ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εγκατάσταση, Αξιολόγηση και Βελτιστοποίηση του Open Source Συστήματος HELK για την Υποστήριξη SOC (Κέντρων Επιχειρήσεων Ασφαλείας) για SMEs

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΚΑΝΤΖΗ ΔΙΟΝΥΣΙΟΥ

Εξεταστική Επιτροπή

Καθηγητής Σωτήριος Ιωαννίδης

Καθηγητής Μιχαήλ Γ. Λαγουδάκης

Καθηγητής Απόστολος Δόλλας

TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Deployment, Testing, Evaluation, and Improvement of the HELK Open Source System for SOCs (Security Operations Centers) Supporting the SME Sector

THESIS

of

SKANTZIS DIONYSIOS

Examination Committee

Professor Sotirios Ioannidis

Professor Michael G. Lagoudakis

Professor Apostolos Dollas

ABSTRACT

In an era marked by escalating cyber threats and an increasingly interconnected digital landscape, the need for robust cybersecurity measures is paramount. Small and medium-sized enterprises (SMEs), often constrained by budgetary considerations, face the challenge of securing their digital infrastructure effectively. Traditional commercial Security Information and Event Management (SIEM) solutions, such as QRadar and Splunk, while powerful, may impose significant financial burdens on SMEs.

This thesis embarks on a comprehensive exploration of the feasibility and practicality of SMEs adopting an open-source SIEM system known as HELK (Hunting ELK). The study delves into every facet of deploying, installing, and configuring HELK, creating a roadmap accessible to businesses of varying technical proficiencies. Moreover, it elucidates the intricate processes involved in configuring a Windows Host to seamlessly transmit logs to the HELK SIEM. A thorough analysis of HELK's inner workings is undertaken, followed by a rigorous evaluation of its efficacy in detecting simulated cyberattacks. By subjecting the SIEM to a series of carefully orchestrated attacks, this research assesses its ability to identify and mitigate threats. The findings shed light on HELK's strengths and weaknesses, offering insights into potential enhancements. In light of the above, this thesis endeavors to address a critical question: can SMEs rely on the open-source HELK SIEM as a cost-effective alternative to commercial counterparts? By navigating the intricacies of SIEM deployment, testing its performance, and scrutinizing its practicality, this research provides valuable guidance to SMEs seeking comprehensive yet budget-conscious cybersecurity solutions.

The findings aim to empower SMEs to make informed decisions regarding their cybersecurity strategies, enhancing their resilience in an ever-evolving threat landscape. Through this thesis we delve into an in-depth exploration of HELK's benefits, weaknesses, and potential avenues for improvement, offering a holistic perspective on its viability as a cybersecurity solution for SMEs. This study, which combines hands-on experimentation with a comprehensive evaluation of HELK's capabilities, serves as a testament to the potential of open-source SIEM solutions in supporting the cybersecurity needs of SMEs while minimizing financial constraints.

ΠΕΡΙΛΗΨΗ

Σε μια εποχή που χαρακτηρίζεται από κλιμακούμενες απειλές στον κυβερνοχώρο και ένα όλο και πιο διασυνδεδεμένο ψηφιακό τοπίο, η ανάγκη για ισχυρά μέτρα κυβερνοασφάλειας είναι πρωταρχικής σημασίας. Οι μικρές και μεσαίες επιχειρήσεις (ΜΜΕ), που συχνά περιορίζονται από δημοσιονομικές αδυναμίες, αντιμετωπίζουν προκλήσεις στην αποτελεσματική διασφάλισης της ψηφιακής τους υποδομής. Οι παραδοσιακές εμπορικές λύσεις Διαχείρισης Πληροφοριών και Εκδηλώσεων Ασφαλείας (SIEM), όπως το QRadar και το Splunk, αν και ισχυρές, ενδέχεται να επιβάλλουν σημαντικές οικονομικές επιβαρύνσεις στις ΜΜΕ.

Αυτή η διατριβή ξεκινά μια ολοκληρωμένη διερεύνηση της σκοπιμότητας και της πρακτικότητας υιοθέτησης ενός συστήματος SIEM ανοιχτού κώδικα γνωστό ως HELK (Hunting ELK) από τις ΜΜΕ. Η μελέτη εμβαθύνει σε κάθε πτυχή της ανάπτυξης, εγκατάστασης και διαμόρφωσης του HELK, δημιουργώντας έναν οδικό χάρτη προσβάσιμο σε επιχειρήσεις διαφορετικών τεχνικών δεξιοτήτων. Επιπλέον, διευκρινίζει τις περίπλοκες διαδικασίες που εμπλέκονται στη διαμόρφωση ενός κεντρικού υπολογιστή Windows για την απρόσκοπτη μετάδοση αρχείων καταγραφής στο HELK SIEM. Γίνεται μια διεξοδική ανάλυση των εσωτερικών λειτουργιών του HELK, ακολουθούμενη από μια αυστηρή αξιολόγηση της αποτελεσματικότητάς του στον εντοπισμό προσομοιωμένων κυβερνοεπιθέσεων.

Υποβάλλοντας το SIEM σε μια σειρά από προσεκτικά ενορχηστρωμένες επιθέσεις, αυτή η έρευνα αξιολογεί την ικανότητά του να εντοπίζει και να μετριάζει τις απειλές. Τα ευρήματα ρίχνουν φως στα δυνατά και αδύνατα σημεία του HELK, προσφέροντας πληροφορίες για πιθανές βελτιώσεις. Υπό το πρίσμα των παραπάνω, αυτή η διατριβή προσπαθεί να απαντήσει σε ένα κρίσιμο ερώτημα: μπορούν οι ΜΜΕ να βασίζονται στο ανοιχτού κώδικα HELK SIEM ως μια οικονομικά αποδοτική εναλλακτική λύση σε σχέση με τις εμπορικές λύσεις; Με την πλοήγηση στις περιπλοκές της ανάπτυξης του SIEM, δοκιμάζοντας την απόδοσή του και ελέγχοντας την πρακτικότητά του, αυτή η έρευνα παρέχει πολύτιμη καθοδήγηση σε ΜΜΕ που αναζητούν ολοκληρωμένες λύσεις κυβερνοασφάλειας με γνώμονα τον προϋπολογισμό.

Τα ευρήματα στοχεύουν να δώσουν τη δυνατότητα στις ΜΜΕ να λαμβάνουν τεκμηριωμένες αποφάσεις σχετικά με τις στρατηγικές τους στον κυβερνοχώρο, ενισχύοντας την ανθεκτικότητά τους σε ένα συνεχώς εξελισσόμενο τοπίο απειλών. Μέσω αυτής της διατριβής εμβαθύνουμε σε μια εις βάθος διερεύνηση των πλεονεκτημάτων, των αδυναμιών και των πιθανών οδών βελτίωσης του HELK, προσφέροντας μια ολιστική προοπτική για τη βιωσιμότητά του ως λύση κυβερνοασφάλειας για τις ΜΜΕ. Αυτή η μελέτη, η οποία συνδυάζει τον πρακτικό πειραματισμό με μια ολοκληρωμένη αξιολόγηση των δυνατοτήτων του HELK, χρησιμεύει ως απόδειξη για τις δυνατότητες των λύσεων SIEM ανοιχτού κώδικα για την υποστήριξη των αναγκών κυβερνοασφάλειας των ΜΜΕ, ελαχιστοποιώντας ταυτόχρονα τους οικονομικούς περιορισμούς.

CONTENTS

Table of Contents

CHAPTER 1: Introduction

1.1 THESIS SCOPE AND GOALS

Small and medium-sized enterprises (SMEs) stand as the lifeblood of economies around the world, contributing significantly to innovation and employment. This is even more prevalent in the EU, where SMEs represent 99% of all business activity and contribute to over 50% of Europe's GDP. Despite their economic vitality, these businesses often operate in the shadows of larger corporations, facing unique challenges and resource constraints.

As per the insights gleaned from the ENISA study "Cybersecurity for SMEs"[1], the predominant challenge facing most Small and Medium Enterprises in the realm of cybersecurity pertains to awareness and commitment. More precisely, it is the lack of awareness regarding the perils posed by cybercrime, financial constraints, and the absence of cybersecurity frameworks and tailored solutions designed to meet the specific needs of SMEs that emerge as the foremost obstacles to enhancing the overall security posture within the SME sector.

This thesis embarks on a comprehensive exploration of the open-source HELK (Hunting ELK) Security Information and Event Management (SIEM) system, with a primary focus on its applicability within the Small and Medium-sized Enterprise (SME) sector. As the cybersecurity landscape continues to evolve, the need for effective, accessible, and cost-efficient security solutions for SMEs becomes increasingly vital. This chapter outlines the overarching goals and the comprehensive scope of this thesis. It establishes the foundation for the subsequent chapters, providing a clear roadmap for the analysis, evaluation, and improvement of the HELK SIEM system as an accessible and cost-effective cybersecurity solution for SMEs.

1.1.1 SCOPE

The scope of this thesis encompasses a multifaceted analysis and evaluation of HELK SIEM, addressing several key aspects:

- **Ease of Deployment**
  One of the primary objectives is to assess the ease of deploying HELK on an Ubuntu host, ensuring that SMEs with limited IT resources can effectively implement this SIEM solution. This involves a step-by-step guide to HELK installation and configuration, allowing for a straightforward deployment process.

- **Detectability of Threats**
  This research scrutinizes HELK's efficacy in detecting and responding to security threats. To achieve this, we simulate various attacks on a Windows host configured to send logs to the HELK SIEM. The detection capabilities of HELK in the context of real-world attack scenarios are closely examined and evaluated.

- **Improvements**
  HELK's adaptability and potential for customization are assessed. The research explores the ease with which SMEs can tailor HELK to suit their specific security needs and requirements. Identifying areas for improvement and fine-tuning the HELK SIEM system is a key aspect of this evaluation.

- **Advantages and Restrictions**
  The advantages and potential drawbacks of HELK as an open-source SIEM solution for SMEs are examined comprehensively. This includes an analysis of the benefits, such as cost-effectiveness and scalability, as well as the inherent limitations and challenges associated with an open-source approach.

1.1.2 METHODOLOGY

The methodology adopted in this thesis combines practical experimentation and evaluation with a thorough analysis of HELK's features and capabilities. The research methodology comprises three primary phases:

- **Deployment of HELK on an Ubuntu Host**
  The initial phase involves the deployment of HELK on an Ubuntu host. Detailed step-by-step instructions for the installation and configuration of HELK are provided. This phase aims to ascertain the user-friendliness and accessibility of HELK's deployment process.
- **Configuration of Windows Host for Log Forwarding**
  To evaluate HELK's log collection capabilities, a Windows host is configured to send event logs to the HELK SIEM. This step involves setting up the necessary log forwarding configurations to ensure seamless data transfer.
- **Simulated Attack Scenarios**
  To assess HELK's threat detection capabilities, simulated attack scenarios are executed. A Kali Linux virtual machine is utilized to launch a series of security attacks on the Windows host. The detection and response mechanisms of HELK are closely monitored and analyzed throughout these simulations.

1.2 CHAPTER OUTLINE

In this section, we provide an overview of the structure and content of each chapter in this thesis, offering a clear roadmap of the topics covered:

- **CHAPTER 2: Cybersecurity Fundamentals**
  This chapter delves into the foundational aspects of cybersecurity, discussing its historical evolution and growing significance in the modern world. It outlines the most common types of cyber threats and introduces the MITRE ATT&CK Framework, providing essential context for understanding the subsequent chapters.

- **CHAPTER 3: Threat Hunting Defined**
  Chapter 3 focuses on defining threat hunting as a proactive cybersecurity practice. It elaborates on essential terms such as Security Operations Center (SOC) and Security Information and Event Management (SIEM) systems. The chapter delves into the inner workings of SIEMs, exploring their functions and highlighting their role in modern cybersecurity. Additionally, it examines the advantages of integrating SIEMs into enterprise security operations and addresses the cost considerations associated with these solutions.

- **CHAPTER 4: The HELK Platform**
  Chapter 4 introduces the HELK SIEM, a pivotal element of this thesis. It provides an in-depth understanding of HELK's core components, including the ELK (Elasticsearch, Logstash, Kibana) stack, Beats, Kafka, and others. Each component's function and contribution to the overall capabilities of HELK are detailed. The chapter concludes with a comprehensive guide to installing HELK and a brief overview of its Sigma rules, which play a crucial role in threat detection.

- **CHAPTER 5: The HELK Lab Environment**
  Chapter 5 outlines the lab environment established for this research. It delineates the key components, including the HELK SIEM deployed on Ubuntu, the Windows Victim Host configured for log forwarding, and the Kali Linux virtual machine responsible for simulating attacks. This chapter further details the configuration process for the Windows Host to generate and forward logs, involving Sysmon, Winlogbeat, and Windows Policy configurations.

- **CHAPTER 6: Analyzing Attacks and HELK's Response**
  Chapter 6 is dedicated to the analysis of simulated attacks within the lab environment. The Kibana user interface (UI) is explored in-depth, providing a detailed description of each attack, including Reconnaissance, Brute Force, Malware Injection, Persistence, and Covering Tracks. The chapter also examines the outcomes produced by HELK for each attack, scrutinizing the system's detection capabilities.

- **CHAPTER 7: Experimental Results and discussion**
  Chapter 7 serves as a critical evaluation of HELK's performance. It investigates the benefits and limitations of HELK as an open-source SIEM solution, with a focus on its potential for deployment in small and medium-sized enterprises (SMEs). Additionally, this chapter explores possible avenues for improvement within HELK. It concludes with a basic comparison of HELK against other commercially available SIEM tools, providing valuable insights into the suitability of HELK for SMEs.

- **CHAPTER 8: Conclusions and Related Work**
  In the concluding chapter, Chapter 8, we draw our conclusions regarding the viability of deploying HELK and its utility as a compelling SIEM alternative compared to other commercial tools. Additionally, we delve into an examination of pertinent related works concerning HELK and Small and Medium-sized Enterprise (SME) Cybersecurity in a broader context.

CHAPTER 2: Cybersecurity Fundamentals

2.1 INTRODUCTION TO CYBERSECURITY

Cybersecurity represents a vast domain within the realm of Information Technology, focused on safeguarding critical systems and sensitive data from digital threats. It is grounded in the fundamental recognition that in our increasingly interconnected world, nothing remains impervious to potential cyberattacks. Whether it's devices, networks, applications, or various forms of data, all are susceptible to digital threats. Consequently, cybersecurity experts are continually engaged in the quest for innovative and ingenious methods to fortify data and infrastructure against the ever-present risk of cyberattacks.

The history of cybersecurity and the associated cybercrimes trace their origins back to the nascent stages of the Internet. In the 1970s and 1980s, the realm of computer security was primarily confined to academic circles. It was during this era that the Internet began to take shape, ushering in an era of heightened connectivity, which, in turn, saw the emergence of computer viruses and network intrusions. However, it's worth noting that the 1970s and 1980s were marked by a relative absence of significant computer threats. This can be attributed to the fact that both computers and the Internet were still in their developmental phases, and security vulnerabilities were relatively conspicuous and straightforward to identify. In most cases, the primary sources of threats stemmed from malicious insiders who gained unauthorized access to sensitive documents and files.

Even during the nascent phase of Information Technology, noteworthy incidents of cyber-attacks were already making headlines. One such prominent example was the "Morris Worm," an early manifestation of malicious software that rapidly propagated, infecting tens of thousands of computer systems. Remarkably, this constituted approximately 10% of all computers connected to the Internet during that era. As we transitioned into the latter half of the 1970s, established computer industry players like IBM began introducing commercial access control systems and computer security software products to address these emerging challenges.

Between September 1986 and June 1987, a group of German hackers performed the first documented case of cyber espionage.[2] The group hacked into American defence contractors, universities, and military bases networks and sold gathered information to the Soviet KGB

2.2 THE IMPORTANCE OF CYBERSECURITY TODAY

In the present era, there has been an exponential surge in the multitude of devices interconnected via the Internet. According to data sourced from Statista [3], it is anticipated that by the year 2030, the global count of Internet of Things (IoT) connected devices will ascend to 29.4 billion. These devices exhibit a diverse spectrum of operating systems and protocols, coupled with distinct hardware configurations, thereby rendering them susceptible to unique vulnerabilities.

Among the array of cyber threats targeting this diverse landscape of devices, Distributed Denial of Service (DDoS) attacks are particularly noteworthy. DDoS assaults involve a coordinated effort to overwhelm a server, website, or network with an excessive influx of traffic. A striking example transpired in 2016 when the infamous Mirai botnet[4] emerged as a formidable threat. Remarkably, this botnet comprised primarily of embedded IoT devices, with its peak strength surging to nearly 300,000 devices. Its malevolent power was harnessed to disrupt several prominent Internet services, impacting regions across Europe and the United States.

As highlighted earlier, the proliferation of IoT devices and sensors has ushered in a new era of cybersecurity challenges. Presently, the predominant active threat menacing IoT devices is a strain of backdoor-type malware attributed to the Mirai family, with a primary focus on Linux distributions. A closer examination of Kaspersky's 2021 Security Report on IoT threat statistics [5] illuminates that an astonishing 77.47% of attacks on their traps (honeypots) were executed through the Telnet protocol, an antiquated and, surprisingly, still vulnerable method.

In 2022, the average cost of a data breach was USD 4.35 million globally, and USD 9.44 million in the United States [6]. These costs include the expenses of discovering and responding to the breach, the cost of downtime and lost revenue, and the long-term reputational damage to a business and its brand.

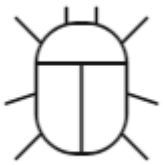
The Financial Sector, moreover, faces cyber-attacks constantly. According to VMware, the first half of 2020 saw a 238% increase in cyberattacks targeting financial institutions[7]. In one of the largest cyber heists that ever took place, the Central Bank of Bangladesh lost over $81 million [8] from its account held in the Federal Reserve Bank of New York. The attackers attempted to steal close to $1 billion, employing among other things a printer, whose main goal was to print out real-time transactions and was connected to the SWIFT software. After investigation, it was discovered that an employee unknowingly downloaded a malware program through phishing emails, which in turn gave the attackers access to the bank’s network. The trespassers managed to remain undetected for over a month before the incident, observing the bank's operations, and on Feb 4. 2016 they started to issue payments through the SWIFT network (totaling $951 million over 35 transfers).

