

ENRICHING HONEYPOT DATA USING CYBER THREAT INTELLIGENCE

PREPARED BY CAITLIN M. ALLEN & ADAM CUNNINGHAM

Presented to the faculty of the Information Technology & Sciences Academic Division at Champlain College for the Bachelor's of Science, Computer Networking & Cybersecurity and Bachelor's of Science, Computer & Digital Forensics

Under the Supervision of Dr. Ali Hadi and Dr. Elizabeth Allen-Pennebaker

02

TABLE OF CONTENTS

Abstract	04
Dedication	05
Acknowledgment	06
List of Figures	07
Abbreviations	09
Chapter 1	10
<i>Introduction</i>	10
1.1 Motivation	11
1.2 Problem Statement	11
1.3 Objectives of This Work	12
1.4 Contribution	12
1.5 Research Limitations	12
1.6 Research Implications	13
1.6.1 Health and Safety Constraints and Implications	13
1.6.2 Environmental Constraints and Implications	14
1.6.3 Social, Cultural, and Political Constraints and Implications	14
1.6.4 Legal Constraints and Implications	15
1.6.5 Economic Constraints and Implications	17
1.6.6 Ethical Constraints and Implications	18
1.7 Organization of the Thesis	19
Chapter 2	20
<i>Background and Related Work</i>	20
2.1 Background	20
2.1.1 Cyber Threat Intelligence	21
2.1.2 Threat Intelligence Lifecycle	22
2.1.3 Containers	23
2.2 Related Work	25
2.2.1 Threat Intelligence	26
2.2.2 Containers	26
2.2.3 Honeypots	27
	31

03 TABLE OF CONTENTS

Chapter 3	34
<i>Methodology</i>	34
3.1 Overview	34
3.1.1 Set up network & networking services	36
3.1.2 Data generation	36
3.1.3 CTI Automated Enrichment	36
3.1.4 Implement data & review/revise	37
Chapter 4	38
<i>Project Experiments and Observations</i>	38
4.1 Overview	38
4.2 Network Overview	38
4.2.1 Kali Linux Machine	39
4.2.2 Firewall	40
4.2.3 Ubuntu Linux Machine #1	40
4.2.4 Ubuntu Linux Machine #2	40
4.2.5 Windows WorkStation	41
4.2.6 Windows Server	41
4.3 Splunk setup and configuration	41
4.4 Honeypots	44
4.4.1 Cowrie	44
4.4.2 LaBrea	45
4.4.3 Artillery	46
4.5 Attack Simulations	47
4.6 Splunk Services and Monitoring	50
Chapter 5	55
<i>Results and Evaluations</i>	55
5.1 Dashboard Results	55
5.2 Threat Hunting Findings	59
5.3 Honeypot Findings	61
Chapter 6	63
<i>Conclusion and Future Work</i>	63
6.1 Conclusion	63
6.2 Recommendations and Future Work	63
References	64
	68

04**ABSTRACT**

Cybersecurity is a rapidly growing field that becomes more complex as time goes on. There are numerous aspects of security that branch out into their own equally complex fields. Many companies and organizations struggle to properly prepare for attacks against them, and fail to utilize threat intelligence or offensive security measures to mitigate these attacks. Many experts struggle to properly digest the information that can be provided by threat intelligence.

This project aims to take data gathered by honeypots to enrich reports that can be provided to cybersecurity experts to improve their security posture. While honeypots and threat intelligence are properly established in the field and have copious research behind their workings and capabilities, the knowledge around applying them to a readable format is limited.

This research aims to bridge that gap between threat intelligence and security hardening. The project will be accomplished by creating a virtual network that emulates an enterprise network. Offensive security mechanisms will be installed on these machines in the appropriate sections to produce the results needed for enriching reports.

05**DEDICATIONS***Adam's Dedications*

Thank you to all of my friends in the capstone Discord who supported me and helped me through the most unique year of my life. I hope we can all meet again one day soon. Another thank you to all my friends online who listened to me rant about these topics that they don't understand, it means a lot to me.

Thank you to my family for supporting me in this dream. You guys were always there and supported what I wanted to do in life.

And finally thank you to my girlfriend Erin Wang who has supported me throughout college and life. You were the main recipient of all my rants and struggles and always knew how to make me feel better. I hope I was able to do the same for you.

Caitlin's Dedications

Thank you to my boyfriend, Jordan Kimball. You have been incredibly supportive in my cybersecurity and forensics career, which has helped me make it this far in the industry. You have also been incredibly helpful when I needed help and had run out of patience (or did something last minute). Having somebody to do cyber stuff with me has been an absolute blast.

Thank you to my family for being supportive of my endeavors, especially with this school year being incredibly rough online and isolating. These four years have been my most formative and I would not have become the person I am without your support. Every phone call to mom, dad, nana, or just being able to spend time at home I appreciate.

06

ACKNOWLEDGEMENTS

Adam's Acknowledgements

Thank you to T.J. Dorchavyk for the encouragement and wonderful conversations at Travelers, this project was inspired from our many conversations and I am grateful for the feedback.

Thank you to Ali Hadi for supporting our process and always meeting with us when necessary, this project truly reached its potential with your input and feedback.

Caitlin's Acknowledgments

I would like to thank Davis McCarthy and Stephen Lincoln of NuHarbor Security. Throughout my internship, working with the Cyber Threat Analysis Cell (CTAC), I learned that I loved threat intelligence and wanted to one day be a threat analyst. I am excited to continue to work with you both and aid the CTAC team.

I would also like to thank Sebastian Szykier of NuHarbor Security. My internship on the Managed Services team allowed me to hone my skills and learn more about the engineering side of security that would become important in the development of this capstone. The entire Managed Services team has been great to work with and learn from, even though it has been virtual. I am excited to start my career as a threat analyst with such a wonderful team and company.

Thank you to our professor, Ali Hadi, you have been incredibly supportive throughout my cybersecurity career at Champlain. You have been amongst the most supportive professors I have had and every class I have taken with you has made me become so much more skilled.

07

LIST OF FIGURES

Figure 3.1: Project Methodology Flowchart

Figure 4.2: List of Virtual Machines deployed on the enterprise network

Figure 4.3: Time zone set to Eastern Time in the Splunk Interface

Figure 4.4: props.conf file in Splunk logging server, with appropriate parameters set

Figure 4.5: SSH port redirect set through iptables

Figure 4.6: LaBrea initialization command, with flags set for safe operating logging to syslog

Figure 4.7: NMAP scan showing “open” ports added by Artillery

Figure 4.8: SYN flood being run against the fileserver through Metasploit

Figure 4.9: SSH brute force being run against the fileserver through Metasploit

Figure 4.10: Artillery logging connections to “open” ports through the honeypot

Figure 4.11: Scope provided to Brandon Wilbur ‘21 to simulate attacks

Figure 4.12: Alert Manager alert for Audit Log tampering

Figure 4.13: Alert Manager alert for Brute Force attacks

Figure 4.14: Sakura Dashboard in Splunk

Figure 4.15: “Countries of Interest” dashboard pane search

Figure 4.16: “Windows Audit Log Tampering” dashboard pane search

Figure 4.17: “Attacking IPs” dashboard pane search

Figure 4.18: “Port Scanning” dashboard pane search

Figure 4.19: “Remote Powershell” dashboard pane search

Figure 4.20: “Linux Shutdowns” dashboard pane search

Figure 4.21: “Suspicious Network Connections” dashboard pane search

Figure 4.22: “Path Traversal” dashboard panel search

08

LIST OF FIGURES

- Figure 5.1: Attacks by country through the Countries of Interest Panel
- Figure 5.2: Number of attacking IPs in the month of April
- Figure 5.3 Windows Audit Log cleared
- Figure 5.4: Port scanning originating from the Kali Linux machine
- Figure 5.5: Shutdown totals on Linux machines
- Figure 5.6: Time range set from the Start of April 17 to the end of April 19
- Figure 5.7: Nmap and Zenmap scans found
- Figure 5.8: Metasploit initialized through msfconsole and msfdb init
- Figure 5.9: Authentication failure logged
- Figure 5.10: Privileged logons to DC01

ABBREVIATIONS

ABBREVIATION

MEANING

CTI

Cyber Threat Intelligence

IOCs

Indicators of Compromise

DTK

Deception Toolkit

TTP

procedures

Tools, tactics, and

SIEM

Security Information and Event Management

DTSPCD

CDE Subprocess Control Service

VM

Virtual Machine

IDS

Intrusion Detection System

ASN

Autonomous System Number

10

CHAPTER 1

Introduction

As adversaries become more sophisticated, the importance of using honeypots for getting to know the threat landscape and what weaknesses exist in an environment is growing. Security company Rapid7 explains that honeypots are decoy systems or environments that entice adversaries to attack them. Honeypots are often deployed for research purposes or as production systems to emulate the real issues in an environment and see how adversaries go about attacking them. There are now many different honeypot configurations available for research and production purposes that can provide insight for an organization about their security posture (“What are honeypots?” n.d). With the use of Cyber Threat Intelligence (CTI), the findings from a honeypot can be enriched to answer the questions of; who is targeting the organization, where are their weaknesses, where should resources for protection be allocated to, and how are threats evolving? The answers to these questions, according to CERT Cyber Threat Intelligence officials, allow for security professionals in an organization to evolve how they hunt threats and defend environments, this also means forensic professionals can have more information about an environment and what they may potentially be responding to in an incident (“What is Cyber Threat Intelligence and how is it used?” n.d). By deploying honeypots with a commonly used and continuously improving technology, like containers, the use of CTI can be used to provide a deep and accurate analysis of the threat landscape, how adversaries are evolving to exploit improving technology, and even predict future threats which can be used by security and forensic professionals during their own investigations.

11

1.1 *Motivation*

This project was chosen due to the rise in popularity of Honeypots over the past few years. Now more than ever enterprise security is important, and with so many VPN connections and remote workers it can be hard to detect unusual traffic. The implications of the COVID-19 pandemic will alter the way work is done forever, and virtual connections will be a necessity to every business. The knowledge of how to properly secure and defend a website will safeguard companies from the risk of a breach and/or data loss. Reporting has also become increasingly more significant in the last decade, and is the most critical piece of cybersecurity. CTI can be used to add value to data from a honeypot by enriching the IOCs with context that is needed to make attributions. By studying and analyzing the different kinds of honeypots available, the forensic and cybersecurity communities can more effectively ascertain knowledge and information from honeypots.

1.2 *Problem Statement*

Honeypots are a rapidly evolving technology in the cybersecurity world, and according to Dominguez (2017), “there is currently no authoritative study or comprehensive data of the use of honey technologies in real-world applications.” (2). Honeypots are on the rise in popularity and are no longer a “one size fits all” mechanism. With many different types and use-cases for honeypots, there is a lack of research on how to properly and effectively utilize them.

Reporting is something that is not taken seriously enough in large companies without a technological focus. On the contrary, reporting can be so dense with information that it is hard to digest. Creating effective and manageable reports is vital to continue to improve security within organizations, and to stop attacks when they do happen.

12

1.3 *Objectives of This Work*

The main objectives of this work can be summarized as the following:

- Setup, configuration, and deployment of honeypots on a virtual network
- Monitoring networks for intrusion detection and appropriately responding to these threats
- Collection of data to utilize in a report scheme that can be sent to cybersecurity experts to mitigate future threats

1.4 *Contribution*

This research will contribute to cybersecurity experts and forensic analysts, who can utilize the results to improve the detection, mitigation, and eradication of cybersecurity threats to a system and/or network. These results will allow experts to take a more proactive approach to threat hunting and enable experts to enact fitting security controls.

