. Dates with most connections observed during our honeypot's operation.

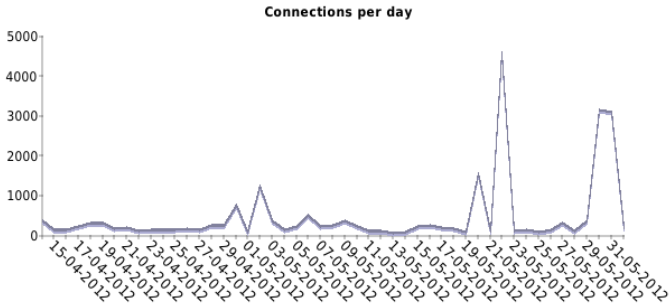

Fig. 4. Number of connections observed per day for the whole duration of our honeypot's operation.

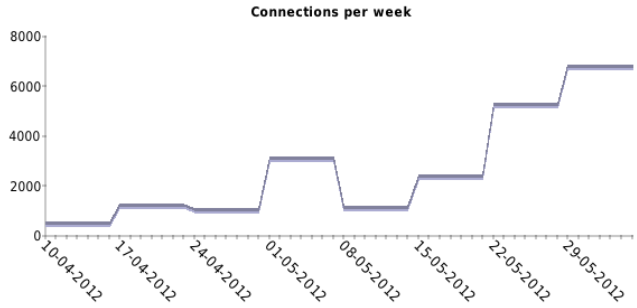

Fig. 5. Number of connections observed per week for the whole duration of our honeypot's operation.

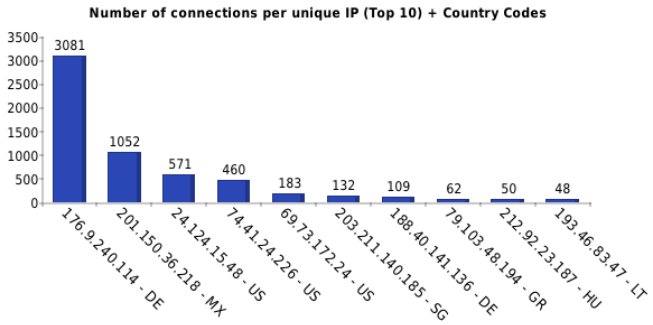

Fig. 6. Number of connections observed per unique IP address during our honeypot's operation along with the two letter country code of their origin.

Fig. 7. The geographical location of the top 10 source IP addresses based on the number of attacks, drawn on a Google Map.

VI. CONCLUSIONS

Dionaea and Honeyd provide an excellent basis for extended experiments. Something that would be interesting to observe is distributed honeypots operating in sync and sharing their results. Our team is working on another open source project called HoneyDrive [30]. It is essentially a virtual hard disk offered in an open virtualization format, ready to be imported in a virtual machine hypervisor. It comes with many honeypot and malware related utilities pre-installed and pre-configured to work in an 'out of the box' fashion. A number of virtual machines like these could be set up in different IP address spaces in order to capture data and accumulate different datasets for analysis. The data logged from an operation of this kind could be used to create blacklist databases. The final results could also be embedded in suites of software used in active or passive network defense.

During our experiment it was shown that honeypots still present a unique security concept. They can aid in the malware analysis process, catching and securely storing malicious binaries or provide an early warning system for attacks and capture new and unknown exploits. In this paper we have used two honeypot systems to setup a malware catcher on one hand and a malicious connections logger on the other. A relative lengthy experiment was run which yielded interesting results. It was shown that auto-propagating malware are constantly launching attacks with great intensity. These can occur in a random fashion and no IP address space seems to be safe. Furthermore, malicious users are compromising systems and use them as pivots; launch pads for more attacks against other systems. Using honeypots in conjunction with other defense-oriented systems such as firewalls and intrusion detection systems can be a good first step for risk mitigation and early warnings of cyber attacks.

REFERENCES

[1] Offensive Security, "Exploits Database." [Online]. Available: http://www.exploit-db.com/.