Many organizations, in fact, are under the impression that they will be unaffected by cyber threats in the future. A study that took place in 2021 by the SANS Institute[9], however, sheds light on the fact that approximately 1 in 3 organizations stated that they had been affected by security incidents or intrusions in their protected environments within the past year.

In an increasingly interconnected world, Cybersecurity has gained paramount importance. The widespread use of cloud services to store sensitive data and personal information has escalated both inherent and residual risks. Our heavy reliance on technology shows no signs of slowing down, making data breaches and identity theft more prevalent than ever. Personal details like Social Security numbers and credit card data are now stored in cloud-based solutions, adding to the urgency of fortifying our digital defenses to safeguard against these evolving threats.

Potential security vulnerabilities are ever-increasing and even Governments are not oblivious to the danger posed by cybercriminals. One prime example is the EU's GDPR (General Data Protection Regulation), increasing the reputational damage of data breaches, by forcing all organizations to communicate and report on data breaches[10]. According to Accenture's cost of Cybercrime study, 43% of cyber-attacks are aimed at small businesses, but only 14% are prepared to defend themselves[11]. Furthermore, 64% of companies worldwide have experienced at least one form of cyber-attack. Every 39 seconds, there is a new attack somewhere on the web. Cybercrime increased six-fold during the COVID-19 pandemic.

2.3 COMMON CYBER THREATS



Malware

"Malware"[12]encompasses various types of malicious software, including worms, viruses, Trojans, and spyware, which are intended to gain unauthorized access or cause harm to computer systems. In recent times, malware attacks have evolved to become more "fileless," aiming to bypass traditional detection techniques like antivirus programs that primarily scan for malicious file attachments.

Ransomware

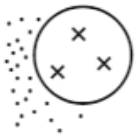
Ransomware is a form of malicious software designed to encrypt files, data, or computer systems, with the intention of extorting payment from victims. The attackers demand a ransom in exchange for decrypting the data or restoring access to the affected systems. In recent times, there has been a surge in ransomware attacks targeting state and local governments. These entities are often more vulnerable to breaches compared to larger organizations and face increased pressure to pay the ransom in order to regain access to critical applications and domains that are essential for serving the public.

Phishing / social engineering

Phishing is a type of social manipulation technique that deceives individuals into divulging their personal identifiable information (PII) or sensitive data. Phishing scams typically involve fraudulent emails or text messages that impersonate reputable organizations, requesting recipients to provide confidential details like credit card information or login credentials. The Federal Bureau of Investigation (FBI) has observed a significant increase in pandemic-related phishing activities, which can be attributed to the rise of remote work arrangements.

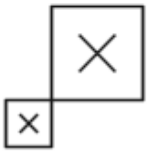
Insider threats

Insider threats encompass individuals such as current or former employees, business partners, contractors, or anyone who has had prior access to systems or networks. They pose a risk when they misuse their authorized access permissions. Unlike external threats, insider threats may go undetected by conventional security solutions like firewalls and intrusion detection systems, which primarily concentrate on external vulnerabilities.



Distributed denial-of-service (DDoS) attacks

A Distributed Denial of Service (DDoS) attack aims to disrupt the normal functioning of a server, website, or network by inundating it with a large volume of traffic, typically originating from multiple sources working in coordination. The objective of a DDoS attack is to overwhelm the targeted system, rendering it inaccessible to legitimate users.



Advanced persistent threats (APTs)

An Advanced Persistent Threat (APT) refers to the infiltration of a system by an individual or group with the objective of remaining undetected for a prolonged period. During an APT, the intruders intentionally avoid triggering defensive measures and maintain the integrity of networks and systems. This allows them to covertly monitor business operations, gather sensitive data, and carry out espionage activities. The Solar Winds incident in 2019, which targeted the systems of the United States government[13], serves as an illustrative example of an APT

2.4 MITRE ATT&CK FRAMEWORK AND CYBER KILL CHAIN

Understanding how attacks work is critical for defense. There have been many attempts throughout the years to come up with a common framework for information security teams to speak the same language. A threat-hunting tool requires an established framework, to act as a compass for the actions of an attacker.

One such framework is the Cyber Kill Chain, published in 2011 by Lockheed Martin, in one of the first attempts in explaining how attacks work [14]. The Cyber Kill Chain attempted to cover the 7 steps or tactics attackers perform during an attack, thus providing insight into these techniques and procedures which can prove invaluable for Security personnel. These are

- Reconnaissance (Research and selection of targets)
- Weaponization (Insertion of malware into a deliverable payload e.g. PDF)
- Delivery (Transmission of weapon to target e.g. email, USB)
- Exploitation (Trigger weapon’s code, thus attempting exploitation)

- Installation (Backdoor Installation onto a target system for Persistence)
- Command & Control (Comms between weapon and Outside Server)
- Actions on Objective (Exfil or destruction of data or any other objectives)

Another established framework is the MITRE ATT&CK released in 2015 which stands for Adversary Tactics, Techniques, & Common Knowledge. It is a globally accessible knowledge base of tactics and techniques based on real-world observations. The ATT&CK framework is the industry standard when it comes to understanding and communicating how attacks work. It builds upon the Cyber Kill Chain by expanding the attacker's goals to 14 distinct tactics[15]. Moreover, it covers the specific techniques and sub-techniques used by threat actors to achieve those goals. There are over 180 techniques and over 370 sub-techniques described.
Every possible action taken by adversaries described in the ATT&CK framework has a unique ID, a name, and a description explaining this specific action.

CHAPTER 3: Threat Hunting Defined

3.1 SECURITY OPERATIONS CENTERS

A Security Operations Centre (SOC) plays a pivotal role in safeguarding an organization from cyber threats by ensuring continuous monitoring of its network, prompt investigation of potential threats, and swift response to neutralize any identified risks. To effectively monitor network activities, SOCs heavily rely on data logs generated by various network devices and computer hosts within the organization. These logs are meticulously analyzed and assessed for any signs of suspicious or malicious activities.The central component of their defense mechanism is the Security Information and Event Management (SIEM) system, which provides comprehensive insights into security events across the network. Additionally, SOC efforts are reinforced by the incorporation of Endpoint Detection and Response (EDR) solutions, which enhance visibility and monitoring capabilities at the endpoint level. Intrusion Detection Systems (IDS) are also integrated to proactively identify and thwart potential attacks in real-time.The core objective of these integrated tools is to expedite the SOC's response time to potential threats. This response time encompasses two crucial scenarios: First, mitigating the damage caused by an attack that has already occurred but has been promptly reported and addressed. Second, proactively detecting and halting an ongoing attack before it can inflict any harm.

3.2 SIEMs DEFINITION

A Security Information and Event Management (SIEM) solution plays a crucial role in alleviating the workload of SOC analysts. By aggregating data from multiple sources and leveraging advanced data analytics, SIEM solutions efficiently identify and prioritize potential threats. Essentially, they offer a comprehensive, real-time view of network activities. Over the past decade, SIEM solutions have undergone significant evolution and refinement, solidifying their position as indispensable tools in the cybersecurity landscape.

What is unique about SIEM Solutions is that they combine Security Event Management (SEM) with Security Information Management (SIM). These give the solution the following capabilities:

- Threat Monitoring and Incident Response
- Log file aggregation from multiple sources
- Correlation capabilities (Rule-based, statistical)
- Alerting capabilities (e.g. via email)
- Dashboard functionality (e.g. graphs for pattern identification)
- Compliance functionality (e.g. reports generation for audit purposes)
- Retention capabilities (e.g. for compliance requirements and forensic investigations)
- Forensic analysis capabilities - search across logs on different nodes and time periods based on specific criteria

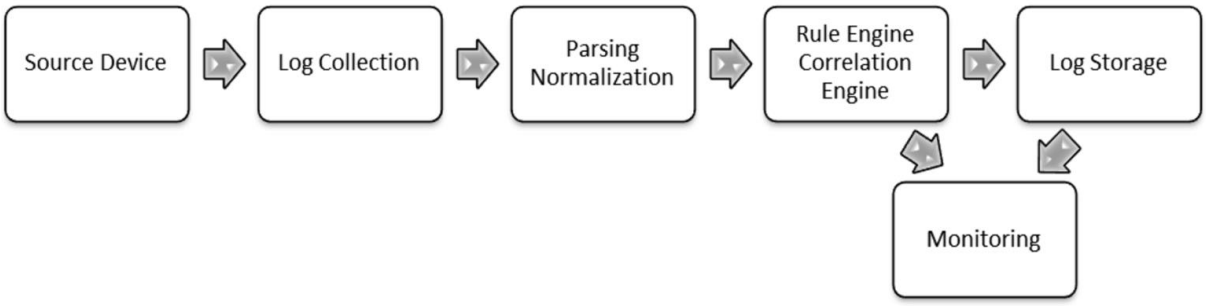


Figure 1: Flow of Logs and Data within a SIEM

3.3 HOW DO SIEMs FUNCTION?

A well-deployed and properly configured SIEM is a highly effective security tool. It operates by collecting vast quantities of logs and event data generated by various components in an organization's infrastructure, including host systems such as computers and servers, applications like antivirus solutions, and security devices like firewalls, IDS, and IPS.All of this information is consolidated within a shared platform. SIEM software plays a crucial role in identifying and categorizing this diverse data based on predefined rules. Categories can range from malware activity and failed logins to detecting malicious scans across a company's network. By collating and analyzing this information, SIEM provides valuable insights into potential security threats and suspicious activities, empowering organizations to proactively respond and strengthen their overall security posture.

The SIEM software identifies potentially threatening activities and generates alerts of varying priority levels, ranging from Low to High, based on the severity of the situation and predefined rules. The effective creation and configuration of these rules are crucial for generating genuine alerts that warrant further investigation by the end user. Otherwise, important alerts may get lost in the event log noise, hindering the ability to promptly address potential security threats.

To ensure optimal functionality, a SIEM relies heavily on well-crafted correlation rules. These rules serve as the backbone of the system, guiding it in identifying potential security anomalies or cyber threats. Keeping these rules up-to-date is crucial, given the ever-evolving nature of security vulnerabilities and attack patterns. By continuously updating or expanding the rules, the SIEM stays abreast of the latest threats, thereby strengthening its ability to detect and respond to emerging risks effectively. In essence, correlation rules play a pivotal role in empowering the SIEM to recognize sequences of events that may indicate security weaknesses or cyber-attacks, contributing significantly to the overall security posture of an organization.

A rule can be simply represented as
When **X** event is true, do action **Y** and notify ***username_1***
When **X** or **Y** events are true, notify ***username_1***

Examples of Correlation rules can be:

```
If User John has 20 failed log-ins within 20 minutes --> Generate Low Severity Alert
Note: This may well be genuine behaviour of a user forgetting their credentials)

If User John has 150 failed logins within 5 minutes -> Generate High Severity Alert
(This could indicate a brute-force attack in order to gain unauthorized access to a user's account)
```

3.4 BENEFITS OF USING A SIEM

SIEM solutions represent a potent means of identifying and responding to threats promptly, offering real-time reporting, and delivering enduring analytical capabilities for security logs and events. Consequently, they assume an indispensable role in protecting organizations and businesses across various scales. The advantages of adopting SIEM encompass the following:

Visibility

Most hosts that log security breaches do not have built-in capabilities for incident detection. These hosts can observe events and generate log entries for them, but they lack the ability to analyze said entries to identify signs of malicious activity. SIEM solutions can gather events across hosts, correlate them and then reconstruct the series of said events to determine the nature of the attack and whether or not it was successful.

In other words, while an N-IPS (network intrusion prevention system) might see one part of an attack and a computer's operating system might see another part, a SIEM can correlate and log data for all of these otherwise unrelated events. It is important to stress, however, that while SIEM tools have many capabilities and benefits, they cannot and should not replace enterprise security controls used to detect attacks. (e.g firewalls, Antivirus Solutions, Intrusion Prevention Systems). A SIEM tool on its own is practically useless unless used in tandem with other solutions, for the simple reason that it cannot monitor raw security events as they happen throughout an enterprise in real-time. The SIEM needs to be fed with log data, captured by other software and appliances.

Another method to enhance the capabilities of a SIEM (Security Information and Event Management) solution involves incorporating valuable threat intelligence data from reputable external sources and security providers, including industry leaders like IBM and Microsoft, among others. For example, if the SIEM tool detects any activity involving known malicious hosts or domains, it can terminate the connection with them before they can infect an organization, thus surpassing detection and entering the realm of prevention.

Streamline compliance Reporting

Many organizations deploy SIEM Solutions only for the fact that they can streamline enterprise compliance reporting efforts through a centralized logging solution. If for example a business or company needs its host to have their logged security events included in reporting regularly,

they can do so by transferring said log data to a SIEM Server. The SIEM Server can then gather log data from multiple hosts and generate a report that addresses all of the relevant logged security events among these hosts.
An organization without a SIEM Solution is unlikely to have robust centralized logging capabilities that can create rich, customizable reports, such as those necessary for most compliance reporting efforts. It allows the organization to generate individual reports for each host or manually retrieve data from a sample of hosts periodically and reassemble said data in a centralized point in order to generate a single report.
The latter is extremely difficult, for no other reason but the fact that different operating systems, apps, and other software are likely to log their security events in various proprietary ways, making correlation quite challenging. Conversion of all of this info into a single format might require extensive code development on the part of the organization.
SIEM tools are also very useful due to the fact that they have built-in support for most common compliance efforts. Their reporting capabilities are compliant with the requirements and standards set by the Health-Insurance-Portability and Accountability Act (HIPAA) or the Payment Card Industry Data Security Standard (PCI DSS). Such capabilities allow organizations to save considerable time and effort when striving to meet their security compliance reporting requirements, especially if they are subject to more than one such compliance initiative.

Incident Handling Efficiency improvement

One of the most important benefits of using a SIEM Solution is that it significantly increases the efficiency of incident handling, which in turn saves time and resources for security analysts and incident handlers. Efficient incident handling ultimately can speed incident containment efforts, thus reducing the amount of damage that a security breach can cause.
The above can be achieved primarily by providing incident handlers with a single interface with which to view all security log data from multiple hosts. Examples of how this can expedite incident handling include:

- Enabling incident handlers to quickly identify the route of an attack through the enterprise.
- Enabling rapid identification of all of the hosts affected by the attack.
- Provide automated mechanisms that can stop attacks still in progress, and then contain the hosts that have been compromised.

All the aforementioned capabilities enable organizations to get a Big-Picture view of their security incidents throughout the enterprise. By centralizing log data from enterprise security controls and solutions, host operating systems, applications, and other software components, SIEM tools can analyze large volumes of security-related log data, in order to identify threats, incoming attacks, and security compromises. This enables the SIEM solution to identify malicious activity in a way that no other single host or piece of software could, due to their lack of true enterprise-wide visibility.

3.5 THE COST OF SIEM SOLUTIONS

The upfront investment required for a SIEM solution from renowned vendors like IBM or Microsoft can often reach hundreds of thousands of dollars. Additionally, there are other associated costs, such as hiring personnel to oversee and monitor the SIEM implementation, annual support fees, and expenses related to acquiring software or agents to collect crucial data. For many small and medium-sized businesses, these costs may appear daunting, particularly when they seek to protect their valuable data and operational capabilities amidst an increasingly network-centric financial landscape.

In most cases, the deployment of a SIEM involves more than just the software itself; it encompasses a comprehensive approach. A typical scenario includes the following components:

1. SIEM Software: The core software that serves as the central hub for log collection, analysis, and threat detection.

2. Deployment Consulting Support: Specialized Security Engineers or Architects are engaged to set up the network and hardware infrastructure of an organization, ensuring that the SIEM receives the appropriate data feeds. These consultants also ensure the continuous and stable operation of all components involved in the SIEM deployment.

3. Training: All security and IT staff of the organization undergo training to understand the inner workings and operation of the SIEM software. This enables them to effectively leverage its capabilities.

4. SOC Analyst Personnel: Dedicated SOC analysts are responsible for continuously monitoring the SIEM solution for potential incoming threats on a 24-hour basis. Their vigilance ensures swift response to any security incidents.

5. Database Administrators: Skilled professionals manage the databases used to store and process vast amounts of log data generated by the SIEM.

6. Hardware: The hardware infrastructure, tailored to the organization's size and SIEM configuration, is designed to meet performance requirements. While off-the-shelf parts can be used, specialized hardware may be required in larger organizations.

7. Intel Feeds: Continuous data streams, such as Open-Source Intelligence (OSINT) and inputs from third-party vendors like IBM, Palo Alto Networks, or Exchange, provide the latest information on emerging threats. These feeds equip security teams with Indicators of Compromise (IoCs), including malicious URLs, IPs, suspicious emails, and malware hashes, to enhance threat detection capabilities.

8. Infrastructure and Solutions: Servers, databases, switches, firewalls, and other network infrastructure components form an essential part of the SIEM ecosystem, ensuring seamless data flow and security measures.

By considering these diverse aspects, organizations can deploy a comprehensive SIEM solution that effectively addresses their security needs while optimizing cost-effectiveness and threat detection capabilities.

Trustwave's white paper on SIEM Budgeting [16] reveals significant cost implications associated with deploying a SIEM solution on-premise. Large enterprises can face an annual

expense exceeding $1 million USD, while medium-sized businesses may incur costs of approximately $600 thousand USD. For small enterprises, the expenditure can reach up to $200 thousand USD. These figures underscore the considerable financial investment required to implement a SIEM solution, posing challenges for organizations of varying sizes.

