Issues in Information Systems
Volume 25, Issue 1, pp. 265--278, 2024

DOI: https://doi.org/10.48009/1_iis_2024_122

Utilizing virtualized honeypots for threat hunting, malware analysis, and reporting

Reilly Holbel, *Mercer University, Reilly.j.holbel@live.mercer.edu*
Johnathan Yerby, *Mercer University, Yerby_jm@mercer.edu*
Warner Smith, *Mercer University, William.warner.smith@live.mercer.edu*

Abstract

Threats against information systems evolve and shift over time. In this paper, we present an analysis of active tactics, techniques, and procedures employed by threat actors against a portable virtualized honeypot. This study is an exercise of implementing multiple previously available tools to work together to create a new system. The new system allowed the researchers to gain insight into modern attacker behavior, specifically against a university network. The honeypot was deployed on an externally facing university network. We were able to capture and positively identify old and unreported ransomware and viruses. We describe our approach in the configuration and deployment of the honeypot network and a Security Incident and Event Management (SIEM) framework for real-time analysis and visualization of attacker behavior. Through analysis and reporting, we determined that threat actors relied on modified variations of publicly available cybersecurity tools and known malware and ransomware. This paper details examples of the configuration of the SIEM, examples of malware captured, and a description of what the group did with captured malware to help protect other information systems around the world. The novelty of this approach is to actively capture, neutralize, and report what attackers are doing live. We discuss mitigations for these attacks, provide a solution to slow some of the real-time attacks, and suggest further research building from this study. Fewer organizations choose to run honeypots because they are not sure of what to do with them. This study explains one of the major benefits of deciding to host a system that invites attackers in.

Keywords: Threat Intelligence, Honeypot, Portable Virtualized Environment, Threat Actor Behavior, Cyber Threat Intelligence Sharing

Introduction

Sun Tzu (2010), the ancient Chinese general, famously said, "If you know the enemy and know yourself, you need not fear the result of a hundred battles." Today, educational institutions have become prime targets for threat actors due to their rich trove of externally accessible assets and vast repositories of Personally Identifiable Information (PII) (Abril & Plant, 2007; Yerby & Floyd, 2018). Safeguarding these complex networks requires a profound understanding of threat actor behavior, encompassing their evolving patterns, tactics, techniques, and procedures (TTPs) in real-time. As cyber threats evolve rapidly, defensive measures effective last month may not suffice against the TTPs of the next.

This paper aims to contribute to enhancing cybersecurity defenses by examining threat actor behaviors in real-time. Utilizing a secure honeypot environment, we capture and analyze these TTPs in a controlled and

isolated setting. By doing so, we not only gain insights into current threats but also provide actionable intelligence to benefit others facing similar challenges in near real-time.

Literature Review

The MITRE Corporation (2023) published the ATT&CK model, a comprehensive taxonomy of adversarial tactics, techniques, and procedures (TTPs) employed by threat actors (Xiong, Legrand, Åberg, & Lagerström, 2022). Cybersecurity professionals leverage the MITRE ATT&CK and D3FEND frameworks to categorize and standardize the terminology used to describe defensive measures (MITRE, 2023).

Organizations, including higher education institutions, must maintain situational awareness of the evolving threat landscape. Cybersecurity researchers have observed a rise in the number of known threats and new, more sophisticated blended attacks (Acronis, 2022). The Tenable 2022 Threat Landscape Report (2023) highlighted that most exploits from the year stemmed from known vulnerabilities that could have been prevented or mitigated by timely patching, configuration management, and supplier assessment. Organizational cybersecurity strategies must address a wide range of issues, including monitoring for current TTPs.

Many organizations strive to adopt a proactive approach to safeguarding their data and systems, recognizing that this strategy demands substantial resources such as skilled personnel, time, and financial investment, often requiring trade-offs in other organizational areas. The commitment to cybersecurity and awareness within an organization is heavily influenced by the availability of these resources and the support of leadership (Yerby & Floyd, 2018).

Proactive cybersecurity does not mean preparing for every possible scenario or acquiring every available tool. Instead, leaders should prioritize efforts based on critical assets, potential threats, and the organization's available resources as advocated by Borum et al. (2015).

One method to draw out real-time threats facing information systems is to deploy a honeypot (Spitzner, 2003; Pa et al., 2016). A honeypot is a trap for attackers that mimics a real system with vulnerabilities. A honeynet is a network of honeypots that simulates a real network with multiple systems (Spitzner, 2003).

In this context, a honeypot refers to an information system designed to emulate vulnerable protocols and services on a network that are enticing to threat actors seeking to exploit or misuse what they perceive as vulnerable systems. While convincingly emulating these services, it simultaneously collects data about the nature of the threat actor all without permitting actual exploitation of the underlying system.

Once the organization draws out the attacks using a honeypot or honeynet, cybersecurity professionals need a method to analyze the TTPs. The activity drawn out through the honeypot can be analyzed with a Security Information and Event Management (SIEM) system or through Endpoint response systems (Spitzner, 2003).

Methodology

To gain further insights into attacks against the honeypot network, we utilized a Security Incident and Event Management (SIEM) system built on Ubuntu and the Elastic Stack. The SIEM that we used was configured to ingest, store, parse, and visualize incident data from the honeypots' services in real-time. Elastic Stack was a versatile suite of software for this task (Elastic, 2023).