1.5 *Research Limitations*

In this research, the scope is limited to virtual environments, which will involve emulating an enterprise setup through the use of Virtual Machines (VMs). The scope of this research includes testing attacks on containers. Containers are a growing technology that are relevant to this research because of recent vulnerabilities and progressions in container technology, like rootless containers, during the spring and summer of 2020. This allows for a more narrow scope that is still relevant to current events in the security and technology industry. Finally, the logging capabilities of this project will be confined to Splunk, a professional data logging platform that provides visualizations and reports on many types of log data. Splunk is an excellent tool for visualizing log data and will create a professional visualization that can be utilized in cybersecurity reports.

13

1.6 *Research Implications*

1.6.1 *Health and Safety Constraints and Implications*

In a constantly changing world with a high reliance on technology, people's lives and well being can be at the mercy of technology. Hospitals are exemplary of this, with patients' medical data, records, and medication prescriptions stored on computer systems. A popular attack on hospitals is ransomware, a type of malware that encrypts all data on a computer, making it inaccessible. Typically, systems hit by ransomware can only be recovered if a large sum of money is paid to the attacker, usually using online cryptocurrency as a payment method. However, if a hospital is hit by a cyber attack, there is more at stake than just company profits. When critical patient data, such as radiation treatment or medicine dosages, is locked and inaccessible because of ransomware, patient's health is adversely affected, as they can miss treatment cycles, dosages, or even receive wrong blood types in a transfusion (Collier, 2020). In recent news, a hospital in Germany experienced a patient death after a ransomware attack led to an ambulance being re-routed. Hospitals in the United States' cyberattacks resulted in surgeries being postponed and hospitals doing everything on paper, delaying care (Simpson, 2020).

This project has a central focus on prevention of threats targeted at critical infrastructure. By improving threat intelligence and reporting on these attacks, organizations, including health care companies, can be more prepared for these types of attacks, and in some cases stop them completely. Although it is impossible to have a perfect success rate of stopping cyber attacks, deploying defensive security measures as a backup protocol can drastically slow threat actors down. When people's health is at risk, any time gained due to preparedness and proper intelligence is immeasurably valuable.

14

1.6.2 *Environmental Constraints and Implications*

We have identified no environmental constraints or implications for this project.

1.6.3 *Social, Cultural, and Political Constrains and Implications*

Geopolitical issues impact the landscape of cybersecurity, which can cause biases in how findings are presented and what conclusions are drawn regarding attribution. In a journal article titled “Cyber Threat Intelligence: A Product Without Process?”, Kris Oosthoek and Christian Doer address the issues of threat intelligence. Biases in presentation of findings and threat attributions stem from not only what typically generates noise on an organization’s endpoints, but also the typical perspective of most commercial cybersecurity companies, which focus on state-sponsored adversaries. The mystique that is created around advanced persistent threats (APTs) has created a tendency to attribute threats to state-sponsored adversaries rather than petty cyber criminals. Everyone wants to blame the “big baddies” of Russia and Iran for their cyberattacks, but in reality it is often petty crime from within the United States.

Indeed, attributions of commercial threat reports, which make up the majority of attributions, range widely depending on the country of origin. For example, US-based CrowdStrike will have no issue reporting on Russian adversaries, but is silent on their own American threat groups, while Russia-based Kaspersky is vice-versa (Oosethek & Doerr, 14 July 2020). It is also important to remember that APT groups aren’t attacking as much as threat analysts think. While many individuals are quick to blame other countries, oftentimes cyber attacks are a result of petty crime carried out by organizations or individuals within the country.

15

1.6.4 *Legal Constraints and Implications*

Honeypots do not host or perform any activity that is considered illegal (Racliffe, 2007). However, they do raise the legal question of entrapment. Under US Federal law, entrapment is defined as “the act of law enforcement officers or government agents inducing or encouraging a person to commit a crime when the potential criminal expresses a desire not to go ahead” (Radcliffe, 2007, p. 14). Honeypots are often cited as a potential source of entrapment, as the primary purpose of honeypots is to lure malicious actors into hacking the honeypot. However, as there is no direct communication between a law enforcement officer and the actor, it is extremely difficult to prove the defense of entrapment, as one of the elements of entrapment is officers or agents “inducing or encouraging a person” to commit a crime. In the case of this particular project, there is no risk of entrapment because the goal is to collect data and perform research, not to catch criminals.

However, consent of the user and/or consumer is another potential challenge of using honeypots that needs to be properly established. Oftentimes law enforcement websites will feature a banner, which is a page that clearly states what the user will consent to by accessing the website, and making sure they click to agree. Making sure that the user agrees to the terms of use (as opposed to placing them in small print at the bottom of a webpage) is critical to confirming consent. Consent is vital to the process because any evidence collected without a user’s consent will be invalid in a court of law. This is an essential part of the honeypot to ensure that all legal standards are met and that all evidence collected remains valid.

16

One example of obtaining a user's consent can be seen on the Department of Defense's website, which displays the following warning is displayed on visitors' device screens:

You have reached the Department of Defense (DoD) computer system!

This is a Department of Defense computer system. This computer system, including all related equipment, networks, and network devices (specifically including Internet access), are provided only for authorized U.S. Government use. DoD computer systems may be monitored for all lawful purposes, including to ensure their use is authorized, for management of the system, to facilitate protection against unauthorized access, and to verify security procedures, survivability and operational security. Monitoring includes active attacks by authorized DoD entities to test or verify the security of this system. During monitoring, information may be examined, recorded, copied and used for authorized purposes. All information, including personal information, placed on or sent over this system may be monitored.

Use of this DoD computer system, authorized or unauthorized, constitutes consent to monitoring of the system. Unauthorized use may subject you to criminal prosecution. Evidence of unauthorized use collected during monitoring may be used for administrative, criminal, or adverse action. Use of this system constitutes consent to monitoring for these purposes.

17

For our honeypot, we will use a message similar to the last paragraph of the above:

Use of this computer system, authorized or unauthorized, constitutes consent to monitoring of the system. Evidence of use collected during monitoring may be used for administrative action. Use of this system constitutes consent to monitoring for these purposes.

It is important to consider the context of the time and location this paper was written, which is in November of 2020 in the United States. Laws are always subject to change, and all cases cited above are U.S. federal law. International law, state laws, and other countries' federal laws may differ from what has been discussed above. While laws relating to technology often lag behind developments in the field, legislation relating to honeypots is starting to catch up. It is always important to make sure that any research or use-cases done with honeypots are in compliance with all legal requirements.

1.6.5 *Economic Constraints and Implications*

While most code this project will be using is open-source or free to use, its infrastructure, as well as the Security Information and Event Management solution (SIEM), requires a license ("What is a SIEM? A Complete Beginner's Guide," 2020). This license price depends on the amount of data ingested, which can add up to thousands of dollars.

Professor Ali Hadi has kindly offered to let us use his infrastructure on-campus as well as his Splunk license, which will allow for this project to continue without out-of-pocket costs.

18

1.6.6 *Ethical Constraints and Implications*

The topic of honeypots often raises a question surrounding morality, as honeypots are a potent example of toeing the line between being ethical and unethical. While the legality of honeypots is mostly undisputable, there are concerns about the way honeypots can entice attackers. The main concern is whether it is ethical to try to trick or lure someone into committing a cybercrime by enticing them with a honeypot (NCSU, n.a n.d). While an argument can be made that these threat actors were going to commit the crime anyway, some honeypots might be said to sweeten the pot a bit too much (NCSU, n.a n.d).

Honeypots and threat intelligence are also controversial because they might “help” hackers become better at what they do (NCSU, n.a n.d). Attackers can become more aware of honeypots that are too obvious or evident, such as those that are clearly and blatantly trying to trick the user into doing something malicious. Many hackers are more cautious due to their increased awareness of honeypot usage, and they could use resources such as this project to increase their own intelligence.

While the target audience is cybersecurity and forensic experts, it would be foolish to ignore the real possibility that a malicious individual could use the findings of this research to help mask their actions and intentions more carefully. Threat actors and hackers will not ignore something just because it is not intended for them, and it has to be kept in mind that this research will be public information. Therefore, the results and improved reporting that is found will be presented as a template to decrease incoming cyber attacks. Presenting information as a template rather than a data analysis will stop hackers from reading numerical reports on which types of their attacks are being detected will help hackers substantially less.

19

1.7 *Organization of the Thesis*

This thesis is divided into six chapters. Chapter 1 covers the basic information and introduction to the work that is proposed. Chapter 2 covers background information about honeypots and threat intelligence, and also includes related work to these subjects. Chapter 3 covers the methodology and the description of the project. Chapter 4 will describe all experiments conducted and all results observed from these experiments. Chapter 5 will cover all results obtained from the experiments detailed in chapter 4. Chapter 6 will conclude the research, summarize the results, and provide an explanation of future work that can be performed.

CHAPTER 2

Background and Related Work

This section will provide background information for the research paper. This will include background information about different topics covered by the project, related research, and the scope of the work. This will expand more on subjects mentioned in the previous section.

2.1 Background

Today, honeypots range quite drastically in variety. Some honeypots are full tool kits that test for different types of attacks and trick threat actors in many ways, whereas others are simple honeypots that simply track for one thing (Kaspersky, 2020). There are also honeynets and honeytokens, which simulate vulnerable networks and databases, respectively.

The development and research of honeypots involves a rich history. Although the first honeypot was only released in 1997, there were several public works in the early 1990s that documented concepts of honeypots, including Clifford Stoll's "The Cuckoo's Egg" and Bill Cheswick's "An Evening With Berferd" (Spitnzer, 2003).

21

Honeypots have been a defensive security mechanism for a couple of decades. In 1997 the first publicly available honeypot created was Fred Cohen's Deception Toolkit. The Toolkit was “intended to make it appear to attackers as if the system running DTK [had] a large number of widely known vulnerabilities” (Peter and Schiller, 2008). This very first honeypot was described as a:

“collection of PERL scripts and C code that is compiled and installed on a Unix system.” It “emulates a variety of known Unix vulnerabilities. When attacked, these emulated vulnerabilities log the attacker’s behavior and actions and reveal information about the attacker” (Spitnzer, 2003).

This was the beginning of honeypot solutions for cybersecurity, but only simulated attacks were tested and they were never used to protect anything real. That would soon change.

In 2002, honeypots captured something more than just a pre-scripted test, they caught a real unknown threat in the wild. This attack involved a known vulnerable service within the CDE Subprocess Control Service (dtspcd), but security experts did not know how this could be exploited (Spitnzer, 2003). Using a honeypot intentionally running a vulnerable version of dtspcd, the threat intelligence community was able to capture a remote attack, locating and patching the code that allowed for the exploit.

22

Since then, hundreds of honeypot solutions have been developed by the threat intelligence community, some falling into a more broad range of attack prevention, and others being geared towards specific attacks and vulnerabilities. Honeypots also extend outside of vulnerable servers, and have been used to emulate suspicious websites in order to catch traffickers and black market dealers. The use of these traps are extremely beneficial in protecting many different fields, companies, and people.