CHAPTER 4: The HELK Platform

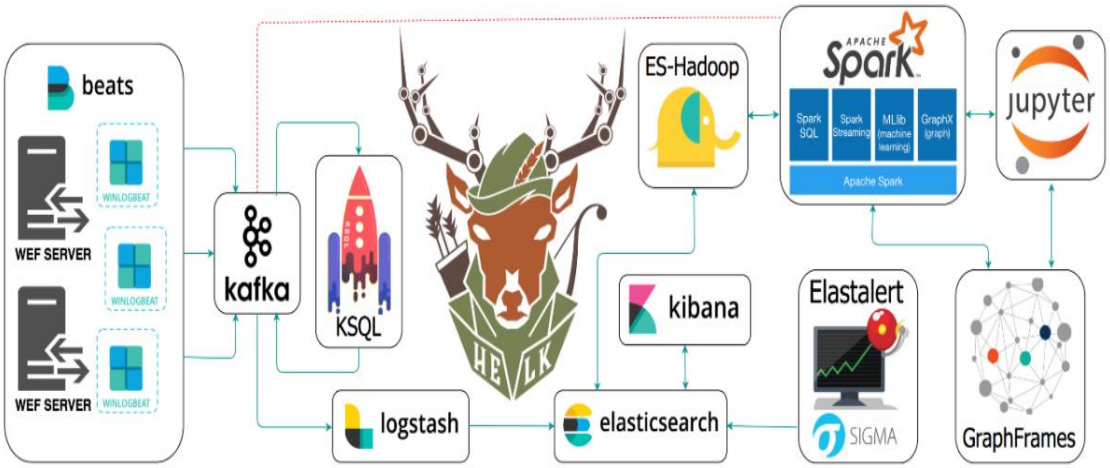


Figure 2: The HELK platform and the different parts that make up its structure

The Hunting-ELK or simply HELK is one of the first truly open-source hunting platforms with advanced analytics capabilities such as:

- SQL declarative language
- Graphing
- Structured Streaming
- Machine Learning (via Jupyter notebooks and Apache Spark)

This open-source platform, visually explained in Figure 2 above, was developed primarily for research, but due to its flexible and scalable design and core components, it can be deployed in both small and large environments with the right configurations and modular infrastructure[17]. HELK is based primarily on the Elastic Stack, which will be explained in detail below. In essence, it is a log analytics engine used to gather log data from a multitude of sources (like servers, apps, and services) and centralize them for further processing or investigation. It is used for a variety of purposes like troubleshooting, monitoring, security, and auditing.

4.1 HELK CORE COMPONENTS

4.1.1 ELASTICSEARCH

Elasticsearch is a distributed, open-source search and analytics engine for all types of data, including but not limited to textual, numerical, geospatial, structured, and unstructured[18].

Elasticsearch is built on Apache Lucene, a free and open-source search engine software library used as a standard foundation for non-research search applications.
Known for its simple REST APIs **(REST API (also known as RESTful API)** is an application programming interface (API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services), distributed nature, speed, and scalability.
Elasticsearch is the central component of the Elastic Stack, a set of open-source tools used for data ingestion, enrichment, storage, analysis, and visualization. Known commonly as the ELK Stack (after Elasticsearch, Logstash, and Kibana), the Elastic Stack includes a collection of lightweight shipping agents known as Beats, used to send data to Elasticsearch.

The benefits of Elasticsearch

The speed and scalability of Elasticsearch, as well as its ability to index many types of data, mean that it can be used for a multitude of use cases:

- Website/Domain Searches
- Application Searches
- Enterprise searches
- Logging and log analytics
- Infrastructure metrics and container monitoring
- Application Performance Monitoring
- Geospatial data analysis and visualization
- Security analytics

How does Elasticsearch work?

Raw data can be fed into Elasticsearch from a variety of sources, including logs, system metrics as well as web applications. The process by which this raw data is parsed, normalized, and enriched before being indexed in Elasticsearch is called Data ingestion.
Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data. From Kibana, users can then create powerful and rich visualizations of their data, share dashboards and manage the Elastic Stack.
Figure 3 below shows us a basic outline of the relationships between the different ELK components.

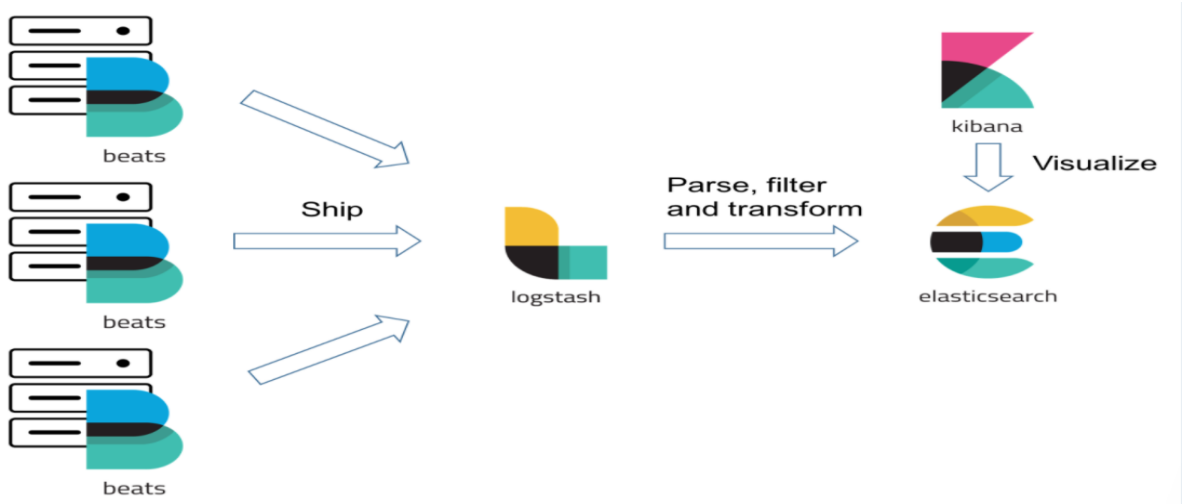


Figure 3:*Visual Representation of the* ***ELK*** *stack and its components*

4.1.2 KIBANA

Kibana[19] is a frontend application that sits on top of the Elastic Stack, providing search and data visualization capabilities for data indexed in Elasticsearch. It also acts as the user interface for monitoring, managing, and securing an Elastic Stack cluster as well as the centralized hub for built-in solutions developed on the Elastic Stack.

It acts as a data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as:

- Histograms
- Line Graphs
- Pie Charts
- Heat Maps
- Built-in geospatial support

The benefits of Kibana:

1. ***Interactive Charts***

Kibana offers intuitive charts and reports that can be used to interactively navigate through large amounts of log data. Users can dynamically drag time windows, zoom in and out of specific data subsets as well as drill down on reports to extract actionable insights from their data.

2. ***Mapping Support***

Kibana provides users with basic visualization tools, such as line graphs, histograms, and pie charts, together with the option of letting them design their own data images. Because of this, the graphical presentation of data is optimally in tune with the users' needs and preferences. This is accomplished through Vega Grammar, which is a visualization language that Kibana can fully integrate with.

3. *Easily accessible Dashboards*

Kibana visualizations and dashboards can be easily shared with other people by simply embedding them into web pages or sending the link to intended recipients. Users can also show their dashboard to more people while retaining full control over what information can be viewed by them, hence securing sensitive information against leakage. Data can also be exported in PDF or CSV format files.

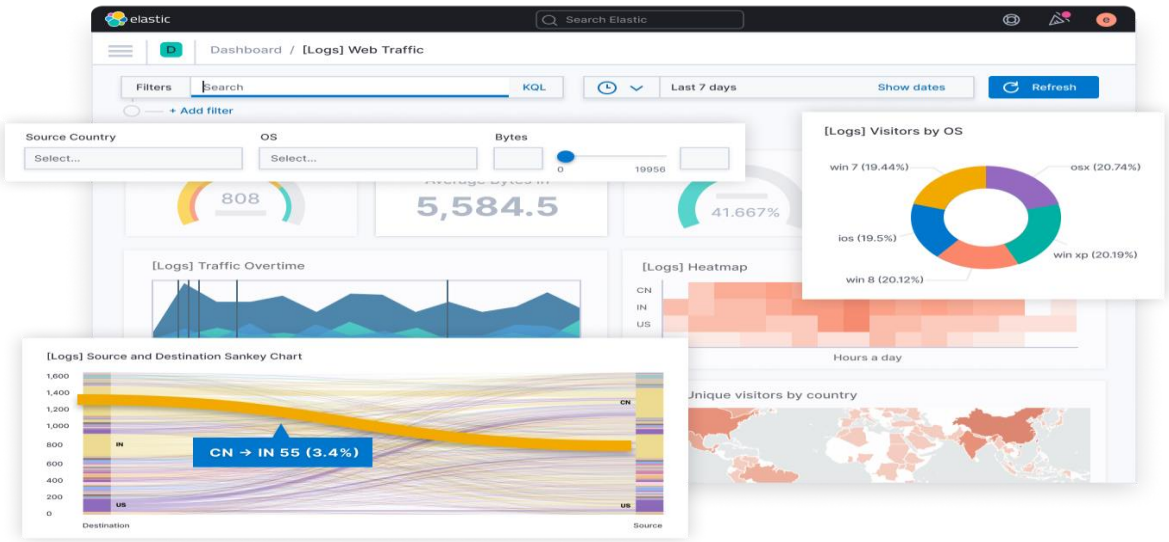


Figure 4: *Visual Representation of the* ***Kibana*** *UI and its Dashboards*

4.1.3 LOGSTASH

Logstash[20] is a lightweight, open-source, server-side data processing pipeline that allows users to collect data from a variety of sources, transform it on the fly and send it to their desired destination. It is most often used as a data pipeline for Elasticsearch. Because of its tight integration with Elasticsearch, powerful log processing capabilities, and over 200 pre-build open source plug-ins used for easy indexing of data, Logstash is a popular choice for loading data into Elasticsearch.

The benefits of Logstash:

1. *Easily Load Unstructured Data*

Logstash allows users to easily ingest unstructured data from a variety of data sources including system logs, domain logs as well as application server logs.

2. *Pre-Built Filters*

Logstash offers pre-built filters, allowing users to readily transform common data types, index them in Elasticsearch and then start querying without having to build custom data transformation policies.

3. *Flexible Plug-in Architecture*

With over 200 plugins already available on GitHub, there is a wide variety of options when users want to customize their data pipelines. Moreover, users can create new ones that suit their needs.
A basic outline of the Logstash capabilities with different forms of input data can be seen in Figure 5 below.

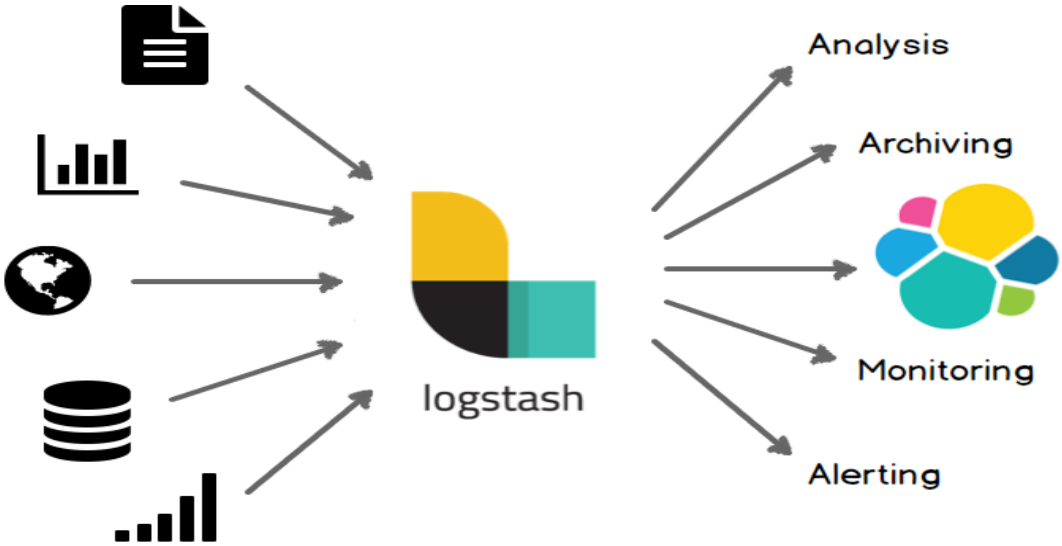


Figure 5: *Visual Representation of* ***Logstash***

4.1.4 BEATS

Beats [21] are open-source data shippers. They are easy-to-install agents that ship data from endpoint computers (or hosts), network devices, and applications to the HELK appliance. There are many different types of beats, each one with a different purpose. For the purposes of this scenario, we will make use of Winlogbeat.
Winlogbeat reads from one or more event logs using Windows APIs, filters the events based on user-configured criteria, then sends the event data to the configured outputs (Elasticsearch or Logstash). Winlogbeat watches the event logs so that new event data is sent in a timely manner. The read position for each event log is persisted to disk to allow Winlogbeat to resume after restarts.

Winlogbeat can capture event data from any event logs running on your system. For example, you can capture events such as:

- Application events
- Hardware events
- Security events
- System events

4.1.5 KAFKA

Apache Kafka [22] is a distributed data store optimized for ingesting and processing streaming data in real-time. Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records in simultaneously. The Apache Kafka software constitutes the most common solution for deployment in coordination with the ELK Stack. In most cases, the Kafka broker is deployed between the shipper and the indexer, serving as an entry point for the data being gathered. Basically, Kafka is the software that sits between Beats and Logstash.

4.1.6 ELASTALERT

ElastAlert [23] is a simple framework for alerting on anomalies, spikes, or other patterns of interest from data stored in Elasticsearch. It functions by combining Elasticsearch with two types of components, rule types and alerts. Elasticsearch is periodically queried and the data is passed to the rule type, which determines when a match is found. When a match is indeed found, it is assigned to one or more alerts, which take the appropriate action based on the match that was found. Therefore, for the purposes of this scenario, the rule defines what behavior is considered malicious or suspicious and in turn, the alert provides us with the information we need in order to investigate the incident that is tied to this alert.

4.1.7 OTHER COMPONENTS

- **KSQL**, a streaming SQL engine for Kafka which works quite differently from an SQL database. While most databases execute on-demand lookups of data, KSQL does constant stream processing, meaning it runs continuous queries as new data is fed through the Kafka broker.
- **NGINX** acts as a web server that can also be used as a Load Balancer, a reverse proxy, or an HTTP cache. It comes installed in a Docker container during the HELK installation.
- **Docker**, lastly, is a software platform that allows us to package applications and dependencies into a single unit, called a container. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries, and settings. Docker is an essential part of the HELK appliance since every different service runs inside its own container.

4.2 HELK INSTALLATION

4.2.1 SYSTEM REQUIREMENTS

The installation of HELK has the following minimum system requirements:

- **Ubuntu 16 or 18.04 or CentOS:** In this instance, I used Ubuntu 18.04 as it is the preferred version as mentioned by the creators of HELK.

- **Docker:** HELK uses the official Docker Community Edition (CE) bash script to install Docker during the installation process if no Docker instance is found.

- **CPU:** In terms of processing power, a minimum of 4 64-bit cores (logical/physical) are recommended. Elastic actually requires CPUs that have the SSE4.2 instruction set. In this case, I assigned 4 Cores to the Ubuntu VM (2 Physical and 2 Logical) from an Intel Core i9-9900K.

- **Storage:** Regarding Storage, 20 GB is the bare minimum if HELK is to be used for testing. However, in this instance, we assigned 150 GB of hard drive space, just to be sure.

- **RAM:** When it comes to system memory, it is important to consider which HELK version we want to install, which will determine how much RAM we need. These are:

  1: KAFKA+KSQL+ELK+NGINX (requires 5GB)
  2: KAFKA+KSQL+ELK+NGINX+ELASTALERT (requires 5GB)
  3: KAFKA+KSQL+ELK+NGINX+SPARK+JUPYTER (requires 7GB)
  4: KAFKA+KSQL+ELK+NGINX+SPARK+JUPYTE +ELASTALERT (requires 8GB)

- **Network:** Lastly, as far as networking is concerned, a Valid IPv4 connection with Internet access is all that is needed. IPv6 has not been tested at the time of writing. Since we are working with Virtual Machines NAT or Bridge will work.

4.2.2 INSTALLATION

In order to ensure a smooth installation of HELK, it is important to use the "*sudo apt install git*" command, which will allow us to download the HELK from its GitHub repository. The installation of Git is clearly shown in Figure 6.

```
dskan@ubuntu:~$ sudo apt install git
[sudo] password for dskan:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
  gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
  grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
  libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1
  libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
  libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libedataserverui-1.2-2 libeot0
  libepubgen-0.1-1 libetonyek-0.1-1 libexiv2-14 libfreerdp-client2-2
  libfreerdp2-2 libgc1c2 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6
  libgpod-common libgpod4 liblangtag-common liblangtag1 liblirc-client0
  liblua5.3-0 libmediaart-2.0-0 libmspub-0.1-1 libodfgen-0.1-1 libqqwing2v5
  libraw16 librevenge-0.0-0 libsgutils2-2 libssh-4 libsuitesparseconfig5
  libvncclient1 libwinpr2-2 libxapian30 libxmlsec1-nss
  linux-headers-5.4.0-84-generic linux-hwe-5.4-headers-5.4.0-84
  linux-image-5.4.0-84-generic linux-modules-5.4.0-84-generic
  linux-modules-extra-5.4.0-84-generic lp-solve media-player-info python3-mako
  python3-markupsafe syslinux syslinux-common syslinux-legacy
  usb-creator-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 5 not upgraded.
Need to get 4,751 kB of archives.
After this operation, 34.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.17025-1 [22.8 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all 1:2.17.1-1ubuntu0.13 [805 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.13 [3,923 kB]
```

Figure 6: *Git installation*

Now, it is time to clone the HELK from its GitHub repository via the use of *git* as can be seen in Figure 7.

```
dskan@ubuntu:~$ git clone https://github.com/Cyb3rWard0g/HELK
Cloning into 'HELK'...
remote: Enumerating objects: 10109, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 10109 (delta 23), reused 37 (delta 16), pack-reused 10060
Receiving objects: 100% (10109/10109), 852.60 MiB | 7.18 MiB/s, done.
Resolving deltas: 100% (6948/6948), done.
```

Figure 7*: HELK Cloning from Github*

Then, we need to change our current directory to the HELK directory, named *HELK/docker*, and use the "*sudo apt install net-tools"* command. The Net-tools package is a collection of programs for controlling the network subsystem of the Linux kernel. The Installation is shown in Figure 8.

```
dskan@ubuntu:~/HELK$ ls
configs  docker  docs  LICENSE  PULL_REQUEST_TEMPLATE.md  README.md  resources  scripts
dskan@ubuntu:~/HELK$ cd docker
dskan@ubuntu:~/HELK/docker$ sudo apt install net-tools
[sudo] password for dskan:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Figure 8: Net-tools install

Lastly, after having successfully cloned HELK from its repository, we need to install it. While we were in the *HELK/docker* directory, we used the *helk_install.sh* file, which will allow us to begin the installation as is shown in Figure 9.