In this study, we set up a virtualized honeypot on a university-hosted IP (Internet Protocol) address to analyze threats in real-time. The activity and malware were safely captured and securely handled for reverse engineering, and the malware was subsequently reported.

This paper presents a novel approach to real-time threat analysis using a virtualized honeypot employed on a university network. By leveraging a Security Information and Event Management (SIEM) system built on Ubuntu and the Elastic Stack, we were able to effectively capture and analyze attacker behavior, providing valuable insights into the tactics, techniques, and procedures (TTPs) employed by threat actors. Our findings highlight the potential of honeypots as proactive cybersecurity tools and contribute to the growing body of knowledge on real-time threat detection and mitigation.

To facilitate portability, ease of deployment, and logical isolation, our honeypot network configuration was primarily virtualized and utilized open-source and freely available software. VMware ESXi has recently switched from being available as a free version, to a paid solution. The change will have to be addressed through licensing or converting to another hypervisor. The network topology was designed to be as simple as possible while maintaining effective isolation. The hardware for this project was two desktop PCs and one machine running ESXi.

The ESXi host that this project used has extensive amounts of storage and RAM available, but it was not required for this project. This project can be run with a minimal amount of hardware. We also had one physical switch that allowed the machines on the ESXi host to transfer data to the SIEM Monitor system. The screens of the SIEM Monitor system were duplicated to large television screens in the lab, but this portion was just using resources already in place and not a requirement for the system to function. The virtualized components of the network, as shown in Figure 1, consisted of:

- A Layer-3 switch
- An ESXi Host that includes a firewall with VPN (Virtual Private Network)
- A host for the honeypot software and data collection agents connected directly to the firewall in a DMZ-like configuration
- A logically isolated host for data aggregation/SIEM functionality
- A physically isolated desktop configured for reverse engineering/malware detonation purposes. The machine has no connectivity to external devices.

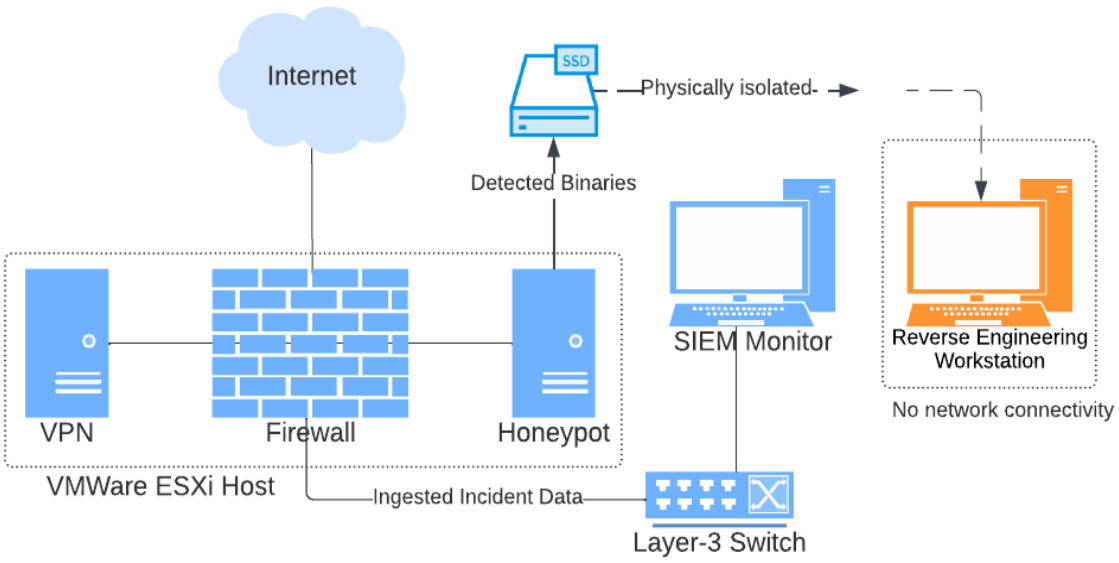

Figure 1: Topology of Honeypot

To achieve further logical isolation, our honeypot host may only communicate with the WAN via a One-to-One NAT rule or with the SIEM host via explicit port-level routing. A single physical layer-3 switch serves as a termination point for the internal portion of the virtual network, allowing physical workstations to access the front-end interface of the SIEM. Additionally, we configure two virtual machines on an isolated (physical) host that possesses no network capabilities for analysis of captured binaries. We describe the specific configuration of the main components of this topology below.

This approach enables the effective capture, analysis, and dissection of malicious traffic, providing valuable insights into the tactics, techniques, and procedures employed by threat actors. The use of open-source software and a virtualized environment facilitates portability, ease of deployment, and cost-effectiveness.

All these individual components have existed, the point of this study is how the components were combined, configured, and then the analysis and reporting out of malware. In the following section we explain the Dionaea Honeypot software that we used, then we connected it to a SIEM system to gain advanced analytics, plus we spent time to document and provide information on how this system can be put together and reproduced. On top of the honeypot, configured for safety and portability, connected to an ElasticStack SIEM system, we added a physically isolated reverse engineering workstation where researchers were able to reverse engineer malware to disarm ransomware and submit real-time information to VirusTotal's VT Hunting program (2024). The reverse engineering workstation is completely isolated. Files were encrypted, moved to a USB, then opened on the isolated machine, then decrypted, then analyzed using REMNux, Windows, StringSniffer, Ghidra, and GDB. The VT Hunting program allows cybersecurity practitioners to help create YARA rules to identify and classify malware samples (VirusTotal, 2024).

