Container-Based Honeypot Deployment for the Analysis of Malicious Activity

Andronikos Kyriakou, Nicolas Sklavos

SCYTALE Research Group,
Computer Engineering & Informatics Department,
University of Patras, Hellas
e-mails: {kyriakou, nsklavos}@ceid.upatras.gr

***Abstract*— In today's world, the field of cyber security is a fast-paced changing environment. New threats are continuously emerging, and the ability to capture and effectively analyze them is paramount. In our work, we are deploying multiple honeypot sensors in order to monitor and study the actions of the attackers. The selected honeypots are Cowrie, Dionaea and Glastopf, presented as a Linux host, a Windows host and a Web application respectively. This enables us to have a diverse and broad environment that can attract attackers aiming at different attack surfaces. The sensors are running on a containerization platform, Docker and in this way, they are lightweight, resilient and could be easily deployed and managed. Our goal is the creation of a single dashboard that can present the captured data effectively in real-time and both in macroscopic and microscopic levels. Thus, we are utilizing the Elastic Stack and we are enriching our data sources using Virus Total's analysis engine. The proposed system ran for a three-month period and provided numerous data points, from which instantaneous useful conclusions were drawn for the behavior and nature of the malicious users.**

Keywords—honeypot, Docker, virtualization, intrusion detection, malware, Cowrie, Dionaea, Glastopf, Elastic Stack

I. INTRODUCTION

In recent years, there has been unprecedented growth in the number of devices connected to the Internet. Reports suggest that the number will reach 100 billion by 2025 [1]. One emerging key challenge is the security of these devices. As suggested in [2], [3] cybercriminals are shifting their attention to this new era of computing, the Internet of Things (IoT), and are finding new ways to exploit it. In the past years, events such as the creation of Mirai Botnet and the overwhelming Distributed Denial of Service attacks that can achieve 1.1 Tbps as in [4], have highlighted major misconfigurations and have drawn special attention to the extensive damage that can be caused. Therefore, the need for real-time, cost-conscious, both in resources and capital, monitoring and protection of a wide range of devices is becoming more imperative than ever.

Based on [5], honeypots are decoy systems placed in a network (internally or externally), that earn their value by being attacked and breached. In this way, the interaction with the attackers is logged and used in order to deduct valuable information for their techniques. For a better assessment of a honeypot's added value in the security posture of an organization, we could categorize security in prevention, detection and response, as in [6]. Honeypots, being bait systems, do not act as a preventive measure, but could be used complementary to already deployed arrangements, in order to detect specific aspects of the targeted attacks and help in the phase of response.

In this work, we implemented a multi-component honeypot system that logs connections to numerous popular protocols. A wide range of services is exposed and, in this way, a heterogeneous system is presented that aims to mimic the available endpoints in a production environment. The honeypots are running on Docker and thus the management and orchestration are efficient and undemanding. Also by using a container-based approach, a low system load is achieved. Virus Total's analysis engine is integrated by automatically sending all collected malware and incorporating the responses in the analysis phase. In order to create an overview of the attacks, we are exploring the usage of the Elastic Stack for the analysis and visualization of all the collected data. This enables us to stream our observed connection attempts and malware samples to a central analysis server and be notified in real–time. Ultimately, the system under review acts as an invaluable information gathering and analysis platform that could be deployed to multiple diverse interconnected devices. As highlighted in Section III, enhanced analytics of the occurring security-related events could be constructed. These events could be gathered in a centralized manner and help in an organization's risk assessment, as well as, for providing a competitive advantage in the race against the attackers.

The rest of the paper is organized as follows. Section II outlines the implemented system. Section III presents an analysis of the captured data. Section IV mentions related work. Finally, Section V addresses our conclusion and the future work.

II. PROPOSED SYSTEM SETUP

An overview of the proposed system is presented in Fig. 1. For our work, we selected three honeypots, both low and medium interaction, offering a wide range of services.