2.1.1 *Cyber Threat Intelligence*

CTI is the knowledge that allows an organization to prevent or mitigate cyber attacks. Cybersecurity and consulting firm, Mandiant, describes this as evidence-based knowledge about adversaries and what their motives, intents, and capabilities are (“Cyber Threat Intelligence 1010”, n.d). Mandiant specifically emphasizes adversaries, with other companies like Kaspersky focusing more on the data analysis (“Threat Intelligence Definition“, 2019, October 02). CTI plays a fundamental part of an organization by enriching and giving context to the raw data received. This data becomes intelligence when it is processed and context is added. This can provide a high-level outlook of what is going on in an environment, which is incredibly useful when using a honeypot to track activity and use this information to better security posture. It goes through something known as the threat intelligence lifecycle to become a finished product.

23

2.1.2 *Threat Intelligence Lifecycle*

The threat intelligence lifecycle is the process of collecting data, processing data, and creating a finished product that can be used for analysis. This life cycle enables security analysts to make informed decisions about activity in the organization as well as address any potential attack vectors in the environment.

1. The threat intelligence lifecycle begins with planning and direction. This is about asking the right questions and creating actionable focus on a specific event, fact, or activity. Any broad or vague questions should be avoided (“Everything You Need To Know”, n.d).

2. The next step is data collection, this involves gathering raw data that fulfills the requirements of the first step. Collecting data from a wide range of sources like network and host event logs, external intelligence reports, and past incident history gives context to the environment and what threats exist. While IOCs like IP addresses, domains, and file hashes are important, data like TTPs, vulnerability disclosures, and even PII can be used in the context of threat intelligence (“Everything You Need To Know”, n.d).

24

3. The third step is processing the data, this means that the data needs to be organized and sorted, using tagging and filtering to only see relevant information. Even small organizations bring in millions of logs a day in their network. Most of this data is not relevant in an intelligence context. A SIEM solution is a good start for filtering and sorting data, solutions like Splunk or Kibana make it easy to structure, correlate, and visualize data that is unstructured. Security engineers work to automate the organization of unstructured data into structured and easier to read through add-ons, app integration like Enterprise Security for Splunk, and creating security rules for what triggers an alert (“Everything You Need To Know”, n.d).

4. After data collection comes analysis in order to make sense of the data. Analysis can involve further investigating suspicious events and looking for patterns that indicate malicious activity. This takes many different forms but the idea is to turn the data into a format that the intended audience will understand. This can range from a list of threats to peer-reviewed reports showing current findings and what this can mean for the organization (“Everything You Need To Know”, n.d).

5. The fifth step is dissemination. This is the distribution of the final product into the right hands, such as clients, analysts, and leadership that work with the security team and risk management team to ensure the safety of an organization. Using a ticketing system or having an automated report that can be sent to those who need the information is one way to achieve this (“Everything You Need To Know”, n.d).

25

6. In step six, the final step, this is where the life cycle becomes full circle. This is related to the planning and direction phase since the feedback after receiving the final product is able to determine if questions are answered and how improvement can be made. This allows for continued improvement of the threat intelligence process (“Everything You Need To Know”, n.d).

2.1.3 Containers

Throughout the spring and summer of 2020, Palo Alto’s Unit 42 Threat Intelligence team wrote many articles about containers and their vulnerabilities and targeted attacks on them. In an article about rootless containers, Aviv Sasson writes about how this is the next big trend in container security. While similar to conventional containers, they do not need root privileges to be created and are in early stages of adaptation. While providing a layer of security by allowing unprivileged users to run containers on the same machine inside of a nested container, vulnerabilities for the technologies used by these containers pose a risk to them (Sasson, 2020). For example, a networking configuration called Slirp is vulnerable to CVE-2020-1983 which is an IP Fragmentation vulnerability, Slirp does not verify the size of IP packets and will crash if it is more than 65,353 bytes which can cause the network stack to become unstable or lost (Sasson, 2020). Daniel Prizmant of Unit 42 writes about how Windows containers are vulnerable to remote code execution in a Kubernetes environment which can be used to spread between nodes, this means a single application running inside a Windows Server Container could be broken out of (Prizmant, 2020). This focus on container security provides a narrow scope that is relevant to security trends seen recently.

26

2.2 Related Work

2.2.1 Threat Intelligence

Kovacs, E. (2015, January 16). False Positive Alerts Cost Organizations \$1.3 Million Per Year: Report. <https://www.securityweek.com/false-positive-alerts-cost-organizations-13-million-year-report>

In Edward Kovac's article for SecurityWeek, Kovacs goes into details about false positive security alerts and a study by the Ponemon Institute. This study found that on average, organizations spend about 21,000 hours each year analyzing false positives which translates to \$1.3 million a year wasted on investigating false positives.

Participants of this study were found to have "ad hoc" approaches to how they go about malware containment, with the same 33% of participants having a structured approach. One in 10 participants said their structured approach relies on manual activities.

In many cases, the CISO was responsible for malware containment, while the other 40% said that there is no sole person or function responsible for containment. Vendors and peer-to-peer intelligence were the main source of intelligence for 66% of participants.

27

Brian Foster, the CTO of Damballa, said organizations need to get a firm grip on their security posture and use the right intelligence to detect active infections to reduce their organization's risk.

With this understanding of false positives, this can be leveraged in the engineering portion of using Splunk to fine-tune in our honeypot environment how we alert and continue to allow us to evolve alert criteria. Having our alert rule syntax written out can help others in cybersecurity understand what they may need to do differently and provide a more granular control over their alerts in something like Enterprise Security.

2.2.2 Containers

Zelivansky, A. (2020, September 10). The Challenge of Persistence in Containers and Serverless. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/persistence-in-containers-and-serverless/>

Palo Alto's Unit 42 Threat Intelligence group in this article discusses how attackers maintain persistence in containers and serverless computing resources. When attackers exploit a vulnerability, it may eventually be patched, which can cause attackers to lose access to the machine they had worked to exploit. Persistence can be achieved through simple methods like editing configurations or by using methods like modifying or installing binaries and libraries.

28

Persistence in containers can be tricky, traditional methods of persistence are only partially effective. When a container is shut down, the persistence gained by modifying cron jobs or binaries is now gone. One way to maintain persistence is by exploiting the applications the architecture is specifically built on.

Serverless persistence is similar to containers, but fewer attack vectors are exposed. Misconfigurations or vulnerabilities of the cloud provider are the only way to truly maintain persistence.

Chen, J. (2020, July 01). Attacker's Tactics and Techniques in Unsecured Docker Daemons Revealed. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/attackers-tactics-and-techniques-in-unsecured-docker-daemons-revealed/>

Palo Alto's Unit 42 Threat Intelligence group between September 2019 and December 2019 periodically scanned metadata from Docker hosts exposed to the Internet. This research has revealed some of the tactics and techniques that attackers are using on compromised Docker engines. Four categories of malicious activity were found:

1. Deploying Container Images with Malicious Code

Malicious images are pushed to a public registry for download. They are then pulled and deployed on unsecured Docker hosts.

2. Deploying Benign Container Images and Download Malicious Payloads at Runtime

Benign images are deployed on Docker hosts but the malicious payloads are downloaded and executed inside these benign containers.

29

3. Deploying Malicious Payloads on Host

Adversaries mount the entire host filesystem to a container and then access the host filesystem from the container.

4. Obtaining Sensitive Information from Docker Log

Adversaries scrape Docker logs to find information like credentials, configurations, etc..

Countermeasures to combat these were also discovered by Unit 42 now that the threat landscape is known, helping secure those in the future who use containers. This is useful for knowing what the current threat landscape looks like and hosting vulnerable containers. We can know what attackers are looking at and how to deploy a honeypot that would be vulnerable to what attackers want to exploit.

Prizmant, D. (2020, July 15). Windows Server Containers Are Open. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/windows-server-containers-vulnerabilities/>

With hosting vulnerable services and containers in a honeypot, the concern of breaking out has come up. In Palo Alto's Unit 42 Threat Intelligence research, the topic of Windows containers and breaking up was discussed. Daniel Prizmant explains how to break out of a container which will be valuable for our understanding later on and securing our environment.

Leveraging remote code execution (RCE) in a Kubernetes environment, this exploit demonstrated in this post shows how this can be used to spread between nodes. This means if a single application running inside a Windows Server Container is breached it could breach the boundaries of the container across applications on the host.

30

Azure Kubernetes Service (AKS) is a managed container orchestration service based on Kubernetes and is available on the Azure Public Cloud. AKS can be used to manage and scale Docker containers and container-based applications across a cluster of container hosts. Every single Kubernetes cluster that has a Windows node is vulnerable to this escape and can continue to the rest of the cluster.

Sasson, A. (2020, May 31). Rootless Containers: The Next Trend in Container Security. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/rootless-containers-the-next-trend-in-container-security/>

Since container security was a hot topic in security this summer, this is a relevant topic that can help with understanding vulnerable services needed to make a successful honeypot. Using a relevant and new tool like rootless containers can help us take a more realistic approach to setting up a honeypot and use a modern and new technology that organizations are adapting. This can also further research on the security of rootless containers and allow for us to compare the differences in the security of traditional vs rootless containers.

Rootless servers are similar to conventional containers, with the difference being they do not need root privileges in order to be formed. These containers are in their early stages of adoption but have been supported by many. They provide a layer of security, allow multiple unprivileged users to run containers on the same machine, and isolate inside nested containers.

Vulnerabilities for services and technologies used by rootless containers exist, a networking configuration called Slirp is vulnerable to CVE-2020-1983 which is an issue with how Slirp does IP fragmentation. If Slirp does not verify the size of a fragmented IP packet, it will crash if it is larger than 65,353 bytes. This causes the network stack to be lost and unstable. This can lead to code execution on the container and eventually a breakout.

31

2.2.3 Honeypots

ACE Team. (2018, December 11). Secure your Network by setting up a Honeypot - Loginsoft - Cybersecurity, Software Development, Offshore Services. Retrieved October 14, 2020, from <https://www.loginsoft.com/blog/2018/11/16/secure-your-network-by-setting-up-a-honeypot/>

This blog post is by loginsoft, which is a company that provides cybersecurity and software development information. This makes them a credible source when talking about cybersecurity topics, which honeypots fall under. This blog post has some really great diagrams that visually showcase how honeypots work, and also provides some deeper details about how they function. This diagram in particular will be a very useful way to help describe honeypots to those who may not be as familiar with them, and is a great way to build our learning on honeypots.

Kaspersky. (2020, September 10). What is a honeypot? Retrieved October 14, 2020, from <https://usa.kaspersky.com/resource-center/threats/what-is-a-honeypot>

This blog post is by Kaspersky, another company well known for innovative cloud security contributions and internet security. Kaspersky also provides its own antivirus software, and they are a well informed company in the world of cybersecurity. This post is very helpful to furthering our research because it goes into the details about specific types of honeypots. While it is good to know a general definition of what they are, having some valuable information on the different types of honey pots will help us test our research.

32

Sanders, C. (2020). *Intrusion detection honeypots: Detection through deception*. Oakwood, GA: Chris Sanders.

This textbook explores the world of honeypots, giving rich details about building, deploying, and monitoring honeypots. It is designed to help cybersecurity experts utilize honeypots as best as they can, and gain an advantage on attackers that find their way into the system. It follows the "See-Think-Do" framework to help produce the best results.