Dionaea Honeypot

Our honeypot host utilizes a customized configuration of the Dionaea honeypot software (Dinotools, 2021; Dionaea, 2023) running on the Ubuntu operating system. Dionaea employs various techniques to identify potential shellcode or executable code within detected interactions with emulated protocols. A prior study described the techniques which included dynamic, heuristics-based analysis of network traffic, and implemented in the libemu software library (Sethia & Jeyasekar, 2019). Dionaea's SMB (Server Message Block) service also incorporates hard-coded detections for the EternalBlue/DoublePulsar exploits (National Institute of Standards and Technology, 2017). Additional Dionaea functionalities are detailed by (Polychronakis & Streile, 2007; Sethia & Jeyasekar, 2019). We modified Dionaea's ihandler-based logging framework to generate per-incident JSON data for ingestion and analysis by the SIEM system.

Publicly, the honeypot appears as exposed SIP (Session Initiation Protocol), SMB, FTP (File Transfer Protocol), MSSQL (Microsoft Structured Query Language), and HTTP (Hypertext Transfer Protocol) services running on a well-documented university netblock. Threat actors, utilizing tools like Nmap, Metasploit, and Shodan, can potentially discover the honeypot through any standard means of detecting these services hosted on the WAN.

Elastic SIEM

To delve deeper into attacks targeting the honeypot network, we employed a Security Incident and Event Management (SIEM) system built upon Ubuntu LTS and the Elastic Stack (Elastic, 2023). Elastic Stack is a suite of open-source tools that are used for data search, analytics, and visualization. It is comprised of three main components:

- Elasticsearch: A distributed search and analytics engine that stores and indexes data.
- Logstash: A pipeline that collects and transforms data from a variety of sources.
- Kibana: A user interface that enables users to visualize and analyze data.

This SIEM system is designed to collect, analyze, and visually represent incident data and network metrics gathered from our honeypot. We found the Elasticsearch database and Kibana visualization tool (Figure 2), running on a Linux platform, to be a suitable solution for this purpose. We were able to easily find documentation and updates to ensure the security of this SIEM system and ensure this did not become a leak in the process. This SIEM system can be run on local hardware or in the cloud on something like EC2, but this team decided to run locally to maintain portability and control of sensitive materials. In Figure 3 the system showcases the ability to store additional metrics (such as network traffic) as distinct indices within the same unstructured database allows for temporal correlation of application-layer "incident" data from our honeypot with supplementary quantitative factors, such as the volume of a specific protocol or occurrences of broken TCP or TLS (Transport Layer Security) sessions. This correlation facilitates more informed reverse engineering analysis and the development of additional heuristic insights. Moreover, modular visualizations and transformations of ingested data are readily available, such as GeoIP enrichment (Figure 2) for the generation of real-time heatmaps of traffic origin. The project allows additional drilling down into not only types of transactions and services, but also a timeline view as shown in Figure 4.

Figure 2: Elastic Real-Time GeoIP Enrichment of Network Data

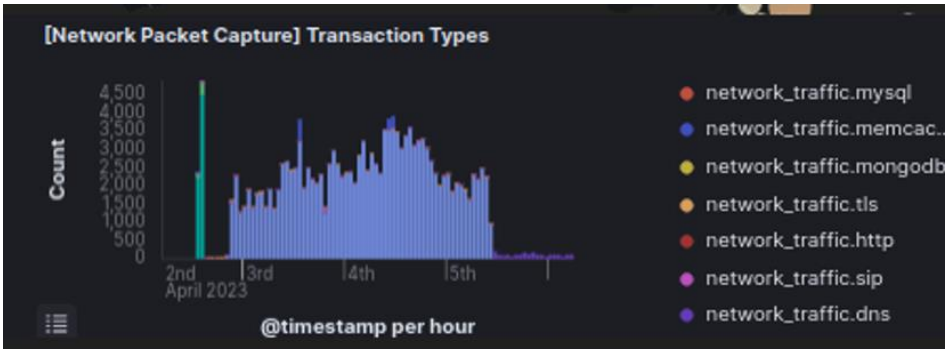

Figure 3: Elastic Honeypot Traffic Monitoring

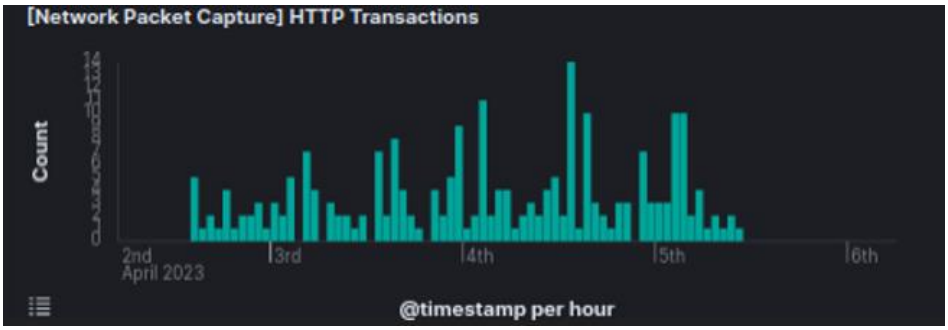

Figure 4: Service-Specific Monitoring of Honeypot Traffic (HTTP)

Reverse Engineering Workstation

To facilitate the disassembly and analysis of captured binaries, we configured a physically isolated workstation with no network interface. This workstation hosts two virtual machines: Windows 10 and REMNux, a specialized Ubuntu distribution preconfigured with a variety of static and dynamic analysis and forensics tools. Notably, we found the StringSifter utility, Ghidra, GDB, and the included PE and .dll-focused Python libraries to be particularly useful in our workflow. While the host machine lacks physical network hardware, the virtualized hosts possess emulated network interfaces, enabling traffic analysis. The Windows 10 environment operates in a "sacrificial" context for malware detonation and was configured with a disabled internal firewall and blank admin credentials.