Considering the often limited capabilities of IoT devices we decided to run our selected honeypots using operating system

978-1-5386-7272-3/18/$31.00 ©2018 IEEE

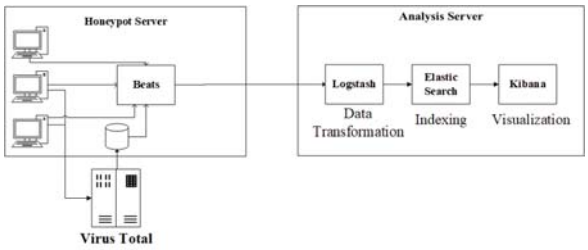

Fig. 1. Architecture of the Proposed System Setup

level virtualization, also known as containerization. A container is an isolated virtual environment that packages together an application and its dependencies. Containers are more lightweight than virtual machines as they can share the kernel and required libraries with the underlying operating system. Also, by running as processes in user space they achieve a low system load. For the purposes of our experiment, we used Docker. Docker natively supports many architectures such as x86-64 and ARM so it fulfills our requirement for interoperability. This selection also enabled us to utilize tools like Docker Compose and in this way have control over the load balance on the target system as well as easily configure, deploy and orchestrate our instances using its API [7].

The first deployed honeypot is Cowrie, a medium interaction SSH and Telnet honeypot that is the successor of the Kippo Honeypot. It is written in Python and provides a modular fake file system, as well as, a fake shell. There is, also, support for downloading files using SFTP for later examination. In the latest version, Cowrie introduces fake TCP/IP tunneling in order to catch proxy requests and the corresponding data. All interaction is stored in JSON format and all created sessions are saved [8].

Additionally, Dionaea is a low interaction honeypot, developed by the Honeynet Project that emulates and supports many protocols such as FTP, TFTP, HTTP, HTTPS (using a self-signed certificate), MQTT, MSSQL, MySQL, SIP, SMB and UPnP. Dionaea is able to capture malware samples by utilizing libemu library for x86 emulation. Having detected a shellcode, it runs it in its native virtual machine and records all API calls and arguments. This honeypot logs connection details in JSON format, but also creates and updates an SQLite database with information about every interaction [9].

Last but not least, Glastopf is a low interaction web application honeypot that doesn't emulate specific vulnerabilities. Its principle is to respond to an attacker with the expected answer in order to convince them of a vulnerability. It has some attack types included such as remote file inclusion, local file inclusion and HTML injection via POST requests but it continuously expands its attack surface by extracting keywords from offensive queries and creating dork pages. These dork pages are fed to crawlers in order to lure more attackers that are using search engines. Glastopf logs all connections to an SQLite database [10].

For the purpose of having a unique input source format, we developed a simple Python script that queries Glastopf's database and saves connection data in JSON format. Also, in order to analyze the malware collected, we developed another Python script that sends all captured samples to VirusTotal [11] via its API, and stores the scan results in a MySQL database. Lastly, we created yet another tool to query the aforementioned database and convert the data to JSON format. All the above scripts, as well as the destruction and recreation of the containers using Docker Compose, were run periodically using cron jobs.

The open source Elastic Stack was used for the analysis of the gathered connection logs. On the honeypot server, we installed Beats, a shipping utility that collects all sources and streams them using public key cryptography to our analysis server. On the analysis server, we deployed Elastic Stack and configured it to receive the encrypted logs. Elastic Stack consists of three independent components forming a pipeline. The logs are first imported to the pipeline using Logstash where they are converted from an unstructured to a structured format and are being processed using the available filters. Then, they are ported to Elastic Search, a scalable, distributed full-text search engine that uses a REST API for all operations. Finally, Kibana is utilized to visualize the data and create an all-in-one dashboard.