```
dskan@ubuntu:~$ cd HELK/docker
dskan@ubuntu:~/HELK/docker$ sudo ./helk_install.sh
[sudo] password for dskan:

**************************************************
**           HELK - THE HUNTING ELK             **
**                                              **
** Author: Roberto Rodriguez (@Cyb3rWard0g)     **
** HELK build version: v0.1.9-alpha10082020 **
** HELK ELK version: 7.6.2      **
** License: GPL-3.0                            **
**************************************************

[HELK-INSTALLATION-INFO] HELK hosted on a Linux box
[HELK-INSTALLATION-INFO] Available Memory: 8822 MBs
[HELK-INSTALLATION-INFO] You're using ubuntu version bionic

*****************************************************
*       HELK - Docker Compose Build Choices         *
*****************************************************

1. KAFKA + KSQL + ELK + NGINX
2. KAFKA + KSQL + ELK + NGINX + ELASTALERT
3. KAFKA + KSQL + ELK + NGINX + SPARK + JUPYTER
4. KAFKA + KSQL + ELK + NGINX + SPARK + JUPYTER + ELASTALERT

Enter build choice [ 1 - 4]: 2
```

Figure 9: HELK Installation Part 1

Beginning with the installation above, we can see that HELK can recognize the available RAM and our Operating system. Here, we are presented with the 4 options mentioned a while back. For the purposes of this thesis, where we want to use HELK purely as a SIEM, number 2 is our choice.

As we can see below, we are asked to provide our desired IP Address that will host our SIEM. We will also use the default username and password for Kibana, namely ***helk*** and ***hunting*** respectively. As Figure 10 shows, at this stage the installer is checking whether **Docker** is installed in our system, which isn't.

```
[HELK-INSTALLATION-INFO] You're using ubuntu version bionic

*************************************************
*      HELK - Docker Compose Build Choices      *
*************************************************

1. KAFKA + KSQL + ELK + NGINX
2. KAFKA + KSQL + ELK + NGINX + ELASTALERT
3. KAFKA + KSQL + ELK + NGINX + SPARK + JUPYTER
[HELK-INSTALLATION-INFO] Set HELK IP. Default value is your current IP: 192.168.196.132

[HELK-INSTALLATION-INFO] HELK IP set to 192.168.196.132
[HELK-INSTALLATION-INFO] Please make sure to create a custom Kibana password and store it securely for future use.
[HELK-INSTALLATION-INFO] Set HELK Kibana UI Password: hunting
[HELK-INSTALLATION-INFO] Verify HELK Kibana UI Password: hunting
[HELK-INSTALLATION-INFO] Installing htpasswd..
[HELK-INSTALLATION-INFO] Installing curl before installing docker..
[HELK-INSTALLATION-INFO] Installing docker via convenience script..
[HELK-INSTALLATION-INFO] Assesing if Docker is running..
[HELK-INSTALLATION-INFO] Docker is running
[HELK-INSTALLATION-INFO] Making sure you assigned enough disk space to the current Docker base directory
[HELK-INSTALLATION-INFO] Available Docker Disk: 130 GBs
[HELK-INSTALLATION-INFO] Installing docker-compose..
[HELK-INSTALLATION-INFO] Checking local vm.max_map_count variable and setting it to 4120294
[HELK-INSTALLATION-INFO] Setting local vm.swappiness variable to 25
[HELK-INSTALLATION-INFO] Building & running HELK from helk-kibana-analysis-alert-basic.yml file..
[HELK-INSTALLATION-INFO] Waiting for some services to be up .....


******************************************************************************
** [HELK-INSTALLATION-INFO] HELK WAS INSTALLED SUCCESSFULLY                  **
** [HELK-INSTALLATION-INFO] USE THE FOLLOWING SETTINGS TO INTERACT WITH THE HELK **
******************************************************************************

HELK KIBANA URL: https://192.168.196.132
HELK KIBANA USER: helk
HELK KIBANA PASSWORD: hunting
HELK ZOOKEEPER: 192.168.196.132:2181
HELK KSQL SERVER: 192.168.196.132:8088

IT IS HUNTING SEASON!!!!!
```

Figure 10: HELK Installation Part 2: Docker, Password Setup, IP

The HELK installation displays only the essential details and procedures running to our console while it is taking place. If we want to actually monitor what is happening in the background while our installation is taking place, we need to open a second terminal, and execute the tail -*f /var/log* command on the installation log file. Below in Figure 11 we can see the results of this command, which lists the installed **HELK services** from the installation Log File.

```
dskan@ubuntu:~/HELK/docker$ tail -f /var/log/helk-install.log
Status: Downloaded newer image for confluentinc/ksqldb-cli:latest
 Creating helk-elasticsearch ... done
 Creating helk-kibana        ... done
 Creating helk-nginx         ... done
 Creating helk-logstash      ... done
 Creating helk-zookeeper     ... done
 Creating helk-elastalert    ... done
 Creating helk-kafka-broker  ... done
 Creating helk-ksql-server   ... done
 Creating helk-ksql-cli      ... done
```

Figure 11: HELK Installation Log Activity

4.2.3 HELK SERVICES AND CONTAINERS

Now that our installation of the HELK is complete, we can start to investigate its inner workings in order to comprehend its structure by listing the services and containers that have been installed. To do so, in Figure 12, we executed the command "*sudo docker ps | less –S"* which allows us to list all of the services that are running through Dockers:

```
CONTAINER ID   IMAGE                                                 COMMAND                  CREATED         STATUS         PORTS
d48bdebfa004   confluentinc/ksqldb-cli:latest                        "/bin/sh"                6 minutes ago   Up 6 minutes
b5507e55dd34   confluentinc/ksqldb-server:latest                     "/usr/bin/docker/run"    6 minutes ago   Up 6 minutes   0.0.0.0:8088->8088/tcp, :::8088->8088/tcp
80a3316cf518   otrf/helk-kafka-broker:2.4.0                          "./kafka-entrypoint…"    6 minutes ago   Up 6 minutes   0.0.0.0:9092->9092/tcp, :::9092->9092/tcp
9feacd24d7a7   otrf/helk-elastalert:latest                           "./elastalert-entryp…"   6 minutes ago   Up 6 minutes
cb9514f85084   otrf/helk-zookeeper:2.4.0                             "./zookeeper-entrypo…"   6 minutes ago   Up 6 minutes   2181/tcp, 2888/tcp, 3888/tcp
b723b642e357   otrf/helk-logstash:7.6.2.1                            "/usr/share/logstash…"   6 minutes ago   Up 6 minutes   0.0.0.0:3515->3515/tcp, :::3515->3515/tcp, 0.0.0.0:5044->5044/tcp, :::5044->504
1bae8655a80c   otrf/helk-nginx:0.3.0                                 "/opt/helk/scripts/n…"   6 minutes ago   Up 6 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
090ee120ac0a   docker.elastic.co/kibana/kibana:7.6.2                 "/usr/share/kibana/s…"   6 minutes ago   Up 6 minutes   5601/tcp
419c7243d732   docker.elastic.co/elasticsearch/elasticsearch:7.6.2   "/usr/share/elastics…"   6 minutes ago   Up 6 minutes   9200/tcp, 9300/tcp
```

Figure 12: List of HELK Containers

```
CONTAINER ID   NAME                CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O         PIDS
d48bdebfa004   helk-ksql-cli       0.00%     2.277MiB / 10.25GiB   0.02%     8.03kB / 0B       0B / 0B           1
b5507e55dd34   helk-ksql-server    0.30%     401.9MiB / 10.25GiB   3.83%     457kB / 375kB     36.9kB / 2.67MB   39
80a3316cf518   helk-kafka-broker   0.53%     367.4MiB / 10.25GiB   3.50%     1.66MB / 1.63MB   516kB / 1.44MB    74
9feacd24d7a7   helk-elastalert     16.20%    80.42MiB / 10.25GiB   0.77%     6.65MB / 8.88MB   889kB / 1.15MB    12
cb9514f85084   helk-zookeeper      0.05%     84.33MiB / 10.25GiB   0.80%     157kB / 115kB     0B / 504kB        48
b723b642e357   helk-logstash       6.70%     1.138GiB / 10.25GiB   11.11%    1.02MB / 72.8MB   148MB / 504kB     106
1bae8655a80c   helk-nginx          0.00%     7.148MiB / 10.25GiB   0.07%     12.4kB / 1.39kB   2.11MB / 0B       4
090ee120ac0a   helk-kibana         0.71%     430.7MiB / 10.25GiB   4.10%     810kB / 2.07MB    162MB / 4.1kB     13
419c7243d732   helk-elasticsearch  37.42%    3.507GiB / 10.25GiB   34.22%    82.8MB / 7.68MB   209MB / 131MB     76
```

Figure 13: List of HELK Container Resources

Above in Figure 13, we can see the list of containers that come with the HELK. By running the *bash* command with the "NAME" field of a container we can investigate its contents.

As an example, we first use the command *"sudo docker exec –it helk-kibana cd bash"* to look inside the Kibana container as shown in Figure 14.

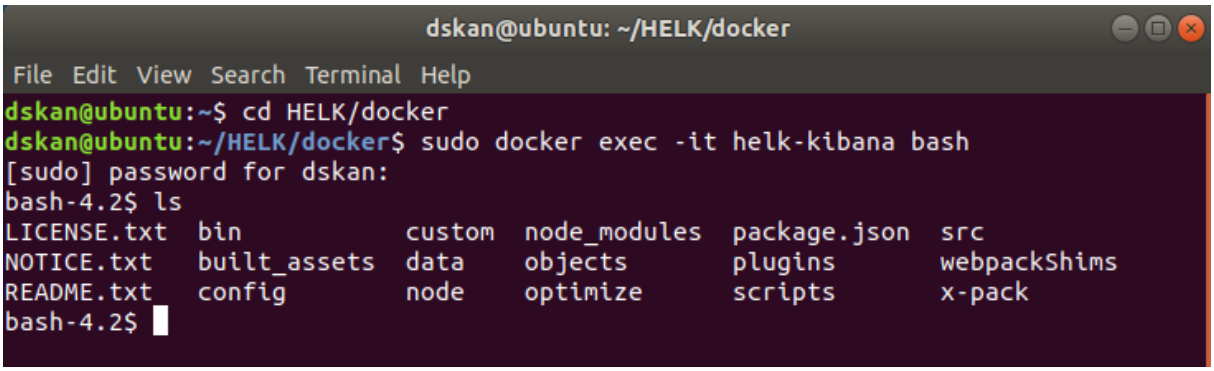


Figure 14: Inside the Kibana Container

In the same manner, we can display the Logstash Container as seen in Figure 15.

```
dskan@ubuntu:~/HELK/docker$ sudo docker exec -it helk-logstash bash
[sudo] password for dskan:
bash-4.2$ hostname
2be7b0c5728c
bash-4.2$ ls
bin                                lib                        output_templates
config                             LICENSE.txt                pipeline
CONTRIBUTORS                       logs                       plugins
cti                                logstash-core              scripts
data                               logstash-core-plugin-api   tools
Gemfile                            modules                    vendor
Gemfile.lock                       mordor_pipeline            x-pack
helk-plugins-updated-timestamp.txt NOTICE.TXT
bash-4.2$
```

Figure 15: Inside the Logstash Container

In Figure 16, we visited the *pipeline* folder, for example, and we can see a list of multiple configuration files for Logstash.

```
bash-4.2$ ls
0002-kafka-input.conf                                  1536-winevent-silkservice-filter.conf
0003-attack-input.conf                                 1542-winevent-process-ids-conversions-filter.conf
0004-beats-input.conf                                  1543-winevent-user-ids-conversions-filter.conf
0005-nxlog-winevent-syslog-tcp-input.conf              1545-winevent-security-conversions-filter.conf
0006-kafka-zeek-input.conf                             1590-winevent-rename-catchall-general-filter.conf
0011-syslog-tcp-input.conf                             1590-winevent-rename-catchall-processes-filter.conf
0011-syslog-udp-input.conf                             1590-winevent-rename-catchall-process-guids-filter.conf
0098-all-filter.conf                                   1590-winevent-rename-catchall-process-ids-filter.conf
0099-all-fingerprint-hash-filter.conf                  1592-winevent-conversions-catchall-process-ids-filter.conf
0301-nxlog-winevent-to-json-filter.conf                1593-winevent-process-path-split-to-name-filter.conf
1010-winevent-winlogbeats-filter.conf                  1594-winevent-cleanup-catchall-guids-filter.conf
1050-nxlog-winevent-to-winlogbeats-merge-filter.conf   2511-winevent-powershell-filter.conf
1051-nxlog-winevent-winevent-filter.conf               2512-winevent-security-schtasks-filter.conf
1090-helk-ecs_to_ossem-filter.conf                     3101-zeek_corelight-all-filter.conf
1216-attack-filter.conf                                8012-dst-ip-cleanups-filter.conf
1500-winevent-cleanup-no-dashes-only-values-filter.conf 8013-src-ip-cleanups-filter.conf
1521-winevent-conversions-ip-conversions-basic-filter.conf 8014-dst-nat-ip-cleanups-filter.conf
1522-winevent-cleanup-lowercasing-windows-filter.conf  8015-src-nat-ip-cleanups-filter.conf
1524-winevent-logon-ids-conversions-filter.conf         8112-dst-ip-filter.conf
1531-winevent-sysmon-filter.conf                       8113-src-ip-filter.conf
1532-winevent-security-filter.conf                     8114-dst-nat-ip-filter.conf
1533-winevent-system-filter.conf                       8115-src-nat-ip-filter.conf
1534-winevent-application-filter.conf                   8211-winevent-hostname-cleanups-filter.conf
1535-winevent-wmiactivity-filter.conf                   8251-helk-domains-and-hostnames-enrichments_and_additions-filter.conf
```

Figure 16: Inside the Logstash Container. 2

4.2.4 HELK RULES AND ELASTALERT

The effectiveness of HELK as a SIEM, rests largely on the rules and detections based on which it triggers the appropriate alerts. As explored in Chapter 2 correlation rules tell a SIEM what the sequence of events was which in turn could constitute a possible security risk. Elastalert periodically queries data from Elasticsearch in order to detect anomalies and generate alerts, if a rule is triggered.

HELK comes preloaded with a number of rules which we can modify or add upon. Firstly, though, we need to locate the directory where this ruleset is located in order to inspect the rules that constitute its ruleset.

```
root@ubuntu:~/HELK/docker# sudo docker exec -it helk-elastalert bash
elastalertuser@9feacd24d7a7:~$ cd /opt/sigma/rules
elastalertuser@9feacd24d7a7:/opt/sigma/rules$ ls -lh
total 40K
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 application
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 apt
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 cloud
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 compliance
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 generic
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 linux
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 network
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 proxy
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 web
drwxr-xr-x 1 elastalertuser elastalertuser 4.0K Oct  8  2020 windows
```

Figure 17: Elastalert Rule Categories

Above, in Figure 17, we can see the main rule categories present in Elastalert. They represent the main types of security vulnerabilities an attacker can exploit.

While below in Figure 18, we can see an example of a typical rule related to Windows-based vulnerabilities and more specifically related to Windows Powershell commands.