Data Collection

After configuring and testing our environment, we exposed the network to the Internet through a university-owned gateway for four days, beginning on April 2, 2023. During this period, all incident data generated by the honeypot and raw network traffic metrics were collected in the SIEM database. The system was openly accessible to the Internet, but we did not do any extra steps to advertise or raise attention to this honeypot. Binary files detected by Dionaea were quarantined in a jailed directory, while their hashes and metadata were exported via logs ingested by the Elastic Agent. We limited the service information ingested for visualization and analysis to the FTP, SMB, TFTP, MSSQL, and SIP protocols.

The implementation of a virtualized honeypot network, in conjunction with a dedicated SIEM system and reverse engineering workstation, provided a robust and efficient platform for real-time threat analysis. By leveraging open-source tools and maintaining layers of logical isolation, we successfully analyzed and identified the characteristics of multiple captured samples, demonstrating the effectiveness of our design in detecting and understanding network-based threats.

Results

After the data collection period had elapsed, the network was disconnected from the WAN and the honeypot interfaces isolated from the virtual network. All files contained within the quarantine directory at the end of the collection period were transported to an external NTFS-formatted storage media, which was subsequently encrypted with AES-256. The captured samples were then mounted to the isolated reverse engineering host for static/dynamic analysis.

We also limit the service information we ingest for visualization and analysis to the FTP, SMB, TFTP, MSSQL, and SIP protocols. The SIEM subsequently reported 5,019 unique sessions of interaction in the targeted services as reported by the honeypot's logs (see Table 1), while the quarantine directory contained forty-eight unique binary files as determined by MD5 checksum.

Table 1: Total Detected Incident Sessions During Collection Period

FTP/TFTP	SMB	MSSQL	SIP
103	3,328	1,435	153

Throughout the data collection period, we observed consistent volumes of attempted service interactions, with one notable exception being a brute-force attempt on our SMB and MSSQL services. Dionaea logged a diverse array of commands during this hour-long period, originating from various source addresses and containing varied content. Our analysis revealed that this attack exhibited characteristics of being both distributed and automated, as evidenced by the absence of a corresponding increase in captured binaries.

During the data collection, we prioritized analyzing the most prevalent attacks detected during our data collection period. During this collection period the attacks centered around MSSQL, SMB, and SIP. The

analysis is discussed in the following section. Upon detection of an executable file by the honeypot, we proceeded with the disassembly of the executable, utilizing correlated incident and network traffic data exported from the honeypot and stored in the SIEM database.

Our investigation revealed that nearly every threat actor interacted with the emulated services multiple times, both before and after attempting exploitation. This pattern of interaction before and after exploitation supports what we would expect to see following Cyber Kill Chain (2015) framework. Cyber Kill Chain is a security model delineating the stages of a cyberattack with the aim of identifying and thwarting adversarial activity. The early interpretation was that attacks occur in a linear fashion from reconnaissance, weaponization, delivery, exploitation, installation, command and control, and then actions on objectives. A modernized interpretation in Lockheed Martin's (2015) Gaining the advantage: Applying cyber kill chain methodology to network defense, is that attackers do not move in a predictable linear fashion, but their goals still include each step completed before achieving actions on objectives.

The recurrent behaviors of attackers revisiting stages within the Cyber Kill Chain framework mirror contemporary elusive tactics, offering valuable insights into the operational realities of cyberattacks on information systems. Figure 5 shows this recurrent timeline.

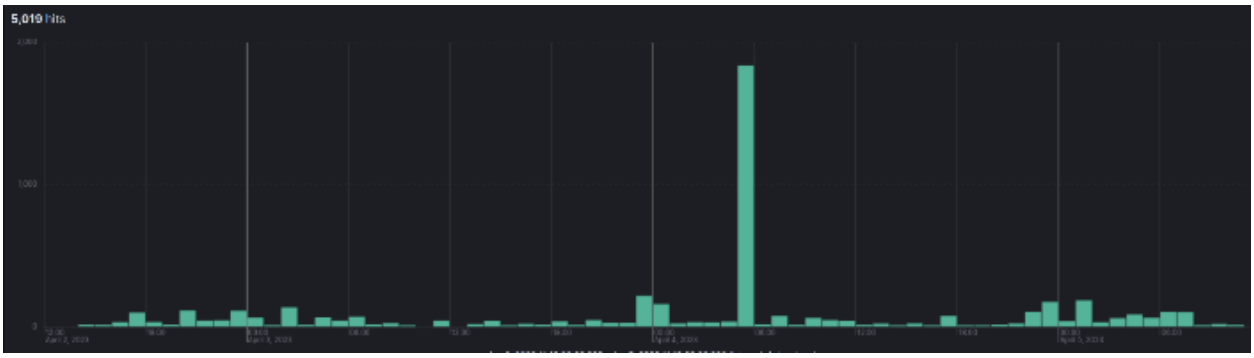

Figure 5: Volume of Detected Incidents Over Time

CLR-Based Exploit Loading against Microsoft SQL (MSSQL) Services

After detecting several strangely formatted SQL statements against Dionaea's MSSQL module, we identified several attempts to load privilege escalation exploits in memory by leveraging the Microsoft SQL (MSSQL) Common Language Runtime services. Our would-be attackers connected to the database with what they believed were valid credentials and initially attempted to load a .NET .dll file as a CLR object using the 'Create Assembly' command. Subsequently, the attackers created a procedure referencing this object to run the malicious assembly with the 'Create Procedure' and 'EXEC' statements (Figure 6).