The honeypots were deployed on a Virtual Private Server run by Greek Research and Technology Network (GRNET), with 4 GB RAM, 40 GB storage space and an assigned static IP. The host operating system was Ubuntu 16.04.4 LTS. The Analysis server was hosted on a remote cloud provider and had similar specifications. The system ran totally for 92 days, from April 15, 2018 to July 15, 2018. In this period of time, for the first 25 days Cowrie's SSH service was listening to port 65534 and after May 10, was moved to the default port 22 to allow for easier discovery. The majority of the provided services were running with their default configuration for the same reason. Overall, the proposed system consists of the most popular operating systems and services and in this way creates an attractive attack surface.

III. Analysis Of The Captured Data

The total number of attempted connections captured by the honeypots was 2,750,654. As shown in Fig. 2, the majority of the connections came to the Dionaea honeypot. It is observed that there was an immediate increase after the movement of the SSH service to its default location. This observation confirms our initial hypothesis that there is a small number of agents scanning the full range of ports at each machine and that the majority are trying to exploit default setups.

Also, considering the data of Fig. 3, we can suggest that most attacks targeted the Server Message Block (SMB) protocol, followed by the Secure Shell (SSH) protocol.

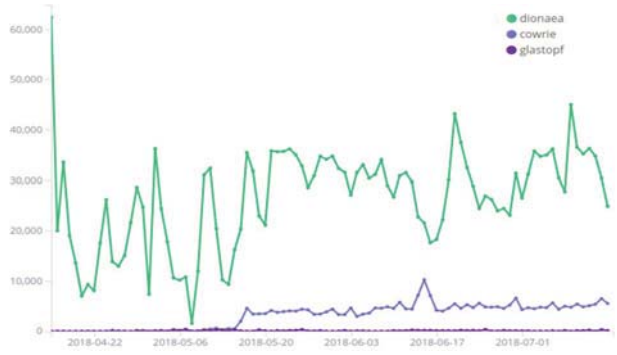

Fig. 2: Connections per day

The majority of the created sessions come from Russia and Vietnam. The sources presented are the last hop of attack because we could not infer from the logs if the hosts contacting the honeypot are compromised and used as proxies. We should also consider that Fig. 4 depicts the observed connections in total and as a result, connections to Dionaea that are larger in volume, are a ruling factor.

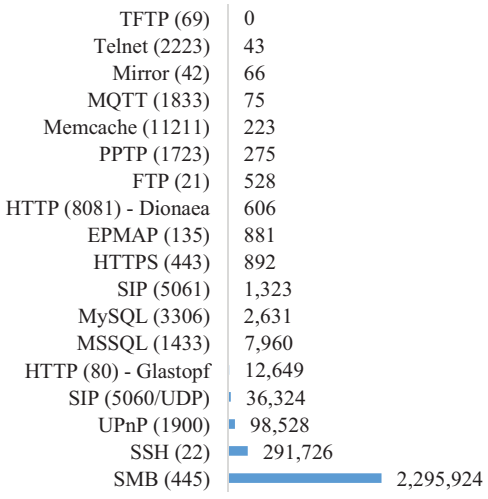

Fig. 3: Service (Port) vs Number of Sessions

A. Cowrie Sessions

Cowrie honeypot was able to capture 291,726 sessions to the SSH service, from which 262,434 were successful logins to the system. There were, also, 43 connections to the Telnet protocol.

In Table II the top usernames and passwords used can be found. One of the combinations that gave access to the system was *root/admin* and so it is justified that they appear as the top ones. As far as the usernames are concerned, we can deduct that attackers are targeting administrative accounts by trying *root* and *admin* and default configurations of vulnerable devices such as Raspberry Pis. On the password side, we can observe that the majority of the tried tokens consists of weak credentials, so attackers are trying to gain access utilizing automated dictionary-based attacks.

TABLE II. TOP 5 USERNAMES, PASSWORDS AND OCCURENCES

Username	Count
root	263,534
admin	5,912
test	869
(empty)	574
Pi	393

Password	Count
admin	258,667
123456	2,507
111	1,006
0	996
password	762

As it can be seen in Table III, most attackers, after compromising the system, are trying to gather information about its architecture and configuration, as well as, about the state of the running processes. This indicates that the attacks are not targeted, but are aiming to compromise a bulk number of hosts and then gather information about their state, capabilities and topology. This deduction is also supported by the malware samples as discussed below.