```
elastalertuser@9feacd24d7a7:/opt/sigma/rules/windows/powershell$ cat powershell_suspicious_download.yml
title: Suspicious PowerShell Download
id: 65531a81-a694-4e31-ae04-f8ba5bc33759
status: experimental
description: Detects suspicious PowerShell download command
tags:
    - attack.execution
    - attack.t1059.001
    - attack.t1086  #an old one
author: Florian Roth
date: 2017/03/05
logsource:
    product: windows
    service: powershell
detection:
    downloadfile:
        Message|contains|all:
            - 'System.Net.WebClient'
            - '.DownloadFile('
    downloadstring:
        Message|contains|all:
            - 'System.Net.WebClient'
            - '.DownloadString('
    condition: downloadfile or downloadstring
falsepositives:
    - PowerShell scripts that download content from the Internet
level: medium
```

Figure 18: Windows Suspicious Powershell Download Rule

We can clearly see in the. yml rule file pictured above, that this specific rule has a unique ID related to the MITRE ATT&CK Framework. T1059.001 represents the *Command and Scripting Interpreter: Powershell* Technique [24]. Adversaries can abuse PowerShell, a powerful interactive Windows command-line interface, in order to perform a number of malicious activities like information gathering and code execution. The rule pictured above

addresses PowerShell commands which are used to download and run executable files from the Internet.

As previously stated, Elastalert works in tandem with Elasticsearch. More specifically, Elastalert constantly runs queries based on the rules that were shown above. These queries run against the data that is stored in Elasticsearch, and together with any matches that were found are saved on Elasticsearch indices. Kibana then allows us to visualize queries, triggers, or any errors that may occur. The relationship between all these different components that make up the ELK stack is presented in Figure 19.

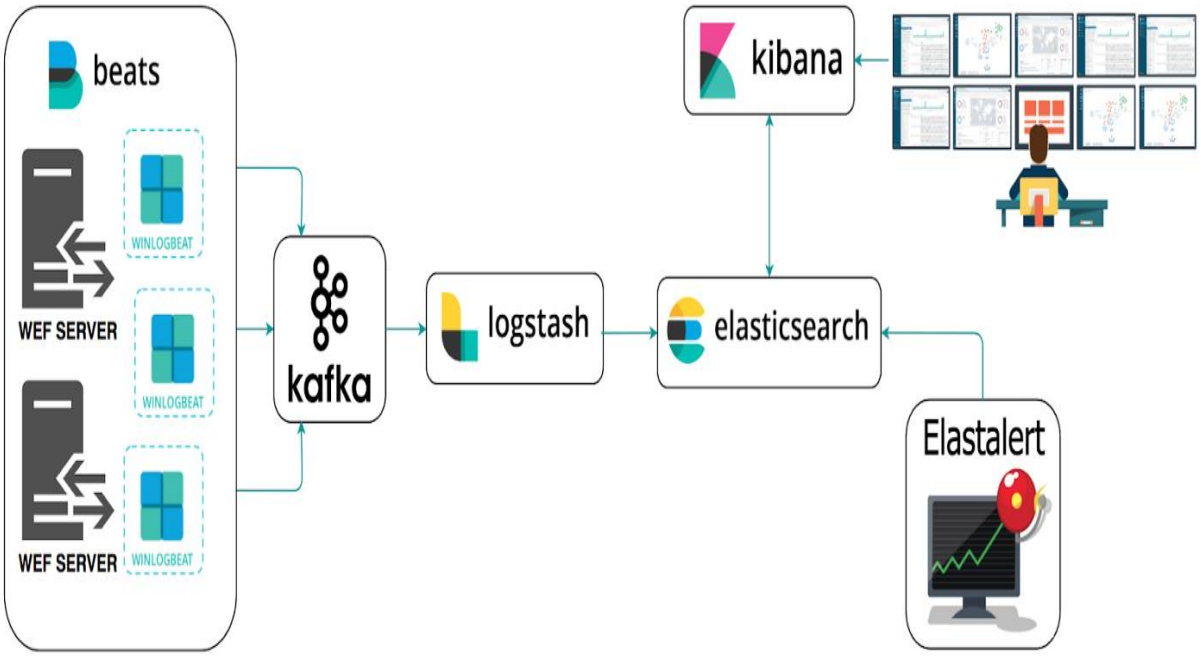


Figure 19: Infographic of **Kafka**, **Elastalert** and the **ELK stack** components working in-tandem

CHAPTER 5: The HELK Lab Environment

5.1 LAB HOSTS

To evaluate the effectiveness and detection capabilities of the HELK platform as a SIEM, a series of prevalent attacks and infection attempts need to be executed against a target host. This host must continuously feed log data into HELK for thorough analysis. To facilitate this process, a lab environment was set up, consisting of the three workstations shown in Figure 20:

1. The first workstation serves as the Ubuntu host responsible for hosting the HELK Platform services. Here, the HELK is installed and operational, as demonstrated in Chapter 3.
2. The second workstation acts as the victim host. In this scenario, the victim host is a Windows 10 64-bit VM workstation configured to send monitored log data and other activities to HELK. This is achieved using tools like PSSysmon Tools, Sysmon Modular, Winlogbeat, and other Windows Policy configurations, all of which will be explained in detail later.
3. The third workstation serves as the attacker's platform. In this case, the attacker's system is a Kali Linux VM workstation equipped with all the necessary tools for executing the predefined attacks.

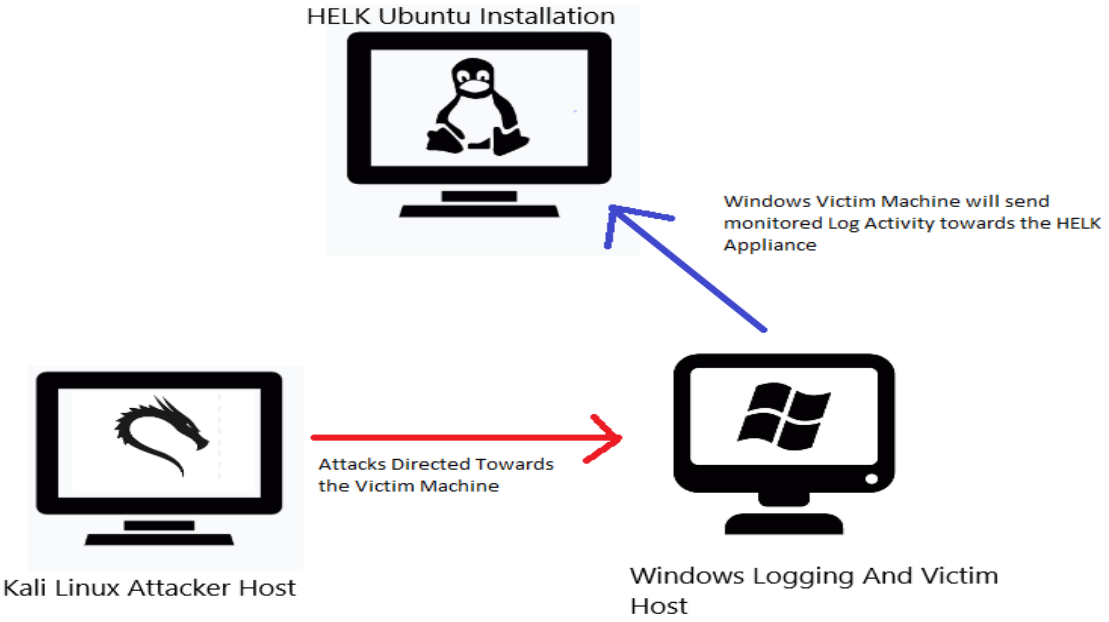


Figure 20: The Lab Environment

5.2 LOGGING AND DELIVERY

In order to effectively determine the effectiveness of HELK as a SIEM solution, sensible and accurate data originating from the Victim host needs to be logged and shipped to HELK. As was explained in chapter 3, a primary tool for shipping data are Beats, that act as agents which

can ship data from various endpoints toward the HELK appliance. The tools that will allow us to log and deliver data to HELK will be the following:

- **Sysmon-modular**
- **PSSysmonTools**
- **Winlogbeat**
- **Windows Policy**

The installation, properties, and use cases for each of the aforementioned tools will be explained below.

5.2.1 PSSYSMON TOOLS

Sysmon[25] is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to **monitor and log system activity** to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. By collecting the events it generates using Windows Event Collection or SIEM agents and subsequently analyzing them, we can identify malicious or anomalous activity and understand how intruders and malware operate on any given network. Note that Sysmon does not provide analysis of the events it generates, nor does it attempt to protect or hide itself from attackers.

In this scenario, PSSysmonTools is just a version of Sysmon designed for Powershell. The required files were cloned from the respective GitHub repository[1] as presented in Figure 21. We then used the commands shown in Figure 22 to import those capabilities into PowerShell.

Figure 21: Cloning PSSysmon Tools

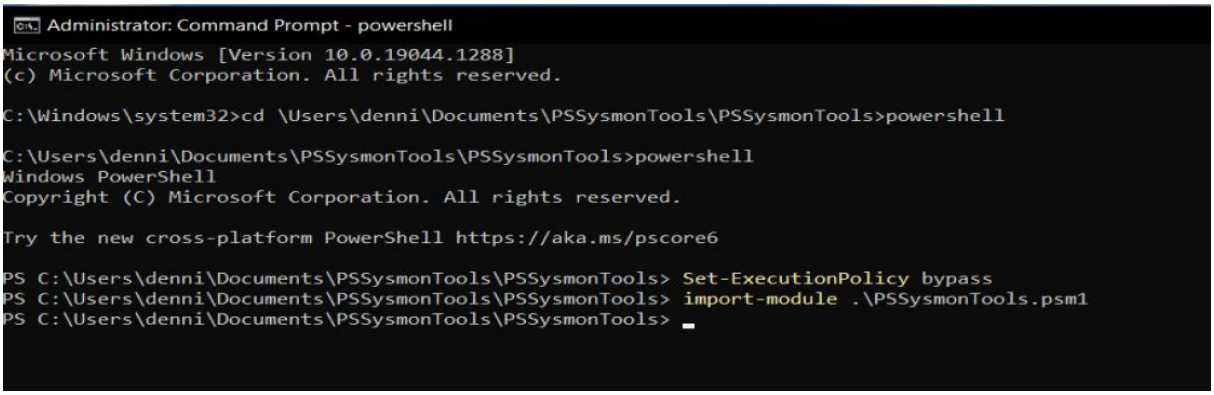


Figure 22: PSSysmonTools installation

[1] https://github.com/mattifestation/PSSysmonTools

5.2.2 SYSMON MODULAR

Sysmon modular is a Microsoft Sysinternals configuration repository for Sysmon, that allows us to customize it for maintenance purposes or for generating specific config files.
First, we need to clone it from its GitHub repository[2] (Figure 23).

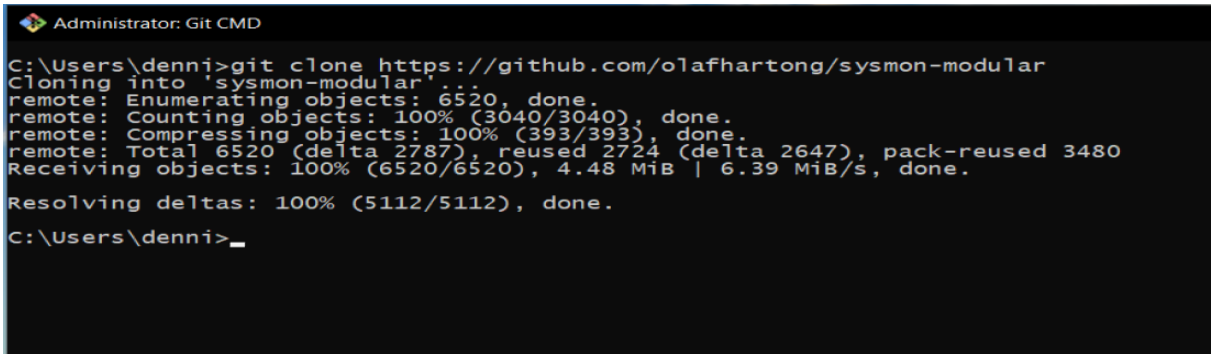


Figure 23: Sysmon Modular cloning

To install Sysmon, we download it from the official Microsoft website and then move it to the sysmon modular folder. Here we execute the command shown in Figure 24:

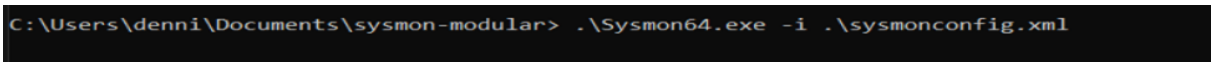


Figure 24: Sysmon installation

We verify that the service is running by checking the Windows Services as clearly shown in Figure 25:

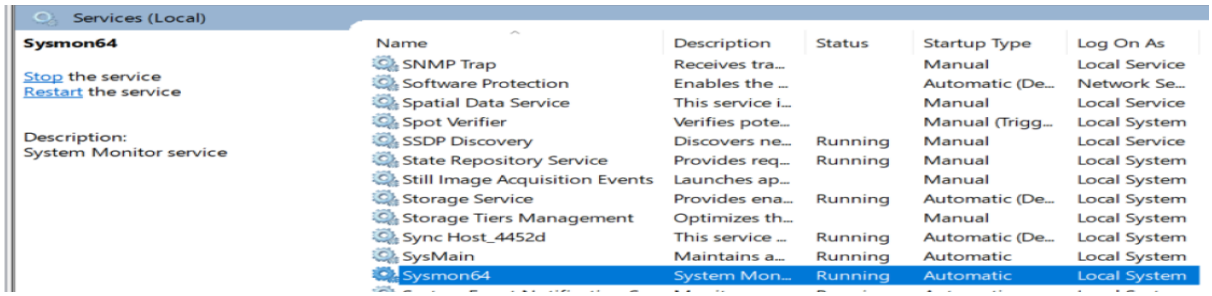


Figure 25: Sysmon Service running

5.2.3 WINDOWS POLICY CONFIGURATION

After installing the Sysinternals utilities, we gain the capability to provide logs to the Beats, enabling them to be shipped to our HELK SIEM. However, additional adjustments are necessary on our client to generate the required logs, such as Powershell logs, Scheduler logs, cmd logs, and more. To accomplish this, we must configure the Windows Policy to capture all activities occurring on the victim host. Heading to the Microsoft Management Console shown in Figure 26 (MMC), we incorporate the Group Policy Object to edit the policy settings for our victim host.
This allows us to fine-tune the logging behavior and ensure comprehensive monitoring of system activities. By carefully configuring the Windows Policy, we establish a robust logging

[2] https://github.com/olafhartong/sysmon-modular

framework, providing valuable insights for our SIEM solution.

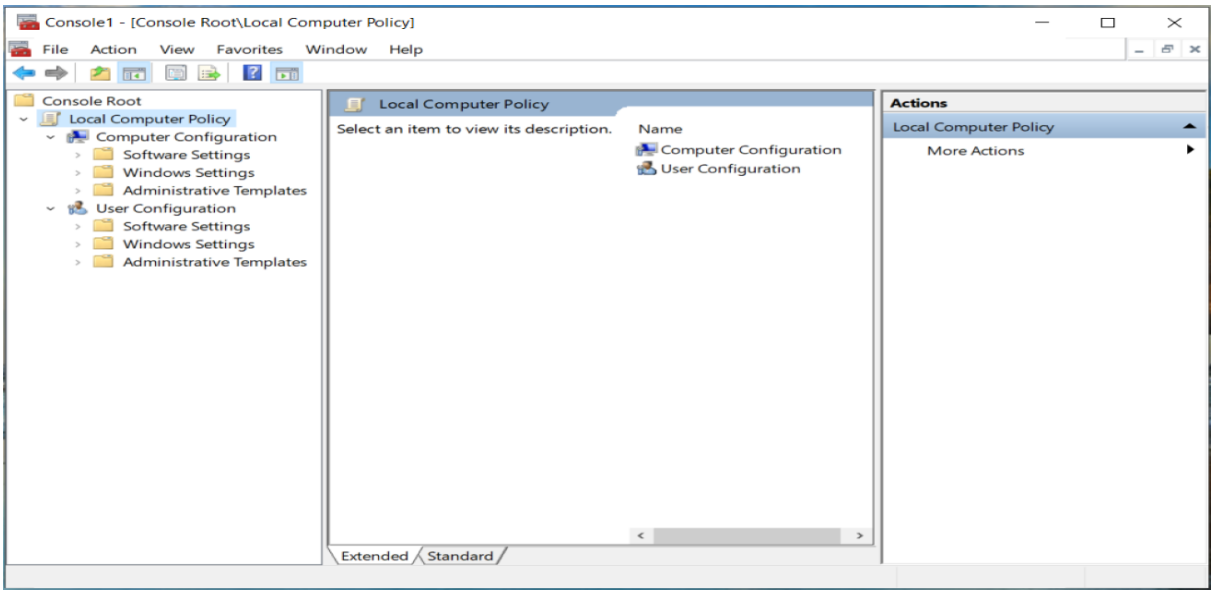


Figure 26: Local Computer Policy

Here we can enable logging for the following categories which will help us monitor our client effectively:

- Process creation
- Audit policy
- Command-lines
- PowerShell
- Scheduled tasks

The respective Windows Policy configuration settings will be shown below:

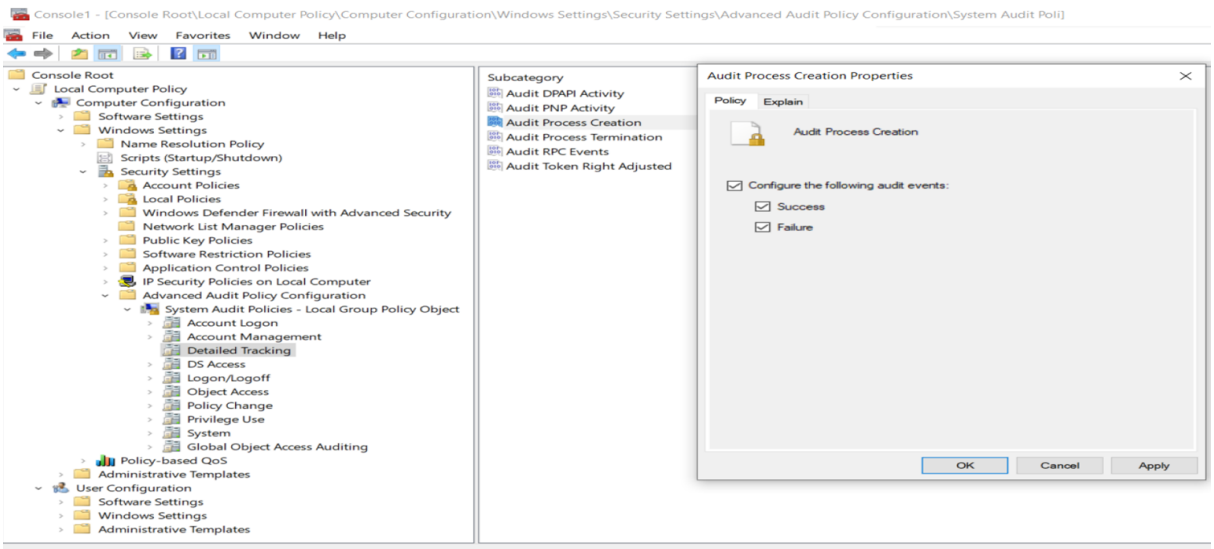


Figure 27: Process Creation Logging

Process creation logging, shown in Figure 27, enables monitoring of Windows Events with ID 4688 in the local Event Viewer. These events occur every time a new process starts and provide crucial information, including the timestamp, process name, parent process, process command line, and more. The MITRE ATT&CK framework heavily relies on Process Creation events to classify threats, making it imperative for us to diligently monitor such activities.

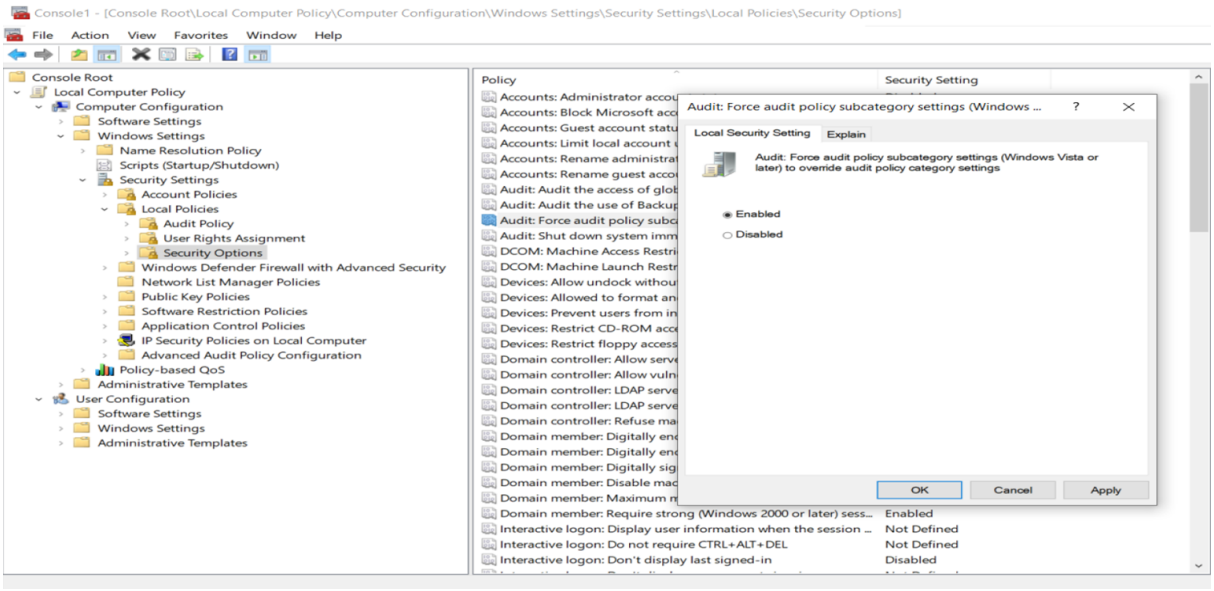


Figure 28: Audit Policy Logging

The primary objective of the system audit logging policy is to ensure the accurate and consistent collection of critical system information. This enables the timely detection of security violations, unauthorized data disclosures, as well as performance issues and application flaws. By maintaining comprehensive logs, the organization gains valuable insights into the system's activities, facilitating proactive monitoring and effective incident response. The enablement of this logging is shown in Figure 28.

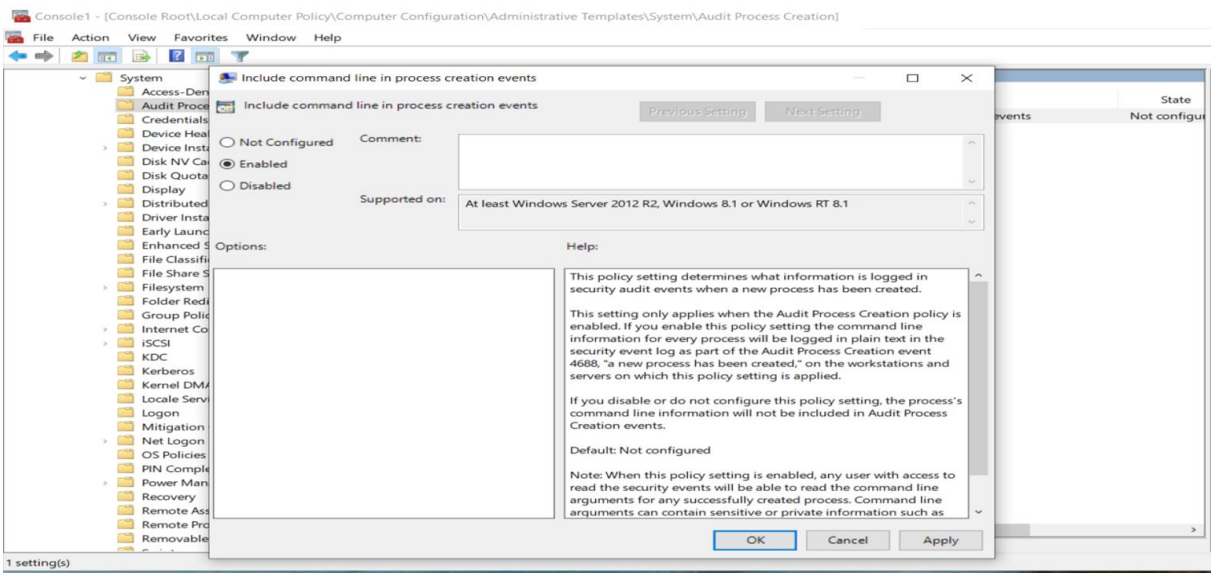


Figure 29: Command-Line Logging

Command-line logging is an invaluable extension to the Windows auditing and event system. After enabling command line logging in Figure 29,the information available in ID 4688 events within the Windows security event log is significantly enhanced. This enhancement comes in the form of detailed command-line arguments utilized by a process. By capturing and logging this data, it offers a deeper level of insight into the activities and actions taken by various processes on the system.

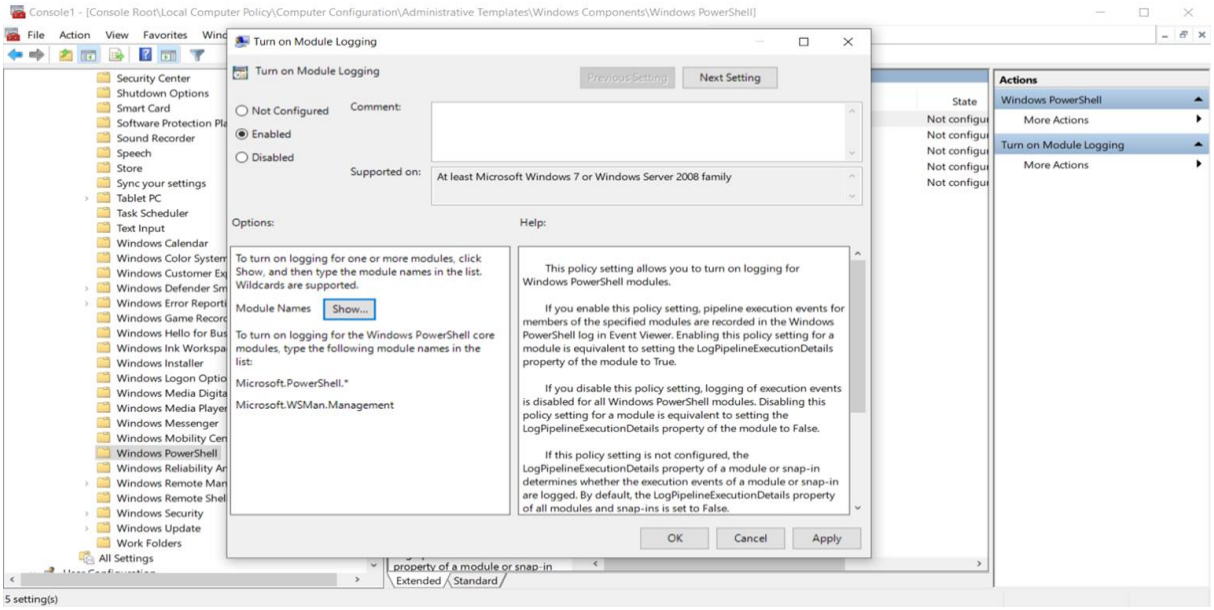


Figure 30: Powershell Logging

As discussed in Chapter 3, attackers are increasingly exploiting the capabilities of Windows PowerShell to carry out their malicious operations. PowerShell serves as a potent command-line environment and scripting language that, by default, leaves minimal traces of its execution in typical Windows environments. Thus we enabled Powershell Logging as shown in Figure 30.

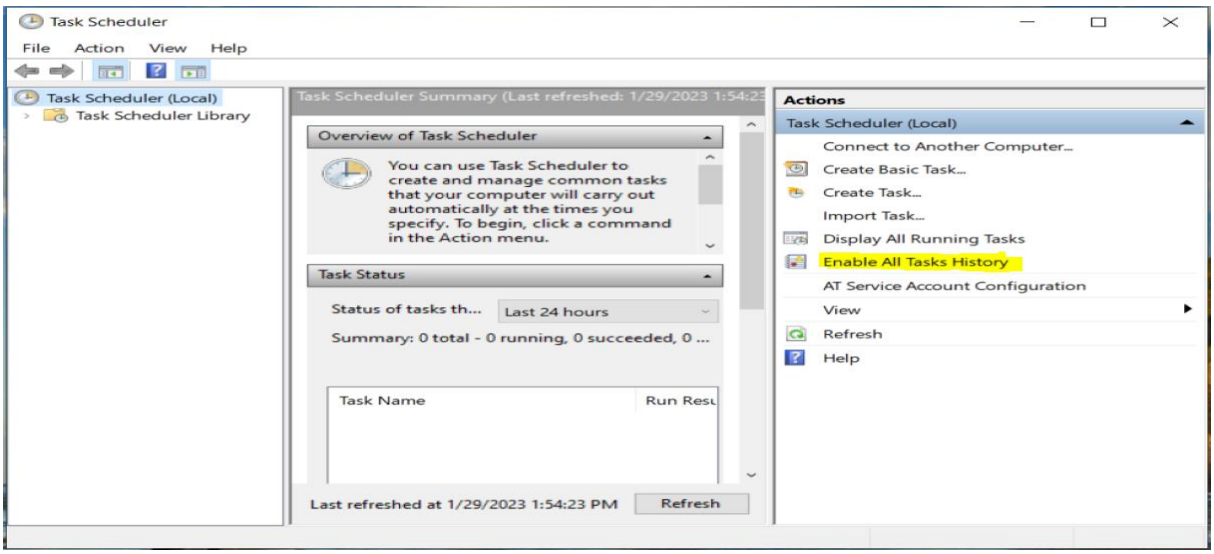


Figure 31: Task Scheduler Logging

Enabling Task Scheduler event logging, seen in Figure 31, facilitates the monitoring of automatically executed routine tasks. By doing so, potential attempts by attackers to exploit the Windows Task Scheduler for scheduling malicious programs for initial or recurrent execution can be detected. Moreover, attackers might leverage the Windows Task Scheduler for achieving persistence by launching applications during system startup or on a scheduled basis. Additionally, this feature could be misused to execute remote code and run a process under the context of a specified account, potentially leading to Privilege Escalation scenarios.

Finally, we activated firewall logging on our Windows 10 Host which we can see in Figure 32. This is a crucial step that grants us visibility into network traffic and essential security events. The significance of firewall logs lies in their ability to detect potential security threats, including unauthorized access attempts and malware activities. By diligently analyzing these logs, we gain valuable insights into patterns of behavior that may indicate a security breach.

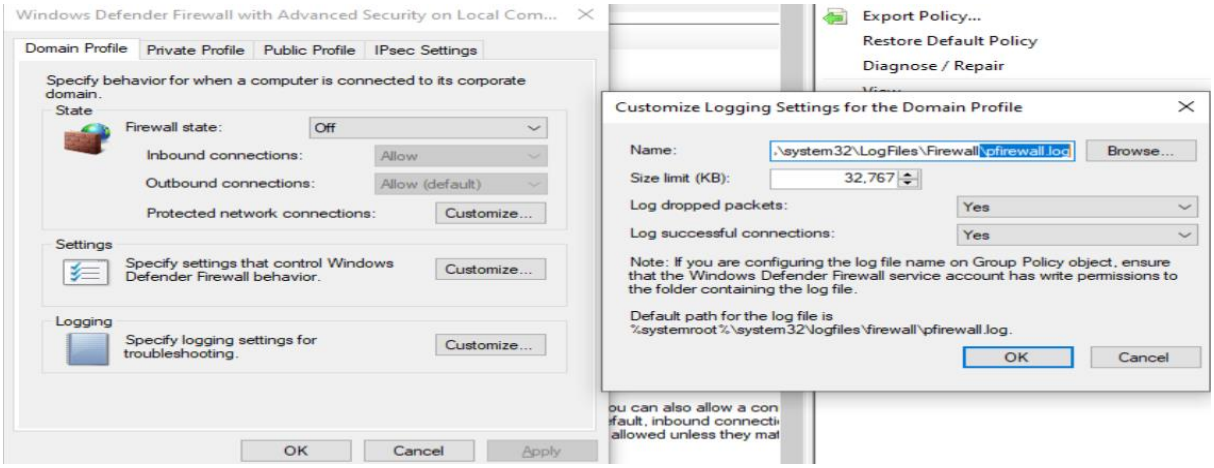


Figure 32: Firewall Logging

5.2.4 WINLOGBEAT

Winlogbeat serves as the essential shipping agent, enabling the seamless transfer of logs generated by the Windows victim host to Logstash within the HELK ecosystem.

To get started, we'll obtain the 64-bit version of Winlogbeat [26]. This package includes a .yml configuration file, which we'll need to replace with another .yml configuration file that is compatible with HELK.

Next, we'll carefully modify the "hosts" field in the configuration file to align with our HELK IP address. This crucial step ensures that Winlogbeat precisely knows the destination to ship our logs to, as depicted below in Figure 33.