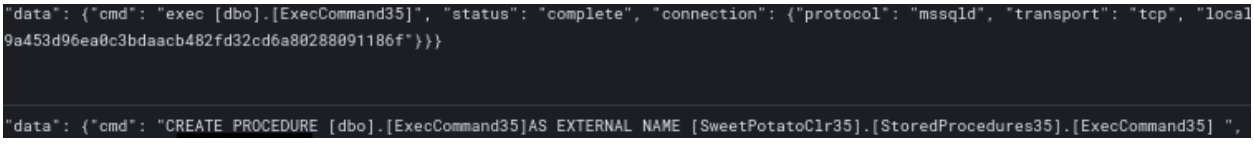

Figure 6: Attempted Procedure Loading

The attacker covered their tracks by dropping the procedure and associated assembly objects. This technique resulted in an unusually large initial Transact-SQL statement being logged into our database, containing the entire compiled .NET library (Figure 7).

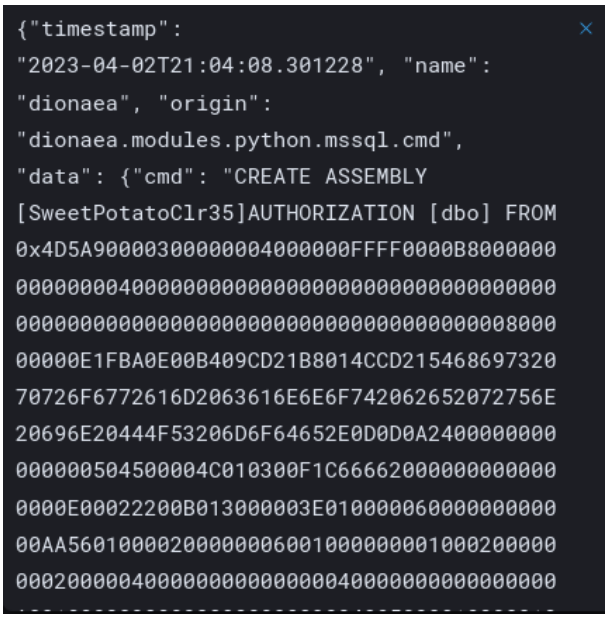

Figure 7: Initial Loading of JuicyPotato Exploit

Cross-referencing strings scraped from this assembly against GitHub led us to a self-proclaimed "weaponized" variant of the SweetPotato Windows privilege escalation exploit family, dubbed "JuicyPotato." SweetPotato exploits leverage flaws in NTLM authentication to elevate existing users with the "SeImpersonate" privilege to SYSTEM for secondary loading in an elevated context or to achieve shell access (Trotta, 2021; Breen & Mallz, 2016).

The use of JuicyPotato in tandem with an Execute-Assembly command for initial exploitation is well-documented and available in a modular fashion as part of command-and-control frameworks, such as Cobalt Strike (Coburn, 2020). A report by Blackberry (2023) demonstrates that usage of the SweetPotato family as a secondary exploit is also still prevalent outside of this context and corroborates its active use.

SipVicious

We identified and profiled multiple attacks against the Session Initiation Protocol of the honeypot. By analyzing the network-layer attributes of ingested traffic (user agent, packet body) combined with the session-aware incident data provided by Dionaea leads us to the conclusion that threat actors have adopted another open-source tool, SipVicious, originally developed for security researchers to assess the confidentiality and authentication strength of Private Branch Exchange systems (Enable Security, 2022). It is worth noting that Dionaea's SIP protocol emulation is not as robust as other services, and only the initial handshake is managed. We posit that this is partially the reason for the relatively low volume of requests observed. Identification was rendered trivial once multiple user-agents and SIP display names were detected, as our threat actor did not seem to be interested in further configuration of the tool (Figure 8).