Symanovich, S. (2020, May 26). What is a honeypot? How it can lure cyberattackers. Retrieved October 14, 2020, from <https://us.norton.com/internetsecurity-iot-what-is-a-honeypot.html>

This blog post was published by Norton, a famous security provider for computer. They are best known for their antivirus software, so their sources are reliable in regards to the matter of cybersecurity. This article provides an excellent overview on the uses of honeypot, and how they work on a more sophisticated level. It also provides some recent examples of how honeypots have helped mitigate attacks from threat actors around the world.

33

Wallen, J. (2019, October 02). How to quickly deploy a honeypot with Kali Linux. Retrieved October 14, 2020, from <https://www.techrepublic.com/article/how-to-quickly-deploy-a-honeypot-with-kali-linux/>

This blog post is by techrepublic, a website that has many blog posts by different writers about all sorts of different technological topics. The article was written by Jack Wallen, who is an award winning writer for the website. He is a promoter of open source software, which Linux falls under, so he is well informed about the topics covered by this article. This article shows how to install a honeypot on a Kali Linux Virtual Machine. This type of research is helpful because it will help us with the technical aspects of setting up honeypots. We can follow similar guides like this in order to set up different types of honeypots.

34

CHAPTER 3

Methodology

This chapter will present the design of the project and show the steps to create the environment, as well as how data will be analyzed. This proposed methodology implements existing techniques to utilize honeypot data for improving security posture as well as utilizing this data during forensic investigations.

3.1 Overview

The proposed methodology will utilize threat intelligence information to enrich our findings from honeypots that are focused on containers. Creating digestible reports from information gathered in this environment will allow for this information to be used by security engineers managing production environments to increase their own security posture, security analysts to know what attacks may happen, and forensic investigators to have an idea of what has happened in an environment. The environment created will emulate a small enterprise environment that will explore existing techniques and approaches to understanding data from a honeypot. This information will be used for predicting threat trends, improving forensic investigations, and improving security posture through automated and on-demand reports created in Splunk based on this project's findings.

35

Upon set-up of the honeypot environment with proper network segmentation and services deployed, data generation will begin. Once sufficient data has been collected, intelligence and forensic analysis will be conducted to put this data into reports. These reports will be created on a weekly basis at minimum to keep up with current threat trends in the environment as well as investigate any incidents that occur to improve security in the environment, once data has been ingested into reports, the forensic and intelligence findings will be reviewed and revised on an as-needed basis. The following methodology will be followed to generate the data and investigate the findings (Figure 3.1).

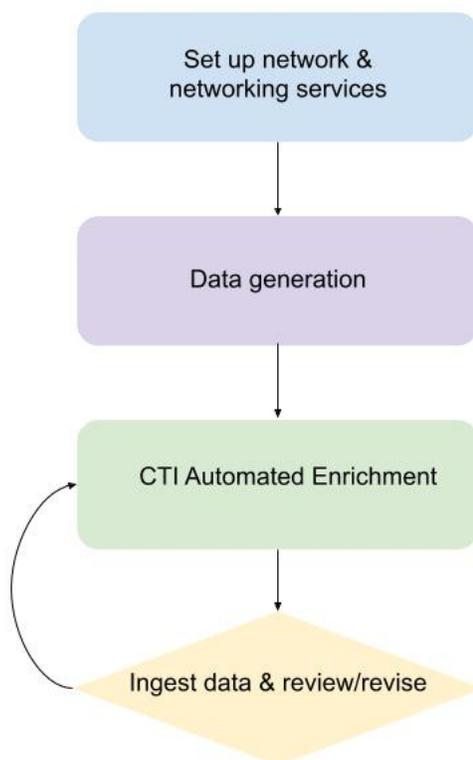


Figure 3.1 Project Methodology Flowchart

36

3.1.1 Set up network & networking services

An enterprise network must be simulated for the findings of this research to be applicable to the security field. The network architecture would simulate a small enterprise environment with a firewall, Intrusion Detection System, SIEM solution, services running in a container such as a Wordpress website, internal servers like a file share, and two or three standard end-user hosts running an operating system like Windows 10 or Ubuntu Linux. These hosts will be segmented to simulate an enterprise environment with a management network that is unable to be reached by the Internet or public facing services as well as internal hosts like the file share.

3.1.2 Data generation

Once the honeypot and network has been set up, the data to be collected will be activity recorded by the firewall and services that will be ingested into a SIEM solution for analysis. This data will be about the attacks attempted or performed on the honeypot, activities performed by different users, network activity like enumeration of systems or services, and commands issued.

3.1.3 CTI Automated Enrichment

When data is generated, services implemented in the SIEM solution can automatically enrich the data. This takes raw data and turns it into what is known as intelligence by processing and sorting the data. The intelligence can then be used to gain insight into what is targeting the honeypot.

37

3.1.4 *Implement data & review/revise*

Once enough intelligence has been gathered, reports can be created to watch for threat trends, see where security posture can be improved, and find where adversaries are attempting to attack. This data will be reviewed for inaccuracies, and the process will be reviewed to correct these errors.

CHAPTER 4

Project Experiments and Observations

This chapter will explain the experiments done during the duration of the project, and the observations made by team members. The processes and explanations for the steps taken during the experiments will be detailed in this section.

4.1 Overview

In this section, the overall architecture of the network is defined, alongside justifications for the operating systems that were chosen for this project. The goal is to simulate a real enterprise network as closely as possible in a virtual environment.

This involves the inclusion of workstations, firewalls, a management server and firewall, a logging server, and more. This section will also cover the honeypots that were installed on the systems, and their purpose and utility within the scheme of the network. Furthermore, this section will cover the simulated attacks that were performed on the network to generate logs.

4.2 Network Overview

Our initial phase of the experimentation process was to obtain the Virtual Machines that were necessary for our project. We reached out to Ali Hadi, who helped host the virtual machines on his private network that he allowed us access to. We decided on hosting this way for ease of accessibility, as both of us felt that with the remote requirements that this was the best solution to cover our needs.

39

For our project, a total of six virtual machines were deployed for experimentation purposes.

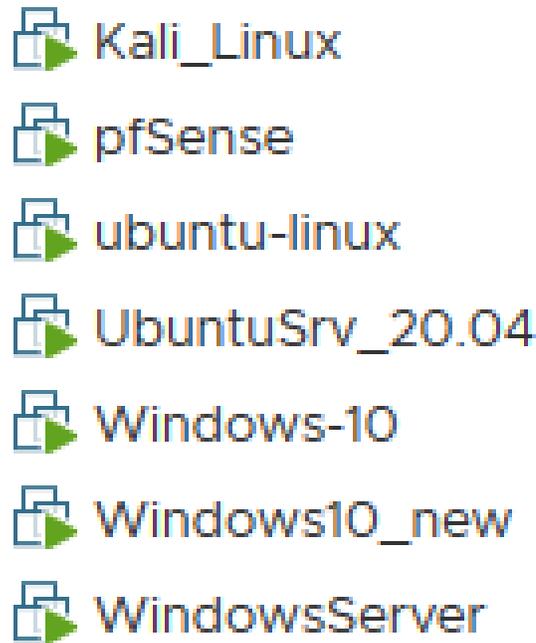


Figure 3.1 List of virtual machines

4.2.1 *Kali Linux Machine*

The first was a 64 bit rolling Kali Linux distribution running version 2020.3. This machine's primary purpose was to simulate attacks on the network using Kali's frameworks such as Nmap and Metasploit. It sits inside the network and has connectivity to all the other machines, representing what would happen if a member of an enterprise network gained access to a computer that could potentially damage other systems.

40

4.2.2 Firewall

The next machine is a 64 bit pfSense machine using the FreeBSD Pre-11 version operating system. This acts as our firewall for the network, filtering the traffic that goes in and out of the network. No traffic is allowed into the network except for established connections coming out from the network. This prevents attackers from easily accessing network resources while being outside the network.

4.2.3 Ubuntu Linux Machine #1

Next is a 64 bit Ubuntu Linux machine running Ubuntu 20.04.2 LTS. This machine serves as the primary file server for the network, and would host company resources that can be accessed by those inside the enterprise network. Because of this, it also is the machine that the majority of the honeypots and offensive security measures are found. This machine contains the most important data to protect, so therefore it is the most locked down with these measures that were installed on the system.

4.2.4 Ubuntu Linux Machine #2

Another 64 bit Ubuntu Linux machine (20.04.2 LTS) is present on our network, with this one being to host our Splunk service. Splunk is a logging service that all of the machines on the network can forward their logs to, so that our logs are standardized and centralized.

41

4.2.5 Windows Workstation

The next machine is a Windows 10 Pro edition running OS build 18363.1379. This acts as the average workstation on a network. Typically there would be multiple of these machines to simulate each person working in an enterprise, but for the sake of simplicity there is one workstation on this network. This machine is used to ensure that the security measures that we have set do not affect the productivity of an average worker.

4.2.6 Windows Server

The last machine is a Windows Server 2019 Standard edition running OS build 17763.1397. This acts as the Domain Controller for the system and also is responsible for accessing the logs in the Splunk server. Because we would only want the logs to be accessed by a select group of individuals, the Windows server is the primary way of connecting to the logging server in order to view them.

4.3 Splunk setup and configuration

In order to collect and centralize the logs on the system, we used Splunk, an enterprise logging tool. We installed Splunk on one of our Ubuntu servers, that in an enterprise environment would be protected by a secondary firewall that would keep the management and system administration machines on it. We installed Splunk 8.1.1 and then worked on installing Splunk forwarders on each of the other machines on our network. The forwarders will allow for each machine to send their logs to Splunk, so that they can all be centralized in one location. We installed 8.1.3 of the forwarders on each machine and configured the Splunk machine to accept these incoming forwards.

42

The Splunk logging configurations needed to be set so that our logs would all show up in the timezone. If this was not configured correctly, then logs would show up 5 hours in the future due to the logs being in UTC instead of EST. To correct this, two actions needed to be taken. First, the Splunk UI needed to have its timezone set to Eastern Time. This is more for visual purposes, so that it is easier to see when our logs are coming in and so we don't have to convert timezones.

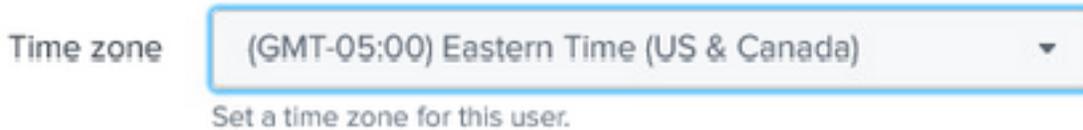


Figure 4.3 Time zone set to Eastern time in the Splunk Interface

The other adjustment that we had to make is to make sure the Splunk server was put into the Eastern time zone within the server itself. This can be set using `timedatectl` on Ubuntu. Furthermore, the `props.conf` file located in the splunk install directory at `/opt/splunk`. Here the `"DETERMINE_TIMESTAMP_DATE_WITH_SYSTEM_TIME"` parameter needs to be set to true, so that timestamps are accurately represented in the Eastern time zone.