```
#-------------------------- Windows Logs To Collect --------------------------------
winlogbeat.event_logs:
  - name: Application
    ignore_older: 30m
  - name: Security
    ignore_older: 30m
  - name: System
    ignore_older: 30m
  - name: Microsoft-windows-sysmon/operational
    ignore_older: 30m
  - name: Microsoft-windows-PowerShell/Operational
    ignore_older: 30m
    event_id: 4103, 4104
  - name: Windows PowerShell
    event_id: 400,600
    ignore_older: 30m
  - name: Microsoft-Windows-WMI-Activity/Operational
    event_id: 5857,5858,5859,5860,5861
    ignore_older: 30m
  - name: Microsoft-Windows-Windows Firewall With Advanced Security/Firewall
    ignore_older: 48h
#------------------------------ Kafka output --------------------------------
output.kafka:
  # initial brokers for reading cluster metadata
  # Place your HELK IP(s) here (keep the port).
  # If you only have one Kafka instance (default for HELK) then remove the 2nd IP that has port 9093
  hosts: ["192.168.196.132:9092"]
  topic: "winlogbeat"
  ############################## HELK Optimizing Latency ######################
  max_retries: 2
  max_message_bytes: 1000000
```

Figure 33: Winlogbeat configuration file

After changing the configuration file to match our desired IP Address/Port, we install the Winlogbeat service with the help of the command console as can be seen in Figure 34:

```
Administrator: Command Prompt - powershell
PS C:\Users\denni\Desktop\winlogbeat-8.6.1-windows-x86_64\winlogbeat-8.6.1-windows-x86_64> .\install-service-winlogbeat.ps1

Status   Name               DisplayName
------   ----               -----------
Stopped  winlogbeat         winlogbeat