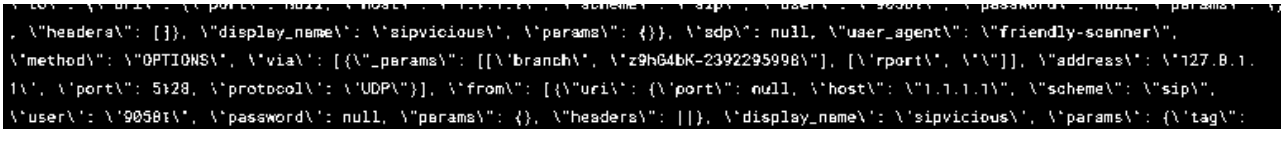

Figure 8: SipVicious Scanner Packets

WannaCry Ransomware

We were successful in identifying several live samples of reactivated WannaCry ransomware binaries through further attacks against the SMB service of our honeypot. WannaCry is a Windows (PE) based malware classified as a "crypto-ransomware worm" that initially gained notoriety for its use in attacking and successfully encrypting substantial amounts of data in hospitals and healthcare systems in May 2017 (Hitrust, 2017). Notably, the malware leveraged two then zero-day exploits extant in certain versions of the SMB protocol dubbed EternalBlue and DoublePulsar that allow for arbitrary code execution on servers running the vulnerable SMB version via a mangled message body that disagrees with the header size (MITRE, 2017) This vulnerability allows the software to self-propagate to other external and internal systems and is responsible for its worm-like behavior. While the initial infestation of this worm was successfully halted by initiating a then-global "kill switch" in the form of an unregistered domain hard-coded into the malware, it is trivial using a hex editor or similar software to disengage this kill-switch and render the malicious binary functional again. We observed such a technique in our captured sample (Figure 9).

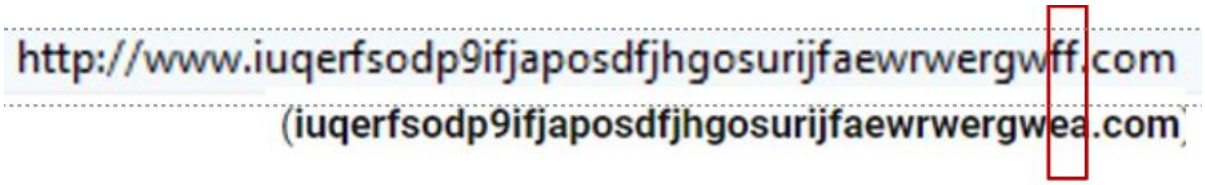

Figure 9: Top: String Extracted from Captured Sample. Bottom: Original Kill-Switch Domain (WhoAPI, 2017)

The presence of a unique "PlayGame" function in the export table of the executable, responsible for the worm's persistence on the system, along with the strings "WanaCrypt0r" and "WNcry@2ol7," and the kill-switch domain, facilitated the preliminary identification of the malware when cross-referenced with the existing work by Berry et al. (2017) (Figure 10).

0002E63A	Windows 2000 5.0
0002E68C	\\192.168.56.20\IPC$
000313B4	kernel32.dll
000400D8	WanaCrypt0r
000400F0	Software\
000403D4	.der
000403E0	.pfx
000403EC	.key
000403F8	.crt
00040404	.csr
00040410	.p12
0004041C	.pem
00040428	.odt

Figure 10: WannaCry File Extension Strings, Hardcoded Addresses

The team successfully loaded the captured .dll into memory in our reverse-engineering environment using rundll32.exe and observed its self-propagation behavior on the virtual network through malicious SMB requests with randomized destinations. This randomization aimed to avoid the previously discovered kill-

switch by registering a new hardcoded domain. The theory behind the domain query function in WannaCry is to check if it is inside a sandbox environment (Cloudflare, 2024). This modern version was designed to be immune to the sinkhole that halted the original WannaCry attack.

We found that the vulnerabilities exploited by WannaCry have been patched by manufacturers through product updates. However, other malware can still utilize parts of the original code to launch new or modified versions of the attack.

Additional Obfuscation Techniques

Dozens of threat actors interacted with our emulated services multiple times throughout the duration of the detection period, and more than half used multiple obfuscation techniques throughout. These include base-64 encoding of malicious SQL statements (Fig. 11) that included job creation. Creating stored procedures or jobs allowed attackers to launch attacks based on logic conditions or recall attacks at will. These obfuscation techniques, while rudimentary, may still serve to frustrate detection and analysis.