43

```
[default]
CHARSET = UTF-8
LINE_BREAKER_LOOKBEHIND = 100
TRUNCATE = 10000
LB_CHUNK_BREAKER_TRUNCATE = 2000000
DATETIME_CONFIG = /etc/datetime.xml
ADD_EXTRA_TIME_FIELDS = True
ANNOTATE_PUNCT = True
HEADER_MODE =
MATCH_LIMIT = 100000
DEPTH_LIMIT = 1000
MAX_DAYS_HENCE=2
MAX_DAYS_AGO=2000
MAX_DIFF_SECS_AGO=3600
MAX_DIFF_SECS_HENCE=604800
MAX_TIMESTAMP_LOOKAHEAD = 128
DETERMINE_TIMESTAMP_DATE_WITH_SYSTEM_TIME = True
SHOULD_LINEMERGE = True
BREAK_ONLY_BEFORE =
BREAK_ONLY_BEFORE_DATE = True
MAX_EVENTS = 256
MUST_BREAK_AFTER =
MUST_NOT_BREAK_AFTER =
MUST_NOT_BREAK_BEFORE =
```

Figure 4.4 props.conf file in Splunk logging server with appropriate parameters

For log standardization, the Common Information Model (CIM) add-on was used to standardize log fields which automatically could tag log components with their use, such as a source IP address versus a destination IP Address.

44

4.4 Honeypots

As mentioned above, one of the Ubuntu servers was dedicated to hosting the honeypots and offensive security mechanisms. This was done because the file share server would be the one in most need of being protected from attacks, and also to have more freedom and control over the honeypots using Linux.

4.4.1 Cowrie

The first honeypot that was installed was cowrie, a honeypot which emulates an SSH terminal. This honeypot by default runs on ports 2222 and 2223. The main appeal to this honeypot is that it can make SSH appear like a vulnerable service, but rather it will track and log all actions that are taken by a user in the fake SSH terminal. This honeypot is extremely customizable, allowing for an entire “fake” filesystem to be created that truly looks like a genuine SSH shell.

This honeypot required that python3 and python3-virtualenv were installed as prerequisites, and git clone was used in order to get the files off of Github and onto the local machine. Once this was set, the default ssh port had to be changed in order to prevent interference with cowrie. The ssh port on the machine was changed to port 22222. Once done, a virtual python environment was created for cowrie to run. The last step that had to be taken was to use iptables to forward traffic coming in to port 22 to ports 2222 and 2223, so that attackers who try to use the default port of 22 to connect to SSH would instead be connected to the Cowrie honeypot.

45

```
ser1@fileserv:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
ser1@fileserv:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223
```

Figure 4.5 SSH port redirect set through iptables

Cowrie tracks a command history of all commands entered by the user in the emulated SSH shell. It also tracks information about their time of connection and where their connection comes from through ip addresses. These are then logged and using the Splunk forwarder, are shipped over to our logging server for further analysis.

4.4.2 LaBrea

The next honeypot installed on the system was LaBrea, a honeypot that takes over unused ip addresses on a network and traps an attacker in these fake ip addresses. These virtual servers make themselves look vulnerable by opening ports that would normally be vulnerable on a system. However, once an attacker connects to these systems, LaBrea “traps” them by getting them stuck in the connecting phase. This can hold them for a substantial period of time.

This honeypot was initialized with parameters for removing the initial warning (-z), run within a safe environment (-s), logging to syslog (-l), logging the bandwidth used by the honeypot (-b), using port 10000 (-p 10000), and using the ens160 interface (-i ens160).

```
user1@fileserv:~/artillery$ sudo labrea -z -s -l -b -p 10000 -i ens160
```

Figure 4.6 LaBrea initialized command, with flags set for safe operating logging to syslog

46

This honeypot was chosen to be a part of our network because it helps obscure the finer details of our network. If there were no fake machines, someone performing a network scan of the network could easily see which ip addresses were being used and which ports were open on all of them. This could easily allow someone to deduce which computer does what based on the ports and services that are opened within the system. This honeypot also allows for threat intelligence workers to respond quickly without risking much damage, as LaBrea actively attempts to sabotage the connection to these fake IP addresses and slow down the attacker from making any more connections. When these logs are sent to the Splunk server, an incident response team could mitigate the threat while they are stuck before they can cause more damage to the system.

4.4.3 Artillery

The final honeypot installed on the system is Artillery by Binary Defense Systems. This is a combination of a honeypot, a monitoring tool, and an alert system. Its main feature is to “open” up common ports that are attacked by hackers, typically opening ports that have exploits associated with them. However, when someone tries to connect through these ports, it will blacklist their IP address from connecting to the system again. Furthermore, it also acts as a monitoring service, allowing for certain directories and files to be monitored. By default, the directories monitored are /var/www/ for potential website hacks/injections, and /etc/ for potential configuration changes. Finally, this system also looks for SSH brute force attempts and notifies the user of when something is happening.

47

```
[*] Nmap: Discovered open port 21/tcp on 192.168.8.110
[*] Nmap: Discovered open port 25/tcp on 192.168.8.110
[*] Nmap: Discovered open port 5900/tcp on 192.168.8.110
[*] Nmap: Discovered open port 8080/tcp on 192.168.8.110
[*] Nmap: Discovered open port 110/tcp on 192.168.8.110
[*] Nmap: Discovered open port 23/tcp on 192.168.8.110
[*] Nmap: Discovered open port 1723/tcp on 192.168.8.110
[*] Nmap: Discovered open port 22/tcp on 192.168.8.110
[*] Nmap: Discovered open port 1433/tcp on 192.168.8.110
[*] Nmap: Discovered open port 44443/tcp on 192.168.8.110
[*] Nmap: Discovered open port 2222/tcp on 192.168.8.110
[*] Nmap: Discovered open port 16993/tcp on 192.168.8.110
[*] Nmap: Discovered open port 5061/tcp on 192.168.8.110
[*] Nmap: Discovered open port 5060/tcp on 192.168.8.110
[*] Nmap: Discovered open port 5800/tcp on 192.168.8.110
[*] Nmap: Discovered open port 10000/tcp on 192.168.8.110
```

Figure 4.8 SYN flood being run against the file server through Metasploit

We installed this honeypot because it helped cover some bases that our other two did not cover. Having “vulnerable” ports open on our file share server may trick a hacker into attempting to attack through one of those ports, which would allow us to quickly shutdown the attacker if they continued to try any more activity. The blacklisting of their IP also serves a defense mechanism that does not require constant surveillance, and can help us defend against attackers.

4.5 Attack Simulations

In order to simulate attacks on a real system, we used Metasploit through our Kali Virtual machine to simulate different attacks. One attack that we performed was a denial of service through metasploit using TCP packets that don't look for responses. This is known as a synflood and can shut down networks accordingly. When performing this attack the idea was to try and shut down the file share server so that it would cause issues.

48

Another attack we performed is an SSH brute force attack through Metasploit, which covers two of our honeypots (Artillery and Cowrie). I managed to trigger both the “real” SSH by forcing my way into port 22222, which would have Artillery create logs about it, as well as going through the “fake” SSH port on 22 which would have Cowrie log my actions.

```
msf5 auxiliary(scanner/ssh/ssh_login) > exploit
[+] 192.168.8.110:22 - Success: 'user2:forensics' ''
[*] Command shell session 2 opened (192.168.8.109:38259 → 192.168.8.110:22)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > █
```

Figure 4.9 SSH brute force being run against the file server with Metasploit

Finally, we performed some basic connections through SSH to the “fake” IP addresses that were established through LaBrea to see how the Kali machine got stuck and how the information was logged down. I also did a similar thing using the “vulnerable” ports that were opened up by Artillery to see the IP get blacklisted and recorded.

```
2021-03-12 06:01:03: Artillery has detected an attack from 192.168.8.109 for a connection on a honey
pot port 5800
2021-03-12 06:01:03: Artillery has detected an attack from 192.168.8.109 for a connection on a honey
pot port 5060
```

Figure 4.10 Artillery logging connections to “open” ports through the honeypot

49

An additional second round of attacks were completed later during the project, utilizing both internal and external resources to help generate further data for building out searches and dashboards.

One service we opted to test against was Nextcloud, which was installed on the fileshare server as the web service for that server. It was able to be connected to using port 8080 on the Windows 10 workstation on our network, and some basic HTTP attacks were performed to generate data. This included attacks such as directory traversal, Cross Site Scripting (XSS), and command injection. While these attacks were not successful due to the security set on the nextcloud service, the attempts would still show up within the logs, and these would be important to monitor and ensure that correct security measures are in place for.

Additionally, Brandon Wilbur, a fourth-year Computer Networking & Cybersecurity student, provided attack simulations for us as an outside resource. He was selected due to his extensive knowledge with penetration testing and attack simulation, and also being unaware of the internal workings of our network. He was given a set of instructions as outlined below:

Scope

Box to use:

In this environment, you will be able to use the Kali Linux host named "Kali Linux" in vSphere under the Projects folder. Please do not use any VM in the FOR340 folder.

When attacking the environment, you will be allowed to access the Windows Server known as "DC01" with an address of 192.168.8.112 and the Ubuntu Linux host known as "fileserver" with an address of 192.168.8.110.

Off limit boxes:

Under no circumstances will you attempt to access host 192.168.8.254, also known as "logger" or "UbuntuSrv_20.04". This host is the Splunk server needed for data collection.

Do not manipulate the firewall in anyway that may cause networking or routing issues.

Rules:

- Do not break anything to the point of being useless.
- Do not manipulate any data being sent to Splunk for the sake of the project integrity.

Credentials to login to forensicslab.champlain.edu

Username: caitin.allen

Password: ██████████

Figure 4.11 Scope provided to Brandon Wilbur to simulate attacks

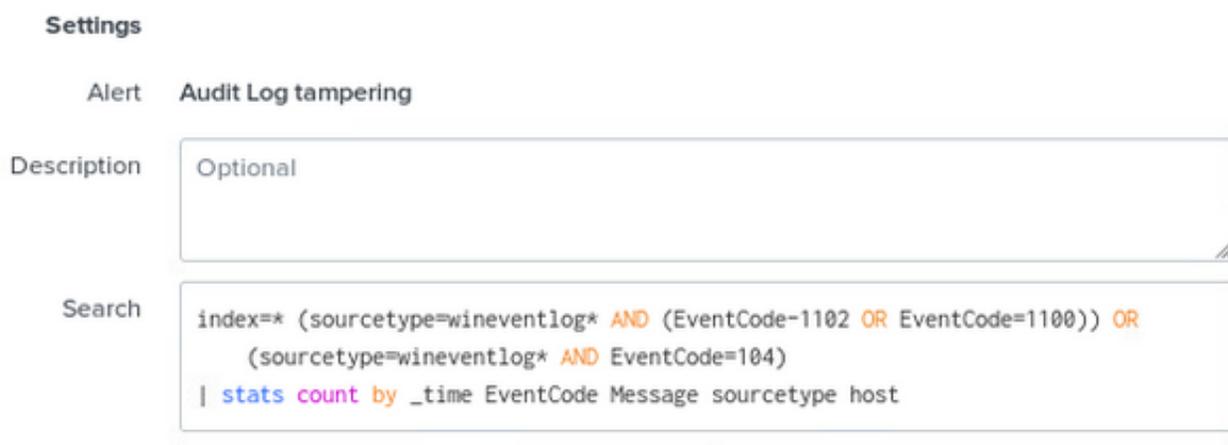
50

Within these instructions, Brandon performed numerous unknown attacks on the system. It was specifically mentioned in-person to not say which attacks he did so that we could simulate a real outside threat, and use our logging and threat intelligence services to help pinpoint the attacks that were made within the system. This would emulate a real attack where there is not direct information about the types of attacks that were being made on the system.