PS C:\Users\denni\Desktop\winlogbeat-8.6.1-windows-x86_64\winlogbeat-8.6.1-windows-x86_64>
```

Figure 34: Winlogbeat installation

With all the aforementioned configurations completed on the Windows 10 VM, we have successfully established the process of shipping security, OS, and network-related logs to the HELK appliance, essential for our testing purposes. In the upcoming chapter, we will utilize the Kibana UI to visualize these logs and embark on exploring the extensive capabilities offered by HELK.

CHAPTER 6: Attacks and HELK's Response

6.1 HELK UI & KIBANA

At this stage, the configuration of both HELK and our Windows VM host has been completed successfully, enabling us to proceed with testing its capabilities and evaluating its suitability as a SIEM and Threat Hunting solution for SMEs. To begin, it is essential to acquaint ourselves with the Kibana UI, depicted in Figure 35, which serves as our primary interface for interacting with HELK. Through Kibana, we can effectively analyze log data, receive alerts, apply filters, and perform various other tasks to gain valuable insights and enhance our security posture.

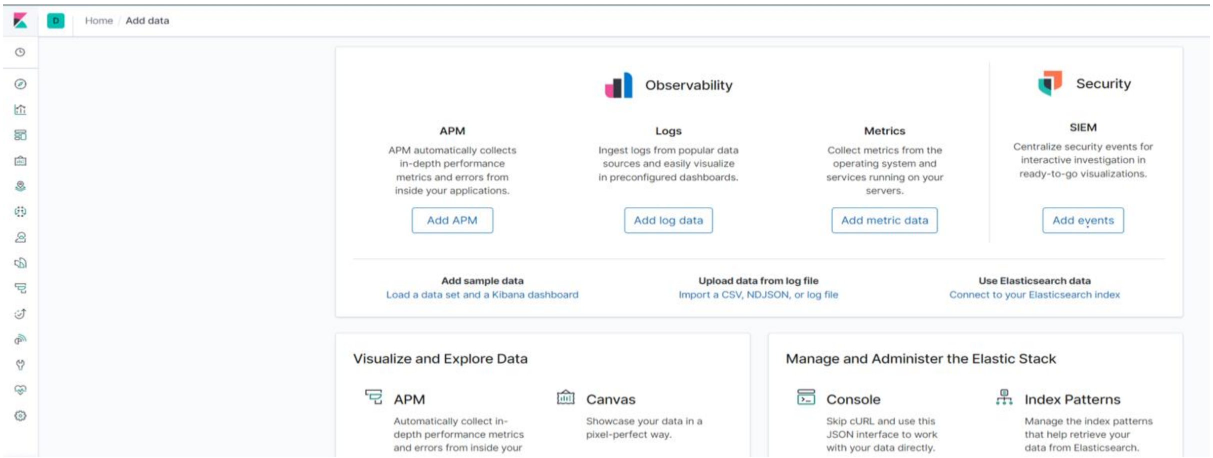


Figure 35:Kibana User Interface

HELK offers a wide array of custom chart fields and filters that enable effective data visualization. As a security analyst, utilizing the different sets of fields offered by HELK, as a means to quickly display necessary information, is an integral part of Threat Hunting

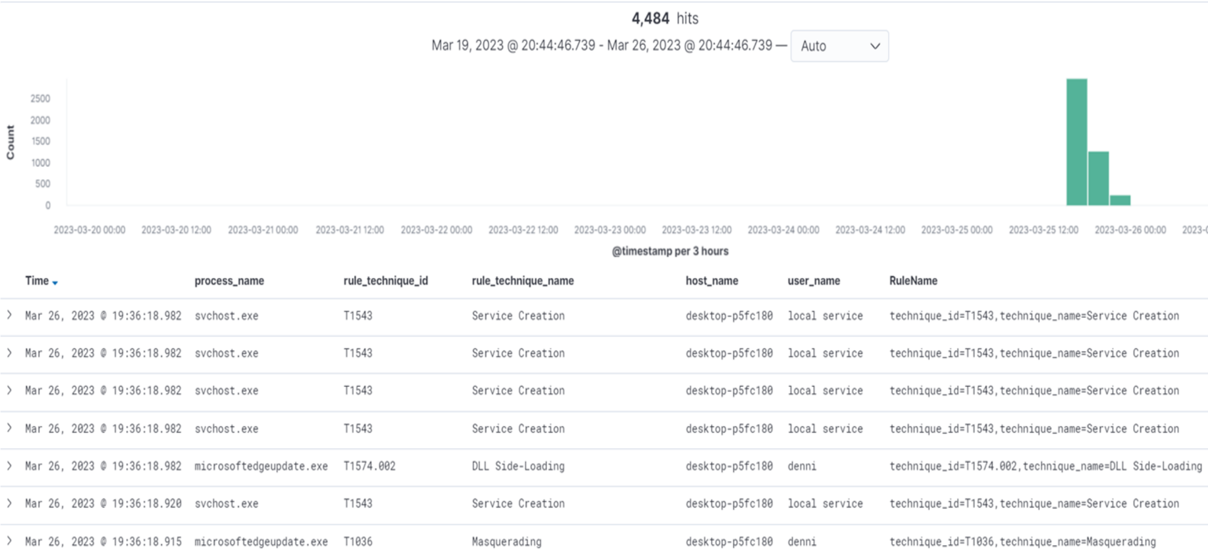


Figure 36: HELK Threat Hunting UI

We have applied the following fields, depicted in Figure 36, which give us an overview and some basic information on the alerts that have been triggered thus far:

- **Time**
- **Process Name**
- **Rule Technique ID**
- **Rule Technique Name**
- **Hostname**
- **Username**

In this instance for example, "**Rule Name**" refers to the respective Elastalert and Sigma rules that are triggered which in turn created an Alert. Additionally, "**Technique ID and Name**" refer to the MITRE ATT&CK framework adversary techniques that match the respective Elastalert rules. Having a technique for each alert allows a security operator to quickly explore the potential impact a specific security event can have, by referring to the MITRE ATT&CK framework.

6.2 SIMULATING ATTACKS

In order to assess the effectiveness of HELK, we conducted a series of tests in a lab environment as previously described. Using **Kali Linux,** we simulated various attack scenarios including:

- **Reconnaissance and enumeration**
- **Exploitation and privilege escalation**
- **Brute force attacks**
- **Malware injection**
- **Maintaining persistence**
- **Covering tracks**

By meticulously crafting a realistic use case wherein each step is interlinked, we gained invaluable insights into the significance of implementing defense mechanisms in layers. Moreover, our experience underscored the essence of threat hunting, which entails diligently tracking adversaries across multiple seemingly disparate events to uncover hidden patterns and potential threats.

At this juncture, it is crucial to emphasize that the objective of this thesis does not revolve around achieving successful infiltration of a "Honeypot " System like the one we are currently engaged with. Instead, our focus lies in evaluating the detectability of HELK. As a result, we have implemented several modifications to the Windows 10 Host, including but not limited to:

- Disabling Windows Defender
- Disabling Windows Firewall
- Reverting to an older and more "un-secure" version of Windows 10 from 2021

The aforementioned changes which will make the system more vulnerable to outside attacks and exploitations, thus giving us more freedom of movement when choosing our preferred method.

6.2.1 RECONNAISSANCE AND ENUMERATION

Reconnaissance and Enumeration is the first phase in a penetration testing process where an attacker collects information about the target system and its environment. [27] This phase involves passive and/or active information-gathering techniques to discover potential vulnerabilities, misconfigurations, and other weaknesses in the target system. In this phase, the

attacker tries to learn as much as possible about the target system, such as the operating system, applications, open ports, network topology, and user accounts.
Enumeration involves actively probing the target system to gather more detailed information about the system and its users. The goal of this phase is to build a complete picture of the target system that can be used in subsequent phases of the attack.

NMAP

Nmap[28], short for "Network Mapper," is a free and open-source tool for network exploration, management, and security auditing. It is used to discover hosts and services on a computer network, as well as to create a map or "topology" of the network. Nmap uses a combination of techniques, including port scanning, service identification, and OS fingerprinting, to gather information about network hosts and services. It can be used to perform a wide range of tasks, such as:

- Host discovery: Nmap can scan a network and identify which hosts are active and available.
- Port scanning: Nmap can scan the ports on a host and identify which services are running on each port. This can help identify potential vulnerabilities or security risks.
- Service detection: Nmap can identify the type and version of services running on a host, such as web servers, FTP servers, or database servers.
- OS fingerprinting: Nmap can attempt to identify the operating system running on a host based on its response to certain network probes.

In this instance, we will use the console version of NMap already present in our Kali installation to scan the Windows Host.
We tried both a full OS, Service, and Port scan with the following Commands shown in Figure 37.

- **sudo nmap -p- -T4 -A -v 192.168.196.129**
- **sudo nmap -Pn -A 192.168.196.129**

Overall, these commands instruct Nmap to perform an aggressive scan of all ports on the target system, using various techniques to identify the operating system and services running on the system.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn -A 192.168.196.129

Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-22 13:30 EDT
Nmap scan report for 192.168.196.129 (192.168.196.129)
Host is up (0.00016s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
MAC Address: 00:0C:29:53:9B:5B (VMware)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1709 - 1909
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2023-04-22T17:30:19
|_  start_date: N/A
|_nbstat: NetBIOS name: DESKTOP-TNC39G1, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:53:9b:5b (VMware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.16 ms 192.168.196.129 (192.168.196.129)
```

Figure 37: NMap Scan Example

Below in Figure 38, we will explore the results we received from our Kibana UI in real-time as the Nmap scan was running. Thankfully we can see that the network traffic originating from the Kali Host is indeed detected on a log level, meaning that logs related to Incoming connections or Firewall traffic are being delivered to HELK, despite whether an Alert was raised or not.

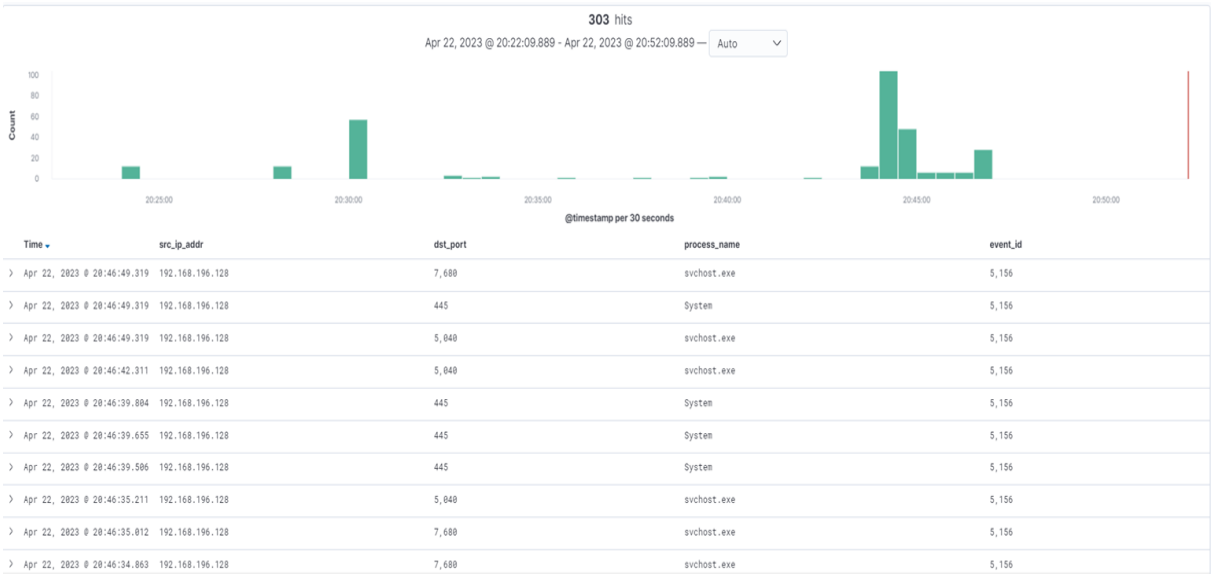
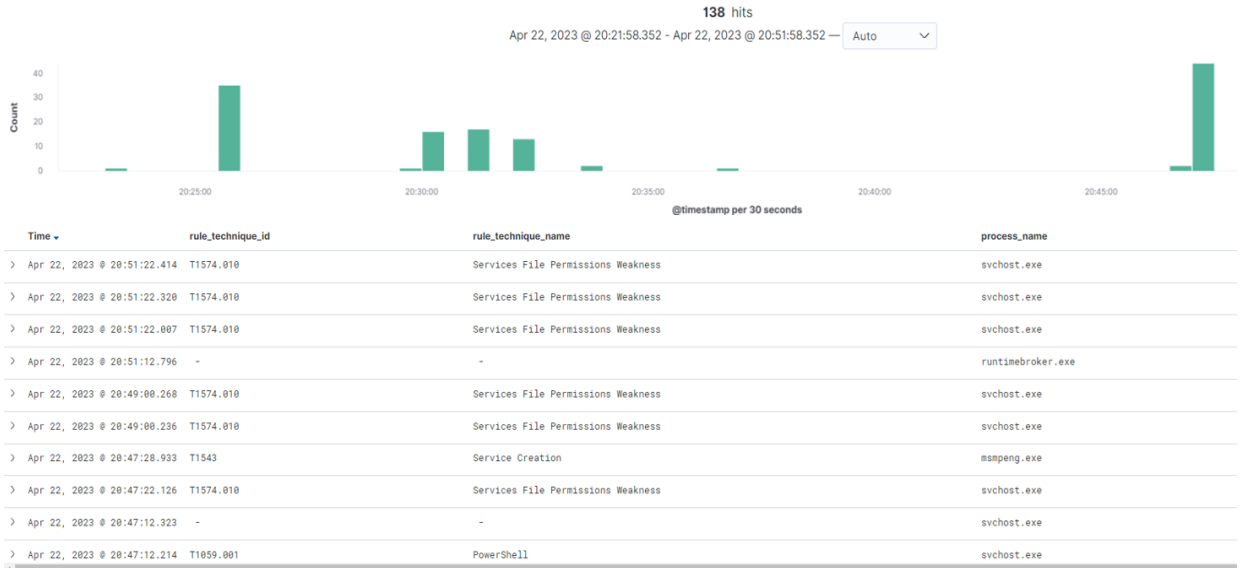


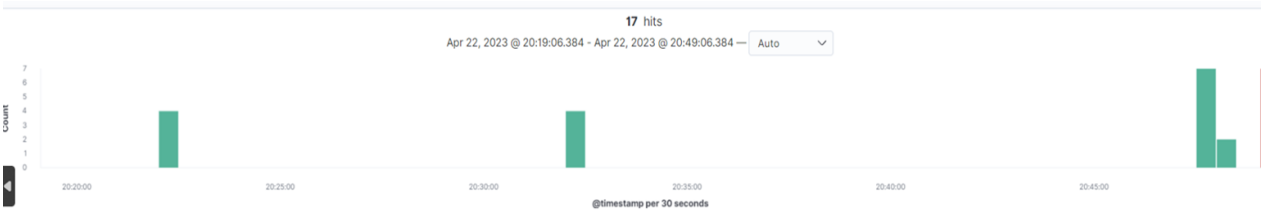
Figure 38: Nmap Scan Results v1

Figure 38 above, clearly shows us that incoming network connections (**Event ID 5156**) from the Kali Host with Source IP **192.168.196.128** are indeed being captured by HELK, which at the very least implies that we have configured our Windows 10 Machine to send logged network traffic to HELK successfully. What we are interested in, however, is whether an alert was raised in the Kibana UI related to the external Nmap scan. Figures 39 and 40 show the respective Sysmon rules and Elastalert alerts that were triggered in the same timeframe as the Nmap scans.



Time	rule_technique_id	rule_technique_name	process_name
Apr 22, 2023 @ 20:51:22.414	T1574.010	Services File Permissions Weakness	svchost.exe
Apr 22, 2023 @ 20:51:22.320	T1574.010	Services File Permissions Weakness	svchost.exe
Apr 22, 2023 @ 20:51:22.007	T1574.010	Services File Permissions Weakness	svchost.exe
Apr 22, 2023 @ 20:51:12.796	-	-	runtimebroker.exe
Apr 22, 2023 @ 20:49:00.268	T1574.010	Services File Permissions Weakness	svchost.exe
Apr 22, 2023 @ 20:49:00.236	T1574.010	Services File Permissions Weakness	svchost.exe
Apr 22, 2023 @ 20:47:28.933	T1543	Service Creation	msmpeng.exe
Apr 22, 2023 @ 20:47:22.126	T1574.010	Services File Permissions Weakness	svchost.exe
Apr 22, 2023 @ 20:47:12.323	-	-	svchost.exe
Apr 22, 2023 @ 20:47:12.214	T1059.001	PowerShell	svchost.exe

Figure 39: Nmap Scan Results v2 (Sigma rules triggered)



Time	rule_name	match_body.process_name	match_body.event_id
Apr 22, 2023 @ 20:48:11.631	96036718-71cc-4027-a538-d1587e0006a7_0 Windows Processes Suspicious Parent Directory	svchost.exe	4688
Apr 22, 2023 @ 20:48:11.623	96036718-71cc-4027-a538-d1587e0006a7_0 Windows Processes Suspicious Parent Directory	svchost.exe	4688
Apr 22, 2023 @ 20:47:43.107	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	svchost.exe	4688
Apr 22, 2023 @ 20:47:43.103	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	conhost.exe	4688
Apr 22, 2023 @ 20:47:43.098	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	conhost.exe	4688
Apr 22, 2023 @ 20:47:43.093	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	conhost.exe	4688
Apr 22, 2023 @ 20:47:43.089	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	conhost.exe	4688
Apr 22, 2023 @ 20:47:43.083	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	conhost.exe	4688
Apr 22, 2023 @ 20:47:43.078	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	svchost.exe	4688
Apr 22, 2023 @ 20:32:27.249	e4a6b256-3e47-40fc-89d2-7a477edd6915_0 System File Execution Location Anomaly	svchost.exe	4688

Figure 40: Nmap Scan Results v3 (Elastalert Alerts raised)

Figures 39 and 40 respectively, show us that a relevant rule or alert was **NOT** triggered for the external Nmap scan on Ports 445 and 135. The fact that they coincide within the same timeframe and the fact that the "Process name" field lists the process *svchost.exe* means that at the very least, HELK was indeed able to raise an alert, just not for the correct offense. External Nmap scans towards a Windows 10 host and the process svchost.exe are related because svchost.exe is a critical system process in Windows that is responsible for hosting multiple Windows services.

When Nmap performs a scan of a Windows 10 host, it sends packets to various ports on the target system to determine which ports are open and what services are running on those ports. In some cases, the Windows services that are being hosted by svchost.exe may be listening on those ports and will respond to Nmap's packets. We can safely conclude then, that HELK only **Partially Detected** the aforementioned Recon activity.

The modularity of HELK allows us to create our own Sigma [29] rules or modify the already existing ones which will enhance its capabilities, and this is exactly what we will do in this instance in order to determine the feasibility of such a practice in a real word environment.

We start by outlining what we need our rule to detect:

1. We want our rule to detect Incoming connections from the Windows Filtering Platform, namely **Event ID 5156**
2. We want it to detect incoming connections to common service ports like **445,135,3389 etc.**
3. We want to filter for inbound connections.
4. Given the scenario in hand, also including the processName **"svchost.exe"** would help as well.

Taking all of the above into consideration, and with the help of the Sigma Rule creation Tutorial on Github[3] our Sigma rule will look something like the one shown in Figure 41:

```
title: Incoming Nmap Scans on Ports 135, 445 with Event ID 5156 and Process Name svchost.exe
status: experimental
description: Detects incoming Nmap scans on ports 135 and 445 with the process name svchost.exe and Event ID 5156.

references:
    - https://attack.mitre.org/techniques/T1046/
    - https://attack.mitre.org/techniques/T1204/

tags:
    - attack.t1046
    - attack.t1204
    - attack.tactic.discovery

logsource:
    product: windows
    service: security
    description: 'Event ID 5156 - Windows Filtering Platform has permitted a connection'
    filter:
        - "5156"

detection:
    selection:
        EventID: 5156
        DestinationIP: '192.168.196.129'
        DestinationPort: 445 OR 135
        ProcessName: "svchost.exe"
        Direction: "Inbound"
    condition: selection

fields:
    - src_ip
    - dst_ip
    - dst_port

falsepositives:
    - None at the moment

level: high
```

Figure 41: Nmap Sigma rule example

As discussed in Chapter 3, the seamless collaboration between Elastalert queries and Sigma rules within HELK enables the detection and alerting of potential security threats. Sigma rules offer a vendor-neutral, easily comprehensible format to define detection logic, making them

[3] https://github.com/SigmaHQ/sigma#readme

translatable to various SIEM solutions, including HELK. Once a Sigma rule is converted to a specific SIEM query format like an Elastalert query, it becomes an integral part of the continuous log monitoring process. Elastalert diligently watches the logs and triggers alerts whenever a matching event is identified.