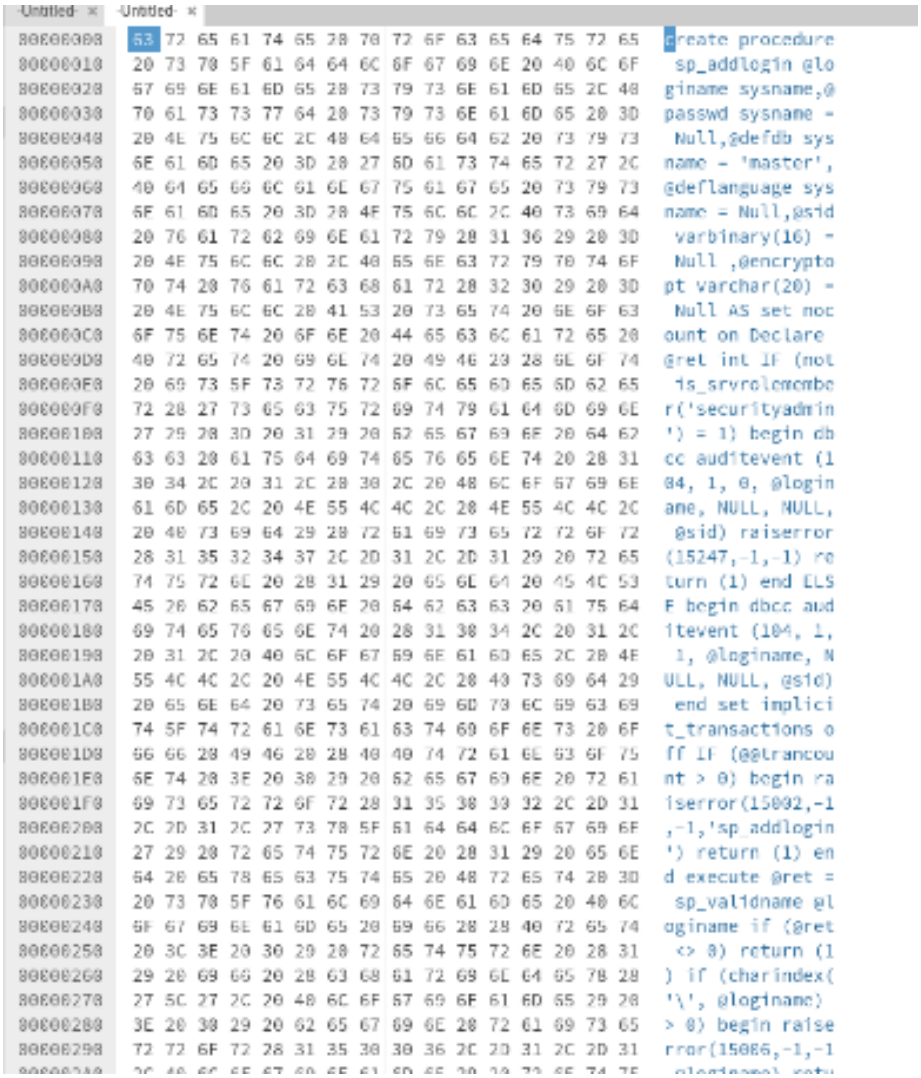

Figure 11. Encoded Procedure Creation

Discussion and Limitations

This study has limitations that warrant consideration for future work. The primary aim of the project was to provide students with the resources and environment to safely explore a topic beneficial to their upcoming careers, while also contributing to the university. Franks & Yerby (2014) employed a similar approach, where a graduating student addressed a student-centered, complex, real-world problem. This method fosters the creation of new knowledge, systems, and skills. However, a significant challenge in undertaking such niche projects is the substantial time and resources required.

In this project, the researchers benefited from an existing lab undergoing a transition, which allowed students access to several powerful servers, isolated networking, and a dedicated space. The system was designed and built in a modular format to facilitate replication of the study. Although the honeypot configuration used open-source technologies, it relied on VMware ESXi Host, potentially complicating efforts to port the honeynet infrastructure to other environments.

The Dionaea libraries implemented for shellcode/binary detection, while largely behavioral in nature, may not successfully identify samples against certain obfuscation techniques. Querying SIEM log files for various encodings of magic bytes yielded more potential binaries than initially detected programmatically, but these were not included in our collected incident dataset. Similarly, heuristic and hash-based detection tools like VirusTotal are only applicable to known malicious samples and may not always be accurate. Consequently, some false negatives in our dataset are anticipated, suggesting that the volume of detected attacks might be greater than reported here. Nonetheless, all samples noted in this study were identified with high confidence.

Furthermore, the Dionaea honeypot is limited in the depth of certain services it emulates. While the SQL, SMB, and FTP services are robust and highly interactive, the SIP and HTTP protocols are less so. This limitation affects both detection capabilities and the completeness of logged sessions, as some sessions may end prematurely due to unsupported behaviors or inappropriate replies (Dinotools, 2021). To replicate this study in other environments, enhancements to the HTTP and SIP protocols would be necessary.

The high volume of perceived vulnerable services emulated on our university netblock may also act as a deterrent to potential attackers, raising a "red flag" and dissuading interaction with the virtual network. In a typical university or business setting, exposing SIP, MSSQL, SMB, and FTP services to a wide area network is unlikely. This could result in attracting more automated and less sophisticated attackers due to the highly vulnerable posture presented.

Relying on specific detection techniques, such as those for EternalBlue and DoublePulsar exploits, might introduce a detection bias. Cyber threats evolve rapidly, and depending solely on hard-coded detections may lead to missing newer or more sophisticated threats. This study aimed to learn and observe a wide range of threats, acknowledging that some might have been missed. It is important for organizations to consider this when implementing such a system. While this solution provides valuable real-time insights into attack attempts, it is not a comprehensive solution.

Finally, the scope of our research is limited to observing threat actor behavior against a known university IP address range, which restricts our analysis. Solutions for expanding our topology to different environments are discussed later in this paper.

Future Work

This study was conducted with the primary objective of gathering live attack data and malware samples to advance research and identify real-time threats targeting a university network. Our analysis of captured

binaries aimed to illuminate prevailing attacks on university networks. To facilitate broader understanding and enhance detection capabilities, we made our dataset accessible in CSV format, containing file hashes and source network addresses. Furthermore, identified malware was promptly uploaded to VirusTotal in a proactive effort to aid real systems encountering cyber threats.

Moreover, we are dedicated to proposing tailored mitigations for detected exploits. For instance, the SweetPotato family exploits poorly configured maintenance account credentials, necessitating their removal from Windows servers and workstations. Additionally, we advocate for the application of patches to SMB services and the implementation of firewall rules to block external traffic bound for vulnerable ports, such as 135-139 for MSSQL servers, mitigating NetBIOS/RPC exploits.

The design of our network infrastructure prioritized portability and deployment flexibility. Detailed descriptions provided in this paper aim to facilitate replication with variations, such as transitioning from a university to a cloud-hosted environment or modifying services within the Dionaea honeypot. Future research avenues may involve comparing changes in attack tactics, techniques, and procedures resulting from alterations in location or variables. Additionally, containerization of components presents a promising avenue for further exploration.