[2] M. S. Obaidat and N. Boudriga, "Security of e-Systems and Computer Networks." Cambridge University Press, Cambridge, UK, 2007.

[3] J. O. Kephart and S. R. White, "Directed-graph Epidemiological Models of Computer Viruses," in Proc. of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy, 1991, pp. 343–359.

[4] J. O. Kephart and S. R. White, "Measuring and modeling computer virus prevelance," in Proc. of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy, 1993, pp. 2–15.

[5] C. Asavathiratham, "The Influence Model: A Tractable Representation for the Dynamics Networked Markov Chains," Massachusetts Institute of Technology, 1996.

[6] M. Garetto, W. Gong, and D. Towsley, "Modeling Malware Spreading Dynamics," in Proceedings of the IEEE INFOCOM 2003, 2003.

[7] Z. Chen, L. Gao, and K. Kwiat, "Modeling the Spread of Active Worms," in Proceedings of the IEEE INFOCOM 2003, 2003.

[8] D. Ramsbrock, R. Berthier, and M. Cuckier, "Profiling Attacker Behavior Following SSH Compromises," in Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007, pp. 119–124.

[9] J. Owens and J. Matthews, "A Study of Passwords and Methods Used in Brute-Force SSH Attacks," Clarkson University, 2008.

[10] C. Seifert, "Analyzing Malicious SSH Login Attempts," Security Focus, Infocus 1876, 2006. [Online]. Available: http://www.securityfocus.com/infocus/1876.

[11] "Observations of Login Activity in an SSH Honeypot," Cisco Security Intelligence Operations, 2009. [Online]. Available: https://www.cisco.com/web/about/security/intelligence/ssh-security.html.

[12] L. Spitzner, "Honeypots: Catching the Insider Threat," in Proceedings of the 19th Annual Computer Security Applications Conference, 2003.

[13] L. Spitzner, Honeypots: Tracking Hackers. Boston, MA: Addison Wesley, 2003.

[14] L. Spitzner, "Strategies and issues: Honeypots - sticking it to hackers," Network Magazine, 2003.

[15] A. Obied, "Honeypots and Spam." University of Calgary, 2006.

[16] F. Fisher, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel, "Large-scale network monitoring for visual analysis of attacks," in *Proceeding VizSec '08 Proceedings of the 5th international workshop on Visualization for Computer Security*, 2008, pp. 111 – 118.

[17] J. R. Goodall and M. Sowul, "VIAssist: Visual analytics for cyber defense," in IEEE Conference on Technologies for Homeland Security, HST'09, 2009, pp. 143–150.

[18] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, "The Nepenthes Platform: An Efficient Approach to Collect Malware," in *Recent Advances in Intrusion Detection (RAID)*, 2006, pp. 165 – 184.

[19] "Dionaea honeypot." [Online]. Available: http://dionaea.carnivore.it/.

[20] "DionaeaFR." [Online]. Available: http://rootingpuntoes.github.com/DionaeaFR/.

[21] "Kippo-Graph." [Online]. Available: http://bruteforce.gr/kippo-graph/.

[22] "Kippo honeypot." [Online]. Available: https://code.google.com/p/kippo/.

[23] N. Provos, "A Virtual Honeypot Framework," in *Proceedings of the 13th conference on USENIX Security Symposium*, 2004.

[24] G. Wicherski, "Medium Interaction Honeypots." 2006.

[25] "Honeyd-Viz." [Online]. Available: http://bruteforce.gr/honeyd-viz.

[26] "Honeyd2MySQL." [Online]. Available: http://bruteforce.gr/honeyd2mysql/.

[27] CWG, "Conficker Working Group: Lessons Learned," 2010.

[28] "VirusTotal." [Online]. Available: http://www.virustotal.com/.

[29] "p0f." [Online]. Available: http://lcamtuf.coredump.cx/p0f3/.

[30] "HoneyDrive." [Online]. Available: http://bruteforce.gr/honeydrive/.