4.6 Splunk Services and Monitoring

Once we had some base logs generated through simulated attacks and general network operations, we looked to our Splunk server to begin to install threat intelligence measures to generate our dashboards and reports.

We are using Alert Manager as an alternative to Enterprise Security due to pricing. Alert Manager is configured at this time to alert in real-time on brute force attacks and Windows Event audit log tampering using the following searches.



The screenshot shows the configuration for an alert named "Audit Log tampering". It includes a description field and a search query field.

Settings

Alert **Audit Log tampering**

Description: Optional

Search: `index=* (sourcetype=wineventlog* AND (EventCode=1102 OR EventCode=1100)) OR (sourcetype=wineventlog* AND EventCode=104) | stats count by _time EventCode Message sourcetype host`

Figure 4.12 Alert Manager alert for Audit Log tampering

51



Figure 4.12 Alert Manager alert for Audit Log tampering

To create an intelligence hub of our findings, a dashboard called “Sakura Dash” was created. Sakura Dash features an overview that can be used to further explore data and show how this data found can help with identifying security threats and weaknesses that may be present in a similar or identical production environment used by an organization.

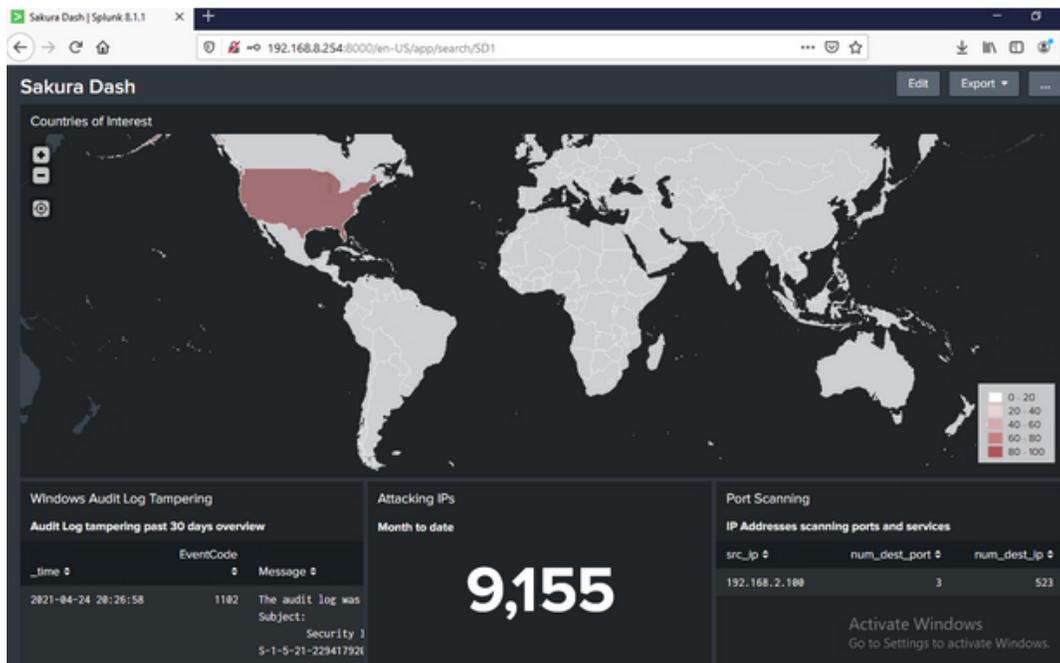


Figure 4.14 Sakura Dashboard in Splunk

52

The following search was used to construct the “Countries of Interest” dashboard pane:

```
index=* sourcetype=* NOT (behavior="Suspicious User Agent String")
| iplocation src_ip
| stats count by Country
| geom geo_countries allFEAtUrEs=True featureIdField="Country"
```

Figure 4.15 “Countries of Interest” dashboard pane search

The following search was used to construct the “Windows Audit Log Tampering” dashboard pane:

```
index=* (sourcetype=wineventlog* AND (EventCode=1102 OR EventCode=1100)) OR
(sourcetype=wineventlog* AND EventCode=104)
| stats count by _time EventCode Message sourcetype host
```

Figure 4.14 Sakura Dashboard in Splunk

This search was used to construct the “Attacking IPs” dashboard panel:

```
index=*
| stats count by src_ip
| chart sum(count) as "Attacking IPs"
| addtotals
```

Figure 4.17: “Attacking IPs” dashboard pane search

53

This search was used to to construct the “Port Scanning” dashboard panel:

```
index=* sourcetype=*
| stats dc(dest_port) as num_dest_port dc(dest_ip) as num_dest_ip by src_ip
| where num_dest_port >50 OR num_dest_ip >50
```

Figure 4.18: “Port Scanning” dashboard pane search

This search was used to construct the “Remote Powershell” dashboard panel:

```
index=* (sourcetype=wineventlog* AND (EventID=4103 OR EventID=400) AND
  Process_Name="*wsmprovhost.exe")
| stats count by host
```

Figure 4.19: “Remote Powershell” dashboard pane search

This search was used to construct the “Linux Shutdowns” dashboard pan

```
index=* source="var/log/messages" OR sourcetype="syslog"
| search "shutdown" OR "halt" OR "poweroff" OR "halt"
| stats count by host
```

Figure 4.20: “Linux Shutdowns” dashboard pane search

This search was used to construct the “Suspicious Network Connections” dashboard panel:

```
index=* sourcetype=wineventlog*
| search (Source_Port=3389 OR Source_Port=9100 OR Source_Port=80) AND
  (Process_Name="*svchost.exe" OR Process_Name="*notepad.exe" OR Process_Name="
    *lsass.exe", "*opera.exe", "*chrome.exe", "*firefox.exe")
| stats count by host
```

Figure 4.21: “Suspicious Network Connections” dashboard pane search

54

This search was used to construct the “Path Traversal” dashboard panel:

```
index=*  
| search "-R" OR "/bin/ls" OR "/usr/bin/file" OR "/usr/bin/find" OR "/usr/bin/  
  /mfind" OR "/tree"  
| stats count by host
```

Figure 4.22: “Path Traversal” dashboard panel search

CHAPTER 5

Results and Evaluations

This chapter will go into detail about the results found from the experiments done. This section will cover the dashboards that were created due to the simulated attacks on the network, as well as the threat hunting that was done in order to create these dashboards.

5.1 Dashboard Results

In the Sakura Dash, an overview is given that can be used to pivot to find more suspicious behavior. This alongside alerting can be used to begin to track behavior and identify malicious or suspicious actions.

Within the Countries of Interest dashboard panel, we were able to see that there were 7 connections from addresses in the United Kingdom and 62 connections from addresses in the United States were used.



Figure 5.1 Attacks by country through Countries of Interest panel

56

In the month of April, 9,155 IP addresses were found to be making connections with our environment.

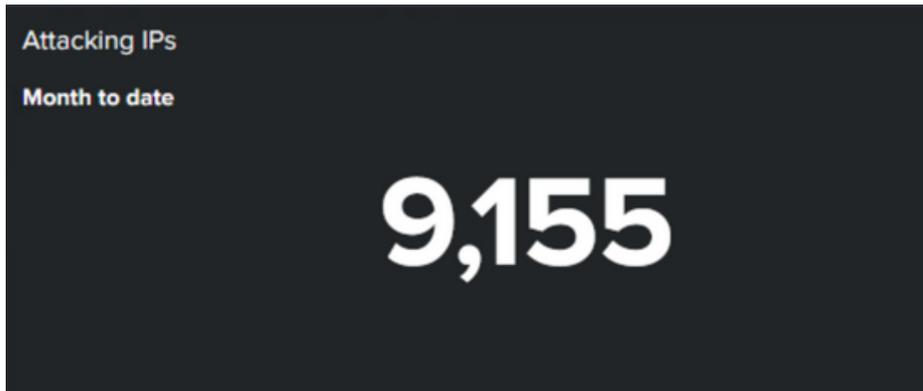
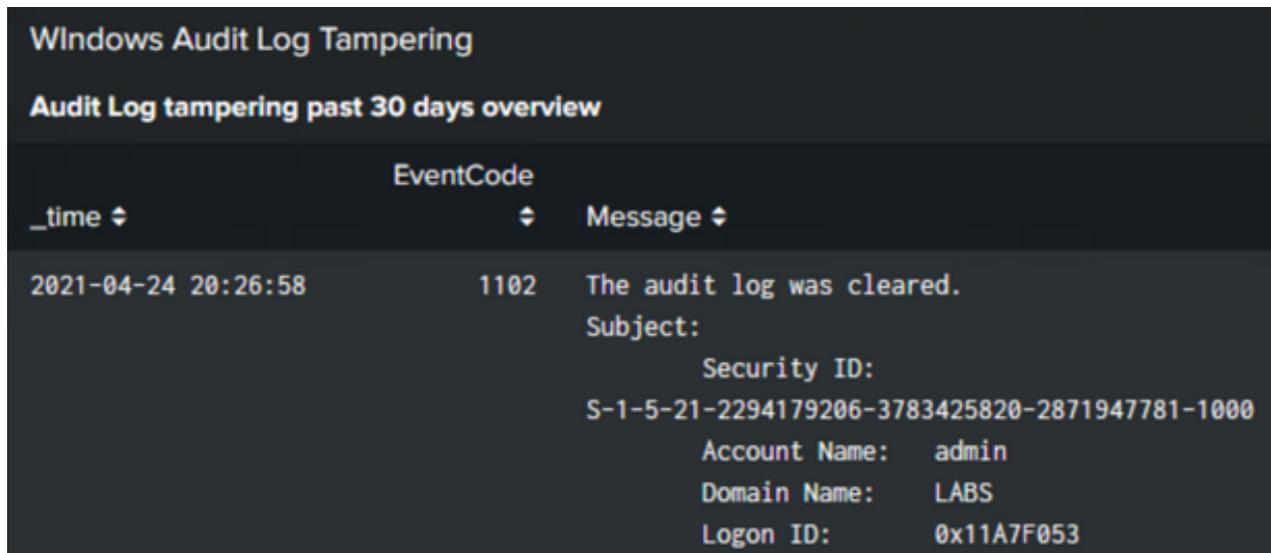


Figure 5.2: Number of attacking IPs in the month of April

On April 24th, the Windows security log on DC01 was deleted.



Windows Audit Log Tampering

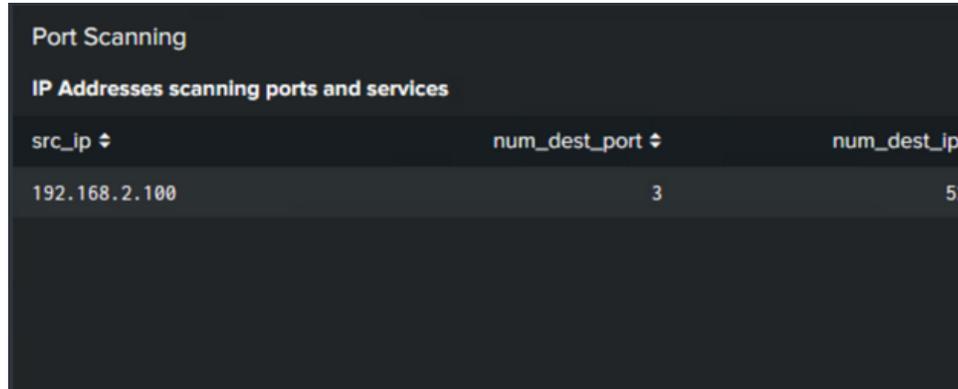
Audit Log tampering past 30 days overview

_time ↕	EventCode ↕	Message ↕
2021-04-24 20:26:58	1102	The audit log was cleared. Subject: Security ID: S-1-5-21-2294179206-3783425820-2871947781-1000 Account Name: admin Domain Name: LABS Logon ID: 0x11A7F053

Figure 5.3 Windows Audit Log cleared

57

Port scanning was found to be originating from the Kali Linux host.