The inherent flexibility and portability of our infrastructure render it an attractive platform for experimentation, both for organizations seeking to comprehend real-time threats to their systems and for academic endeavors aimed at documenting and understanding evolving behaviors across diverse environments.

References

Abril, P. S., & Plant, R. (2007). The patent holder's dilemma: Buy, sell, or troll? *Communications of the ACM*, *50*(1), 36-44.DOI: https://doi.org/10.1145/1188913.1188915.

Acronis. (2022, January 4). How cyberthreats and cybersecurity are evolving in 2022. https://www.acronis.com/en-us/blog/posts/how-cyberthreats-and-cybersecurity-are-evolving-in-2022/

Berry, A., Homan, J. and Eitzman, R. (2017). WannaCry malware profile. Mandiant. https://www.mandiant.com/resources/blog/wannacry-malware-profile

Blackberry (2023, October 18). Silent skimmer: Online payment scraping campaign shifts targets from APAC to NALA https://blogs.blackberry.com/en/2023/09/silent-skimmer-online-payment-scraping-campaign-shifts-targets-from-apac-to-nala

Borum, R., Felker, J., Kern, S., Dennesen, K., & Feyes, T. (2015). Strategic cyber intelligence. *Information & Computer Security*, *23*(3), 317-332.

Breen, S. & Mallz, C. (2016). *Rotten potato – privilege escalation from service accounts to system, foxglovesecurity.com*. https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/

Cloudflare (2024, June). *What was the WannaCry ransomware attack?* Retrieved from https://www.cloudflare.com/learning/security/ransomware/wannacry-ransomware/

Coburn, C. (2020, April 14). *SweetPotato – Service to SYSTEM* https://www.pentestpartners.com/security-blog/sweetpotato-service-to-system/

DinoTools. (2021, January 3). *DinoTools/Dionaea: Home of the Dionaea honeypot*. [Computer software]. GitHub. https://github.com/DinoTools/dionaea

Dionaea. (2023, November 2). Welcome to Dionaea's documentation! — Dionaea 0.11.0 documentation. Retrieved November 16, 2023, from https://dionaea.readthedocs.io/en/latest/

Elastic. (2023, November 2). SIEM & Security Analytics | Elastic Security | Elastic SIEM. Retrieved November 16, 2023, from https://www.elastic.co/security/siem

Enable Security (2022). Enable Security/sipvicious: Welcome to SIPVicious OSS security tools. [Computer Software].

Franks, S., & Yerby, J. (2014). Creating a low cost supercomputer with Raspberry Pi. In *Proceedings of the Southern Association for Information Systems Conference*. https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1007&context=sais2014

Hitrust (2017). WannaCry postmortem: Early warning indicators and lessons learned for the healthcare industry https://hitrustalliance.net/wannacry-post-mortem-early-warning-indicators-lessons-learned-healthcare-industry/

Lockheed Martin. (2015, September). *Gaining the advantage – applying Cyber Kill Chain® ...* Cyber Kill Chain | Lockheed Martin. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf

MITRE. (2017). CVE-2017-0144 Windows SMB remote code execution vulnerability. Retrieved from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144

MITRE ATT&CK. (2024, May). MITRE ATT&CK [Website]. Retrieved from https://attack.mitre.org/

Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., & Rossow, C. (2016). IoTPOT: A novel honeypot for revealing current IoT threats. *Journal of Information Processing*, *24*(3), 522-533.

Polychronakis, M., Anagnostakis, K. G., & Markatos, E. P. (2007). Emulation-based detection of non-self-contained polymorphic shellcode. In *Recent Advances in Intrusion Detection: 10th International Symposium, RAID 2007, Gold Coast, Australia, September 5-7, 2007. Proceedings 10* (pp. 87-106). Springer Berlin Heidelberg.

Sethia, V. & Jeyasekar, A. (2019). Malware capturing and analysis using Dionaea honeypot. 1-4. 10.1109/CCST.2019.8888409

Spitzner, L. (2003, December). Honeypots: Catching the insider threat. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.* (pp. 170-179). IEEE.

Spitzner, L. (2003, September 15). Endpoint Protection - Symantec Enterprise (broadcom.com)

Tenable. (2023). *Tenable 2022 Threat Landscape Report*. https://www.tenable.com/cyber-exposure/tenable-2022-threat-landscape-report

Trotta, G. (2021, March 9). A sugared version of RottenPotatoNG, with a bit of juice, i.e., another Local Privilege Escalation tool, from a Windows Service Accounts to NT AUTHORITY\SYSTEM. [Computer software]. GitHub.

Tzu, S. (2010). *The art of war.* (T. Cleary, Trans.). Boston: Shambala. (Original work published in 6th century BCE)

VirusTotal. (2024). What's VT hunting? https://docs.virustotal.com/docs/whats-vthunting

WhoAPI (2017). What is the domain name that stopped WannaCry? *https://whoapi.com/blog/what-is-the-domain-name-that-stopped-wannacry/*

Xiong, W., Legrand, E., Åberg, O., & Lagerström, R. (2022). Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. Software and Systems Modeling, 21(1), 157-177.

Yerby, J., & Floyd, K. (2018, August). Faculty and staff information security awareness and behaviors. In *Journal of The Colloquium for Information Systems Security Education* (Vol. 6, No. 1, pp. 23-23). https://cisse.info/journal/index.php/cisse/article/view/90/CISSE_v06_i01_p05.pdf