In our connection logs, we have also detected approximately 1,500,000 direct TCP requests to multiple destinations. We have found that 500,000 were requests to one of the most popular websites in Russia offering Internet Services, ya.ru, either to port 80 or port 443. For the majority of the connections bearing a payload, the decoding of it was not possible as it was routed to port 443 and so it was encrypted SSL data.

TABLE III: TOP 5 COMMAND INPUT

Input	Count
uname -a	748
cat /proc/cpuinfo	742
free -m	732
uname	732
ps -x	731

B. Dionaea Sessions

Dionaea was presented as a Windows host. The larger number of connections came to protocols such as SMB and UPnP for the purpose of downloading malware. Furthermore, there were some connections that tried to connect and initiate sessions using the Session Initiation Protocol (SIP). These connections present an opportunity for further research as SIP is one of the leading protocols used for VoIP.

C. Glastopf Sessions

Lastly, the connections to our web application honeypot mainly targeted default locations of administrative dashboards of popular services such as phpMyAdmin and MySQL. With high probability, the probes come from a preconfigured dictionary used by many attackers. Additionally, there were logged 102 remote file inclusion attempts and a set of requests trying to use our server as a proxy machine.

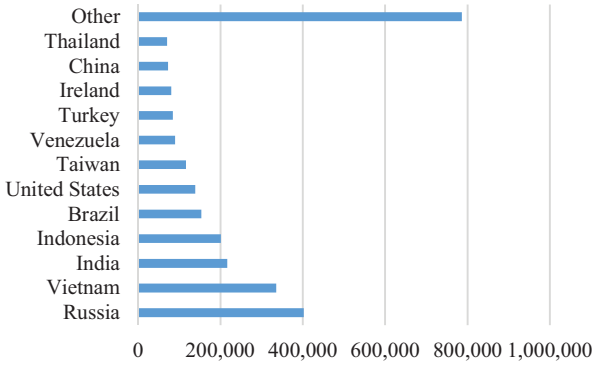

Fig. 4: Origin Country of Attacks

D. Malware Samples

On the malware collected by Cowrie, we were able to capture 129 unique samples. Of these samples, VirusTotal was able to identify all the executables successfully as malicious. A key observation to our research was that three of the shell scripts

captured, tried to download executables from some external asset. The uniqueness of these scripts is that each one tried to download and execute the same file for a wealth of architectures such as *mips, arm4, arm5, arm6, arm7, ppc, x86, i686* and others. This fact confirms our initial notion about bulk attacks trying to compromise any possible system. Also, it is highlighted that IoT devices regardless of their architecture are being actively targeted in order to be exploited and used for malicious purposes, possibly as part of a botnet. Dionaea's malware collecting mechanism was run for the first two of the three months that the honeypot was in place. In this period, it collected 9,199 unique samples that were sent to VirusTotal and the results can be found in Table IV. It is indicative of the motives of the attackers that 9,082 of the 9,199 samples tried to exploit CVE-2017-014, read from process's memory, and install remotely obtained ransomware such as WannaCry.

IV. Related Work

The following works regarding honeypots and virtualization are related to the presented work. The work done by DTAG in using Docker and the Elastic Stack for their honeypot deployment and data analysis, respectively, laid the foundation of our work [12]. Our contribution is the integration of VirusTotal for the automated analysis of all malware samples. In this way, the data are enriched and unique samples are being distinguished. Also, the introduced usage of the lightweight Beats for shipping the logs enables for deployment to low specification devices. In [13], the usage of honeypots for the analysis of malicious activities and connections, as well as, the visualization of the results is presented. Our differentiation is a unified single dashboard that collectively visualizes all honeypot data. A high interaction, Docker-ready, Windows honeypot aimed to be deployed in an enterprise environment is suggested in [14]. The authors are aiming at a transparent integration in a production environment. This requirement is also approached by our work. Lastly, a telnet-based IoT honeypot targeted specifically to collect and analyze attacks and malware families for different CPU architectures is created in [15]. Their approach and results highlight the demand for the deployment of diverse systems.