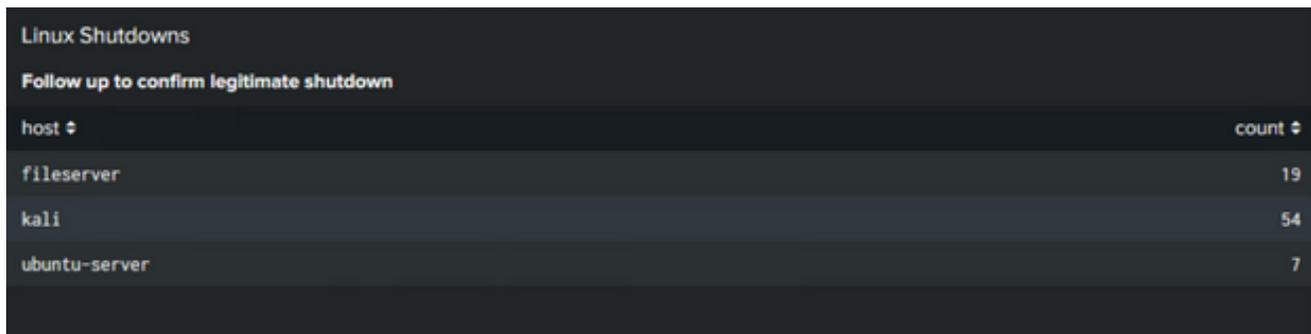
Port Scanning

IP Addresses scanning ports and services

src_ip ↕	num_dest_port ↕	num_dest_ip ↕
192.168.2.100	3	5

Figure 5.4: Port scanning originating from the Kali Linux machine

One instance of path traversal was identified with the search parameters. And the fileserver had 19 shutdowns which need to be investigated to confirm legitimacy.



Linux Shutdowns

Follow up to confirm legitimate shutdown

host ↕	count ↕
fileserver	19
kali	54
ubuntu-server	7

Figure 5.5 Shutdown totals on Linux machines

58

5.2 Threat Hunting Findings

When Brandon performed the attacks, the only knowledge we had was the week the attacks were performed over and the host that was used to perform most of the attacks. No alerts were set off by Alert Manager which meant that manual threat hunting would have to be done.

An initial search using the Splunk Time Range Picker for “Month to Date” and the search parameter “host=kali” would pinpoint logs to all information captured from March 26th-April 26th on the Kali host. Logs were found for 4/18/21 that showed scanning activity, command line parameters, and the use of tools like Metasploit. This would allow for timelining to begin on this day and for the behavior to be tracked from here.

Using the Time Range Picker, a date range of April 17 from 0:00:00 to April 19 at 23:21:00 was chosen to investigate the logs.

The image shows a screenshot of the Splunk Time Range Picker interface. It features a horizontal row of input fields. On the left, there is a dropdown menu labeled "Between" with a downward arrow. To its right are two date and time input boxes. The first box contains "04/17/2021" and the second contains "00:00:00.000", with the label "HH:MM:SS.SSS" centered below them. This is followed by the word "and" and another two date and time input boxes. The third box contains "04/19/2021" and the fourth contains "23:21:09.000", also with the label "HH:MM:SS.SSS" centered below them. To the right of these fields is an "Apply" button.

Figure 5.6 Time range set from the start of April 17 to the end of April 19

59

At 16:34:16, malicious behavior was identified with the root user installing the tool Zenmap. Zenmap is the official Nmap network scanning platform that allows for port scanning, OS fingerprinting, host scanning, etc..

Since we now know that Zenmap or Nmap is being used, we can build upon this by adding the search parameter “| search zenmap” OR “nmap” which will show events that have these two words in them within the same time range.

Four events were found, with three being Nmap scans.

```
> 4/18/21      Apr 18 16:34:16 kali sudo:    kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/apt install -y zenmap
4:34:16.000 PM  host = kali | source = /var/log/auth.log | sourcetype = auth-too_small
```

Figure 5.7 Zenmap installation log

Since we now know that Zenmap or Nmap is being used, we can build upon this by adding the search parameter “| search zenmap” OR “nmap” which will show events that have these two words in them within the same time range.

Four events were found, with three being Nmap scans.

The screenshot shows a Splunk search interface with the query "host:kali | search 'zenmap' OR 'nmap'". It displays 4 events from 4/17/21 12:00:00 AM to 4/19/21 11:21:09 PM. The search results are shown in a table format with columns for Time and Event. The events include three Nmap scans and one Zenmap installation.

Time	Event
4/18/21 4:45:05.000 PM	Apr 18 16:45:05 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/nmap -T5 192.168.8.118-112 -p- -O -v -oX scan.xml host = kali source = /var/log/auth.log sourcetype = auth-too_small
4/18/21 4:44:42.000 PM	Apr 18 16:44:42 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/nmap -T5 192.168.8.112,192.168.8.118 -p- -O -v -oX scan.xml host = kali source = /var/log/auth.log sourcetype = auth-too_small
4/18/21 4:39:35.000 PM	Apr 18 16:39:35 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/nmap -T5 192.168.8.1-254 -p- -O -v host = kali source = /var/log/auth.log sourcetype = auth-too_small
4/18/21 4:34:16.000 PM	Apr 18 16:34:16 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/apt install -y zenmap host = kali source = /var/log/auth.log sourcetype = auth-too_small

Figure 5.8 Nmap and Zenmap scans found

60

We can see that from 16:34:16 to 16:45:00 on April 18th, there was scanning activity against the network block and then specific hosts ending in 112 and 110.

Coming back to the host search, there are two commands issued to initialize and start the Metasploit console as seen here.

```
> 4/18/21 Apr 18 16:49:05 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/msfconsole
4:49:05.000 PM host = kali | source = /var/log/auth.log | sourcetype = auth-too_small

> 4/18/21 Apr 18 16:48:59 kali sudo: pam_unix(sudo:session): session closed for user root
4:48:59.000 PM host = kali | source = /var/log/auth.log | sourcetype = auth-too_small

> 4/18/21 Apr 18 16:48:58 kali sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
4:48:58.000 PM host = kali | source = /var/log/auth.log | sourcetype = auth-too_small

> 4/18/21 Apr 18 16:48:58 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/msfdb init
4:48:58.000 PM host = kali | source = /var/log/auth.log | sourcetype = auth-too_small
```

Figure 5.8: Metasploit initialized through msfconsole and msfdb init

It is likely that Metasploit has been used to attempt to communicate with the 110 and 112 host.

An authentication failure is shown as well.

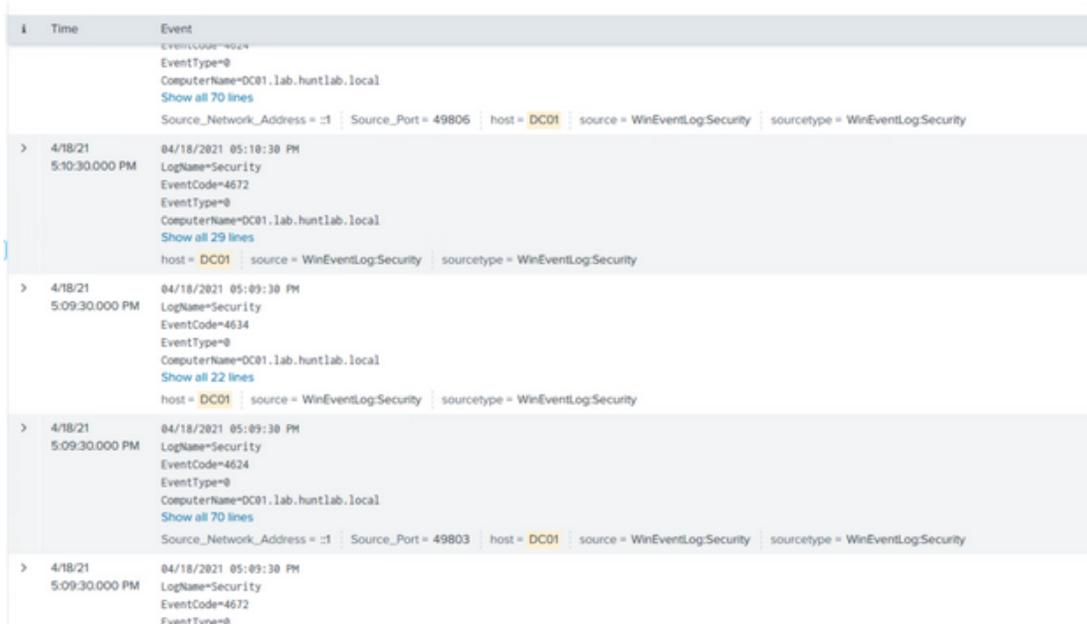
```
> 4/18/21 Apr 18 17:09:18 kali sudo: pam_unix(sudo:auth): authentication failure; logname= uid=1000 euid=0 tty=/dev/pts/2 ruser=kali rhost= user=kali
5:09:18.000 PM host = kali | source = /var/log/auth.log | sourcetype = auth-too_small
```

Figure 5.9 Authentication failure logged

Now that we know the 112 and 110 host were targeted, we can examine logs on these hosts during this time frame.

61

During the 17:09-17:11 time period, many logons and log offs are seen, some being privileged logons for DC01.



Time	Event
4/18/21 5:10:30.000 PM	04/18/2021 05:10:30 PM LogName=Security EventCode=4672 EventType=0 ComputerName=DC01.lab.huntlab.local Show all 29 lines host = DC01 source = WinEventLog:Security sourcetype = WinEventLog:Security
4/18/21 5:09:30.000 PM	04/18/2021 05:09:30 PM LogName=Security EventCode=4634 EventType=0 ComputerName=DC01.lab.huntlab.local Show all 22 lines host = DC01 source = WinEventLog:Security sourcetype = WinEventLog:Security
4/18/21 5:09:30.000 PM	04/18/2021 05:09:30 PM LogName=Security EventCode=4624 EventType=0 ComputerName=DC01.lab.huntlab.local Show all 70 lines Source_Network_Address = ::1 Source_Port = 49803 host = DC01 source = WinEventLog:Security sourcetype = WinEventLog:Security
4/18/21 5:09:30.000 PM	04/18/2021 05:09:30 PM LogName=Security EventCode=4672 EventType=0

Figure 5.10: Privileged logons to DC01

Upon further analysis, no other malicious behavior was seen on either the fileserver or DC01 host. After talking with Brandon, it was revealed he was unable to do anything but login to the domain controller due to his experience and schedule.

5.3 Honeypot Findings

Overall, the use of honeypots were found to be quite effective in the overall architecture of the network. It was found that a lot of the IP addresses that were taken up by LaBrea were attempted to be connected to, which clearly slowed down the threat from finding the actual servers on the network. These findings could be monitored through IP addresses that were connected to that did not have a machine hooked up to them.

62

The open ports from both Cowrie and Artillery helped make it harder to know which ports were open to attacks. This slowed down the threat during the simulated attack phase and would provide more time for intrusion mitigation in the case of an actual threat. Penetration tests rely heavily on this reconnaissance stage of the process to be able to conduct attacks, so having these decoy ports open helped limit the amount of valuable information that could be gained during reconnaissance.

These honeypots were found to be a useful piece of technology when securing the network, and were invaluable to providing data for the project. Furthermore, their reporting capabilities were able to be synchronized into Splunk and provide more data that can be used for threat intelligence and dashboards.

The honeypots installed offer only a small sampling of the capabilities of the technology in general, and organizations could use several different types of honeypots to meet their needs. They are continuing to become more advanced and customizable with each iteration of the technology, and they will soon be a staple in many organizations to strengthen their security posture.

63

CHAPTER 6

Conclusion and Future Work

This chapter will discuss the conclusions drawn from our experimentation and highlight future work that could be done to expand upon this project.

6.1 Conclusion

Overall, the project found that dashboards are the best way to visualize and present Threat Intelligence data to cybersecurity experts. Dashboards are a great way to visualize key metrics, and it makes the data easier to present both to cybersecurity experts, and those who may not have as much knowledge in the field. These dashboards are highly customizable, and can suit the security needs of any organization and display relevant data and fields.

The project also found that honeypots were a vital resource in detecting and identifying weak points in a network. They were a massive asset during the simulated attacks to find these threats, and can be a huge asset to an organization that wants to utilize these technologies. The honeypots were also helpful in preventing reconnaissance on the network, as the ports and IP addresses that were “opened” by the honeypots helped obscure finer details on the network.

64

The tools and measures that were set in place were able to identify attacks from an outside threat, and were able to provide a visual representation of these attacks that a cybersecurity expert could use to strengthen the network. The project was a success in being a secure network while still having the capability to detect, thwart, and rectify outside threats.

Threat intelligence is a crucial field that is rapidly developing, and this project demonstrates the importance of it. Using the newest cybersecurity technologies found with honeypots and defensive security mechanisms, outside threats were able to be detected, reported, and visualized. This is incredibly important in today's security world, as many hackers are able to bypass security measures no matter what challenge they face. It becomes much more important to detect, slow, and thwart these attacks, rather than try to prevent them outright, as that is an impossible task. Furthermore, visualizing these attacks remains exceedingly important, so that experts can present this data to the company in a way that is readable and engaging for those who are not in the industry.

6.2 Recommendations and Future Work

This project has a lot of room for improvement and future research, due to the time and manpower constraints that were faced during the research. We are only a team of two people with limited and diverted attention, and there is a lot that can be improved upon with this project.

One recommendation for future work would be to install a higher variety of honeypots across multiple systems. There were three honeypots installed on the file share server, but this is a small fraction of different types of honeypots available.

65

While the ones installed had their own unique functionality and protection, there are a multitude of attacks that can be performed on a system, and many vulnerabilities that can be utilized as an opportunity for a honeypot or alternate defensive security mechanism. These should be explored similarly in depth as to the ones in this project, and future research could be done to create an analysis on what honeypots provide the most protection and reporting. Similarly, these honeypots come with a multitude of customization options, and no one set of options perfectly suits the needs of any corporation. These options should be explored in further detail, utilizing them to the fullest extent that a corporation can use.

Another recommendation for future work on the project would be to get an increased attack simulation. This was a large part of the project that could only be done internally and by a couple of external resources willing to help out with the project. While the internal attacks did produce a good amount of data, they were also done in mind specifically with the network architecture being known, which would not replicate an outside attacker having little to no information about the systems that they are dealing with. Furthermore, a limited amount of time was put into these attacks, due to other necessities on the project. This is something that should be re-evaluated in future iterations of the project, with a proper penetration testing plan and a full report on the attacks that were done. This would help the team from the threat intelligence side, because the attacks that were done can be corroborated with the evidence found in the logging, and can help customize further searches.

66

In addition, another fully fledged penetration test should be done, but without the report being given to the security expert on the threat intelligence team. In this scenario, it would be up to the team to find these attacks and develop searches and dashboards based on them. Having no information about the attacks done would be helpful in predicting outside attacks, and analyzing the trends found from outside attacks could give information on strengthening the security posture of the organization.

Furthermore, the overall network architecture of the system could be widened in future iterations of the project. While the setup that was designed involves the core parts of a network, large corporate networks have hundreds if not thousands of different machines operating, ranging from workstations, DHCP servers, DNS servers, and management servers. Creating a large and expansive network that more accurately represents a large scale network. This would improve both the reporting capabilities of the logging system and threat intelligence, and also would produce more accurate results during the penetration testing and attack simulation phases of the project. Additionally, this would give the team knowledge on how to deploy honeypots and other defensive security mechanisms, as we cannot just rely on installing them on one server to protect valuable assets.

In additional iterations of this project, using physical servers would also be a helpful tool to improve the setup of the threat intelligence systems. Virtual machines were used to the constraints of the pandemic and the necessity for simultaneous remote work on the project, but the setup of physical servers would more accurately represent the needs of a corporation.

67

Physical servers would also present their own unique challenges that would be important to consider in the setup of these systems. Physical servers would need to be networked slightly differently than virtual servers, which could impact the attacking phase of our project. Performing these experiments with physical servers would be a helpful addition to the findings.

This is a small sampling of the potential future work that can be done. Since this is a unique concept that was invented for the purpose of this project, there are many directions future work could be taken, and not all of them could be detailed in this report. There are so many customization and different options available for honeypots, and every company comes with its own unique needs and challenges, so there could never be a project that perfectly fits the needs of every organization. This project serves as a single scenario and a baseline for threat intelligence combined with honeypot data.

68

REFERENCES

ACE Team. (2018, December 11). Secure your Network by setting up a Honeypot - Loginsoft - Cybersecurity, Software Development, Offshore Services. Retrieved October 14, 2020, from <https://www.loginsoft.com/blog/2018/11/16/secure-your-network-by-setting-up-a-honeypot/>

Chen, J. (2020, July 01). Attacker's Tactics and Techniques in Unsecured Docker Daemons Revealed. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/attackers-tactics-and-techniques-in-unsecured-docker-daemons-revealed/>

Collier, K., Dilanian, K., & Winter, T. (2020, October 31). More hospitals hit by ransomware as feds warn about cyberattacks. Retrieved November 06, 2020, from <https://www.nbcnews.com/tech/tech-news/more-hospitals-hit-ransomware-feds-warn-about-cyberattacks-n1245292>

Curtis Simpson, O. (2020, October 02). Someone died because of ransomware: Time to give hospitals emergency security care. Retrieved December 10, 2020, from <https://thehill.com/opinion/cybersecurity/519267-someone-died-because-of-ransomware-hospitals-emergency-security>

Cyber Threat Intelligence 101. (n.d.). Retrieved October 25, 2020, from <https://www.fireeye.com/mandiant/threat-intelligence/what-is-cyber-threat-intelligence.html>

Department Of Defense. (n.d.). DoD COMPUTER NOTICE. Retrieved November 06, 2020, from <https://www.dmdc.osd.mil/sevod/info/index.html>

Dominguez, A. (2017, November 17). SANS Institute: Reading Room - Intrusion Detection. Retrieved October 27, 2020, from <https://www.sans.org/reading-room/whitepapers/detection/paper/38165>

69**REFERENCES**

- Honeypots Study Guide. (n.d.). Retrieved November 07, 2020, from <https://ethics.csc.ncsu.edu/abuse/hacking/honeypots/study.php>
- Kaspersky. (2020, September 10). What is a honeypot? Retrieved October 14, 2020, from <https://usa.kaspersky.com/resource-center/threats/what-is-a-honeypot>
- Kaybay, M. E. (n.d.). Honeypots (4): Liability & Ethics. Retrieved November 7, 2020, from [http://www.mekabay.com/nwss/206i--honeypots_\(4\).pdf](http://www.mekabay.com/nwss/206i--honeypots_(4).pdf)
- Kovacs, E. (2015, January 16). False Positive Alerts Cost Organizations \$1.3 Million Per Year: Report. <https://www.securityweek.com/false-positive-alerts-cost-organizations-13-million-year-report>
- Ng, C., & Green, A. (2020, March 30). Why A Honeypot Is Not A Comprehensive Security Solution. Retrieved November 06, 2020, from <https://www.varonis.com/blog/why-a-honeypot-is-not-a-comprehensive-security-solution/>
- Oosthoek, K., & Doerr, C. (2020). Cyber Threat Intelligence: A Product Without a Process? *International Journal of Intelligence and CounterIntelligence*, 1-16. doi:10.1080/08850607.2020.1780062
- Peter, E., & Schiller, T. (2008, April 15). A Practical Guide to Honeypot. Retrieved October 25, 2020, from <https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/>
- Prizmant, D. (2020, July 15). Windows Server Containers Are Open. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/windows-server-containers-vulnerabilities/>

70**REFERENCES**

Radcliffe, J. (2007, March 16). CyberLaw 101: A primer on US laws related to honeypot deployments. Retrieved November 06, 2020, from <https://www.sans.org/reading-room/whitepapers/legal/paper/1746>

Sanders, C. (2020). Intrusion detection honeypots: Detection through deception. Oakwood, GA: Chris Sanders.

Sasson, A. (2020, May 31). Rootless Containers: The Next Trend in Container Security. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/rootless-containers-the-next-trend-in-container-security/>

Spitzner, L. (2003). Honeypots: Tracking hackers. Boston, MA: Addison-Wesley.

Symanovich, S. (2020, May 26). What is a honeypot? How it can lure cyberattackers. Retrieved October 14, 2020, from <https://us.norton.com/internetsecurity-iot-what-is-a-honeypot.html>

Threat Intelligence Definition. Why Threat Intelligence Is Important for Your Business and How to Evaluate a Threat Intelligence Program. (2019, October 02). Retrieved October 25, 2020, from <https://usa.kaspersky.com/resource-center/definitions/threat-intelligence>

Threat Intelligence: Everything You Need to Know. (2020, September 11). Retrieved October 25, 2020, from <https://www.recordedfuture.com/threat-intelligence/>

71

REFERENCES

Wallen, J. (2019, October 02). How to quickly deploy a honeypot with Kali Linux. Retrieved October 14, 2020, from <https://www.techrepublic.com/article/how-to-quickly-deploy-a-honeypot-with-kali-linux/>

What is Cyber Threat Intelligence? [Beginner's Guide]. (2020, September 17). Retrieved October 25, 2020, from <https://www.crowdstrike.com/epp-101/threat-intelligence/>

What Is Threat Intelligence? Definition and Examples. (2019, May 14). Retrieved December 01, 2020, from <https://www.recordedfuture.com/threat-intelligence-definition/>

What is SIEM? A Complete Beginner's Guide - Varonis. (2020, June 15). Retrieved December 01, 2020, from <https://www.varonis.com/blog/what-is-siem/>

Zelivansky, A. (2020, September 10). The Challenge of Persistence in Containers and Serverless. Retrieved October 14, 2020, from <https://unit42.paloaltonetworks.com/persistence-in-containers-and-serverless/>