V. Conclusions And Future Work

Honeypots, although they are not considered a cutting-edge technology, are seeming to add great value to the amount of information an organization is able to gather. During our experiment, the honeypots were successful in capturing a large number of connections and many indicative malware samples. It is evident that cybercriminals are shifting their attention to a broader attack surface, that of the IoT, but in the meantime continuing to threaten the existing infrastructure. Taking into consideration our initial goal for the project, that of a real-time warning asset, we were successful in implementing a fast multi-component system that provides researchers with a wealth of data. This system could be potentially used for uncovering zero-day exploits as well as for creating rules for intrusion prevention systems and IP blacklisting. The usage of Docker made the system resilient to software failures and as a result, it could be characterized as highly available. Furthermore, Docker provides support for different architectures and potentially simultaneous automated deployment to multiple IoT devices in an organization's environment could be considered. As future work, it would be interesting to deploy another honeypot in a different part of the world and correlate the observed data. Also, support for other IoT protocols such as XMPP could be introduced and the system could be configured to attract more specific and targeted attacks. In closing, alternatives to Docker could be examined both for performance and security issues.

TABLE IV: CATEGORIZATION OF SAMPLES COLLECTED BY DIONAEA

Type	Count
PE32 executable for MS Windows (DLL) (GUI) Intel 80386 32-bit	9,100
PE32 executable for MS Windows (GUI) Intel 80386 32-bit	56
HTML document text	9
data	9
MS-DOS executable, MZ for MS-DOS	2

VI. References

[1] "Global Sensors in Internet of Things (IoT) Devices Market, Analysis & Forecast: 2016 to 2022," February 2017. [Online]. Available: https://www.researchandmarkets.com/research/bvgxvl/global_sensors_in.

[2] J. Milosevic, N. Sklavos and K. Koutsikou, "Malware in IoT Software and Hardware," in *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE'16)*, Barcelona, 2016.

[3] N. Sklavos and I. D. Zaharakis, "Cryptography and Security in Internet of Things (IoTs): Models, Schemes, and Implementations," in *8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Larnaca, 2016.

[4] C. Kolias, G. Kamburakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer,* vol. 5, no. 7, pp. 80-84, 2017.

[5] L. Spitzner, Honeypots: Tracking Hackers, Addison-Wesley Professional, 2002.

[6] B. Schneier , Secrets and Lies: Digital Security in a Networked World, Wiley, 2000.

[7] A. Mouat, Using Docker, O'Reilly Media, Inc., 2016.

[8] M. Oosterhof, "Cowrie Honeypot," [Online]. Available: https://github.com/micheloosterhof/cowrie.

[9] "Dionaea," [Online]. Available: https://github.com/DinoTools/dionaea.

[10] "Glastopf," [Online]. Available: https://github.com/mushorg/glastopf.

[11] "VirusTotal," [Online]. Available: https://www.virustotal.com/.

[12] "DTAG Community Honeypot Project (T-Pot)," [Online]. Available: http://dtag-dev-sec.github.io/.

[13] I. Koniaris , G. Papadimitriou, P. Nicopolitidis and M. Obaidat, "Honeypots Deployment for the Analysis and Visualization of Malware Activity and Malicious Connections," *IEEE ICC 2014 - Communications, Software, Services and Multimedia Applications Symposium,* pp. 1819-1824, 2014.

[14] M. Valicek, G. Schramm, M. Pirker and S. Schrittwieser, "Creation and Integration of Remote High Interaction Honeypots," in *International Conference on Software Security and Assurance (ICSSA)*, Altoona, PA, 2017.

[15] Y. Minn Pa pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama and C. Rossow, "IoTPOT: Analysing the Rise of IoT Compromises," in *9th USENIX Workshop on Offensive Technologies*, Washington, 2018.