HELK leverages Sigma rules by translating them into Elastalert queries and incorporating them into the Elastalert rules folder. Subsequently, Elastalert processes these rules, executing the specified queries against incoming logs, and promptly generates alerts upon finding a match. The employment of Sigma rules enables security analysts to craft detection logic once and apply it across various SIEM solutions, thus streamlining the management of security threats in a resource-efficient and cost-effective manner.

To make our rule compatible with HELK, we must convert it into an Elastalert Query. Elastalert is designed to read various log types present in Logstash and activate an alert when specific conditions are met. For this scenario, we are focusing on Windows Security Event logs, which are forwarded to HELK using WinLogbeat. To facilitate the conversion process, HELK offers tools that simplify the translation. The resulting Elastalert Query is depicted in Figure 42.

```
alert:
-debug
description: Detect Incoming NMAP Scans on Ports 135 and 445 with process name svchost.exe and WinEvent ID 5156
filter:
-query
    query_string:
        query: (event_id:"5156" AND dst_port:"445\ OR\ 135" AND process_path:"svchost.exe" AND Direction:"Inbound")
index: logs-endpoint-winevent-security-*
name: Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0
priority: 2
realert:
    minutes: 0
timestamp_field: etl_processed_time
type: any
```

Figure 42: Nmap Elastalert Query

In order to test whether our new rule will work, we proceed with executing another nmap scant, as before, towards our Windows 10 Host.

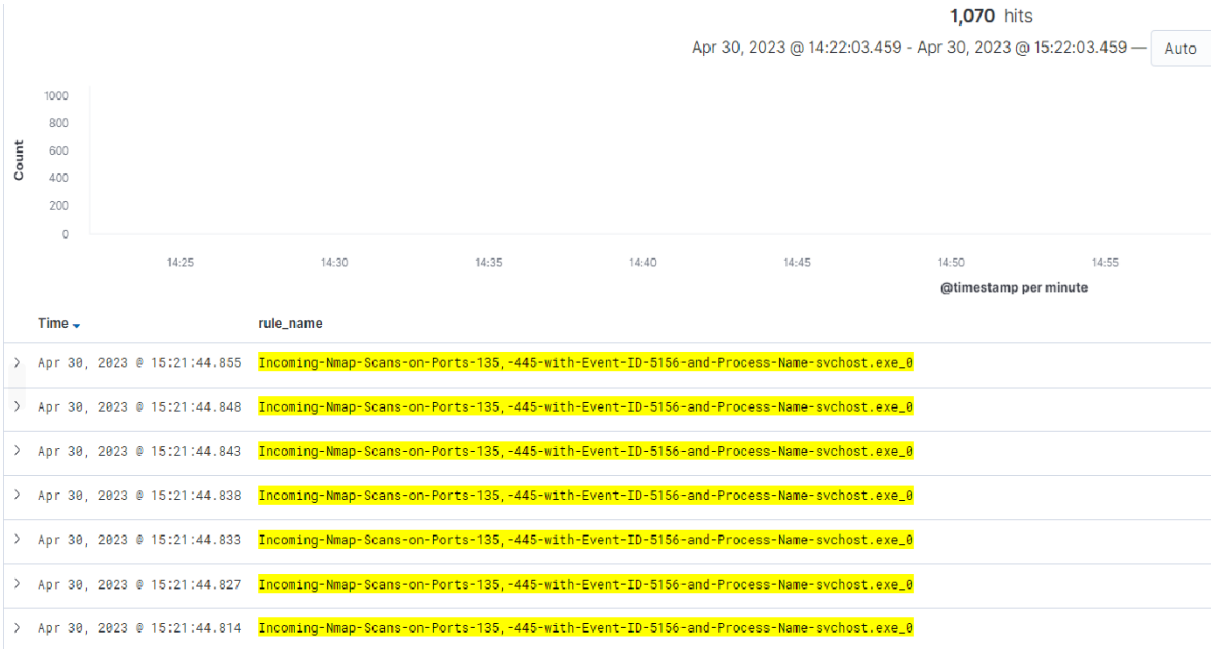


Figure 43: Nmap Detected with new Sigma rule

Figure 43 clearly shows that HELK effectively identified the external Nmap scan on Ports 445 and 135. However, it is essential to emphasize that this rule requires fine-tuning to minimize false positives and align with the specific network requirements of the environment in use. The creation of the aforementioned rule showcases HELK's modularity and flexibility, enabling the expansion of its threat detection capabilities. Nevertheless, crafting robust and efficient rules is an ongoing endeavor that demands a comprehensive grasp of Networks, Malware, and Operating Systems. In summary, HELK demonstrates its potential as a powerful SIEM solution by effectively detecting the Nmap scan. Fine-tuning rules to meet specific needs and staying abreast of cybersecurity trends are crucial for optimizing its performance in real-world production environments.

NESSUS

Nessus [30] is a proprietary vulnerability scanner used for identifying vulnerabilities and misconfigurations in computer systems, network devices, and other applications. It scans computer systems, networks, and applications for security vulnerabilities and provides detailed reports on its findings. Nessus uses a database of known vulnerabilities and attack patterns to identify potential weaknesses in a target system or network. It then assigns a severity level to each vulnerability, based on the likelihood of exploitation and the potential impact on the system.

In this scenario we will make use of Nessus to identify possible security vulnerabilities on the Windows VM while also testing whether HELK will detect this incoming Nessus Vulnerability scan. After the scan is complete, Nessus will compile a comprehensive report with any vulnerabilities and open Network Ports present on our Victim Host.

192.168.196.138

0	0	4	1	20
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 25

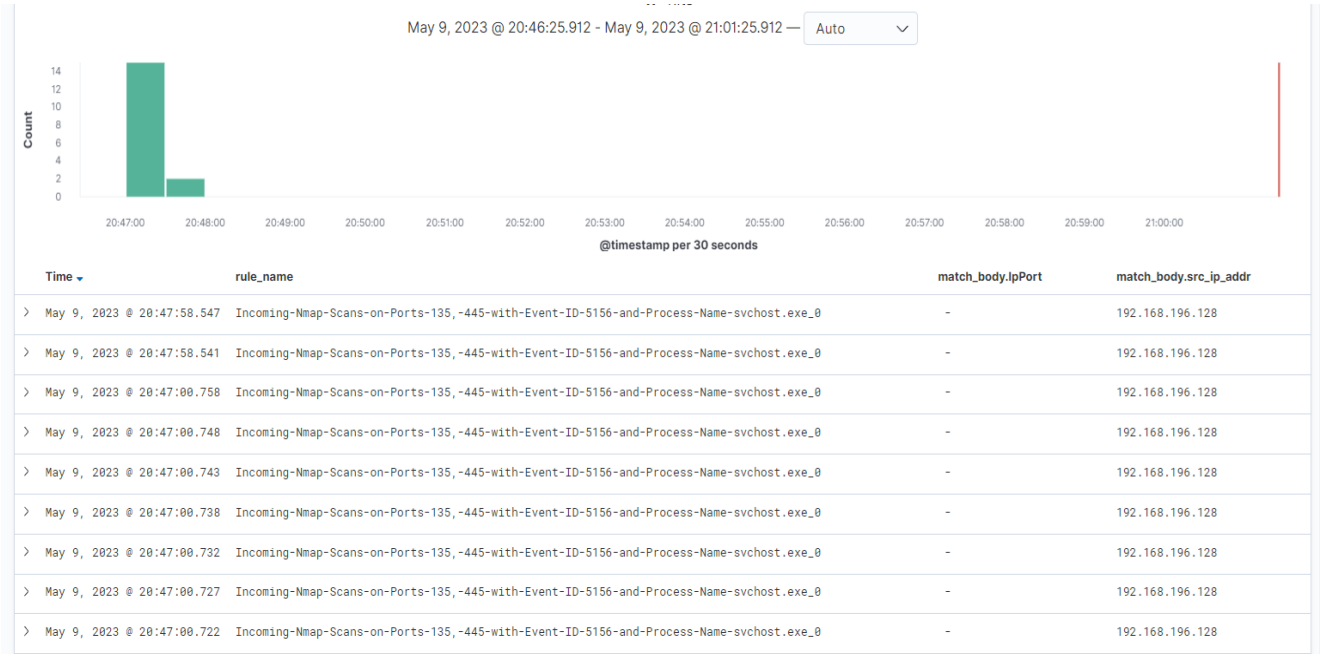
SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
MEDIUM	6.5	2.5	18405	Remote Desktop Protocol Server Man-in-the-Middle Weakness
MEDIUM	5.3	-	57608	SMB Signing not required
MEDIUM	4.0	-	58453	Terminal Services Doesn't Use Network Level Authentication (NLA) Only
MEDIUM	4.3*	-	57690	Terminal Services Encryption Level is Medium or Low
LOW	2.6*	-	30218	Terminal Services Encryption Level is not FIPS-140 Compliant
INFO	N/A	-	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	-	45590	Common Platform Enumeration (CPE)
INFO	N/A	-	10736	DCE Services Enumeration
INFO	N/A	-	54615	Device Type
INFO	N/A	-	35716	Ethernet Card Manufacturer Detection
INFO	N/A	-	86420	Ethernet MAC Addresses
INFO	N/A	-	53513	Link-Local Multicast Name Resolution (LLMNR) Detection

Figure 44: Nessus Scan Results

According to the Nessus report depicted in Figure 44, the most severe of vulnerabilities present in our system is located on **RDP Port tcp/3389.**

The Remote Desktop Protocol Server (Terminal Service) has a vulnerability that could allow a man-in-the-middle (MiTM) attack. The RDP client does not check the identity of the server when setting up encryption, so an attacker who can intercept traffic from the RDP server can establish encryption with the client and server without being detected. This type of attack would allow the attacker to access any sensitive information transmitted, such as authentication credentials.

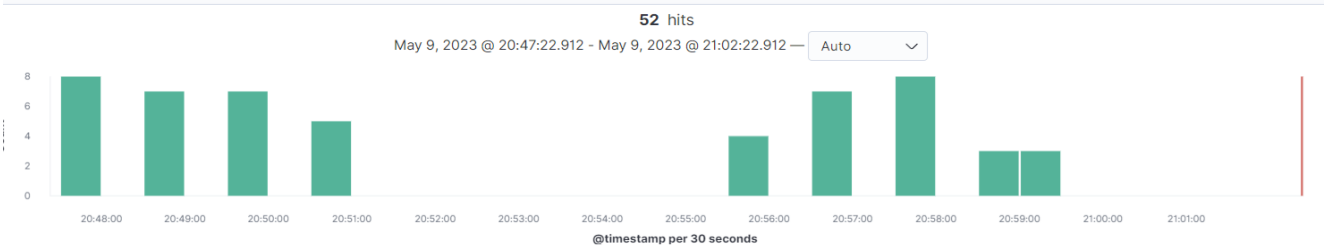
After completing the scan, we can say with confidence that HELK detected this activity which triggered multiple rules**,** including the one we created for Nmap. The triggering of these rules caused by the Nessus scan is presented in Figure 45.



Time	rule_name	match_body.lpPort	match_body.src_ip_addr
May 9, 2023 @ 20:47:58.547	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:58.541	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.758	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.748	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.743	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.738	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.732	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.727	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128
May 9, 2023 @ 20:47:00.722	Incoming-Nmap-Scans-on-Ports-135,-445-with-Event-ID-5156-and-Process-Name-svchost.exe_0	-	192.168.196.128

Figure 45: Nessus Scan triggered NMap rule

It is to be expected then, that the Nessus Network Scan triggered the Nmap rule we created, as both utilities work in similar ways and probe the Windows 10 Host for open ports. We can see however that another rule was triggered, shown in Figure 46, which relates to Windows Event ID 4625 an event that is logged for any logon failure.



Time	rule_name	match_body.lpPort	match_body.src_ip_addr
May 9, 2023 @ 20:59:27.682	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	40496	192.168.196.128
May 9, 2023 @ 20:59:27.676	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	55002	192.168.196.128
May 9, 2023 @ 20:59:27.662	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	54988	192.168.196.128
May 9, 2023 @ 20:58:32.150	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	47362	192.168.196.128
May 9, 2023 @ 20:58:32.143	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	47346	192.168.196.128
May 9, 2023 @ 20:58:32.137	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	52232	192.168.196.128
May 9, 2023 @ 20:57:36.970	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	38694	192.168.196.128
May 9, 2023 @ 20:57:36.964	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	32830	192.168.196.128
May 9, 2023 @ 20:57:36.959	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	32822	192.168.196.128

Figure 46: Nessus Scan detected by HELK

Normally we would expect Nessus to detect a number of CVE's on the system for us to exploit in order to proceed with the "**Exploitation**" part of our testing methodology. CVE's are assigned by the MITRE Corporation, and stand for Common Vulnerabilities and Exposures. It is a standardized system used to identify and track known vulnerabilities in software and hardware products. Each CVE entry represents a unique identifier for a specific vulnerability.

Instead, we can take advantage of the fact that Port 3389 was reported as vulnerable and open by Nessus on the Windows 10 Host, which will allow us to gain entry into the system by Brute-Forcing our way in.

6.2.2 BRUTE FORCE ATTACK

In order to gain access to the Victim Host, we will make use of the fact that port 3389 is open & vulnerable coupled with some built in capabilities offered by our Kali Linux System which will allow us to crack the password of our Windows System User.
To achieve this, we will use a built-in tool of Kali, named **Hydra**. Hydra [31] is a popular and powerful password-cracking tool used in penetration testing and security assessments. It is designed to perform brute-force attacks and dictionary attacks against various network protocols and applications to discover weak or compromised passwords.
In order to make our Brute Force attack as effective as possible, we provided Hydra .txt files of the 10.000 most commonly used Passwords and Usernames as an input. Hydra discovered both the username and password of the local Windows Admin user, “denni” and “1234” respectively. The results of this Brute-Force attack are depicted in Figure 47.

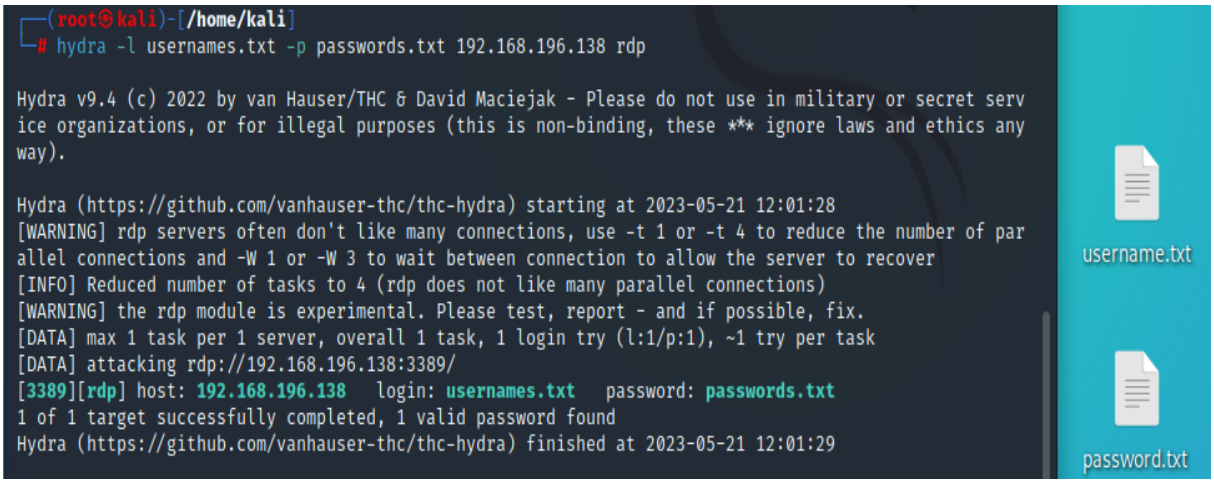


Figure 47: Hydra Brute Force Successful

To our surprise, HELK appears to have detected this Brute force attack as well in the form of Failed Login attempts as well as the Windows process **winlogon.exe** responsible for handling the user login and logout operations. As we can see in Figure 48, it even detected the *usernames.txt* file we used:

› May 21, 2023 @ 19:10:45.103	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	c:\windows\system32\svchost.exe	usernames.txt
› May 21, 2023 @ 19:08:54.563	96036718-71cc-4027-a538-d1587e0006a7_0 Windows Processes Suspicious Parent Directory	c:\windows\system32\svchost.exe	-
› May 21, 2023 @ 19:02:36.084	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public IP	c:\windows\system32\svchost.exe	usernames.txt
› May 21, 2023 @ 19:01:41.387	96036718-71cc-4027-a538-d1587e0006a7_0 Windows Processes Suspicious Parent Directory	c:\windows\system32\winlogon.exe	-
› May 21, 2023 @ 19:01:41.381	96036718-71cc-4027-a538-d1587e0006a7_0 Windows Processes Suspicious Parent Directory	c:\windows\system32\csrss.exe	-
› May 21, 2023 @ 18:57:40.371	f88e112a-21aa-44bd-9b01-6ee2a2bbbed1_0 Failed Logon From Public	c:\windows\system32\svchost.exe	usernames.txt

Figure 48: HELK detects brute force attack

Now to gain access to our victim host, we will use the command ***rdesktop -u denni 192.168.196.138*** from the Kali Command Line Interface as shown in Figure 49, in order to establish an RDP (Remote Desktop Protocol) Session,

Figure 49: Remote Desktop Session to Victim Host

After having gained Admin access, we then proceed to the Windows Command Line and type two of the most typical Commands used in the reconnaissance and information gathering phase of an ongoing attack, as shown in Figure 50:

- **Whoami**: The "whoami" command helps an attacker gather information about the current user context. By running this command, an attacker can determine the current user's username, domain, and group membership
- **Net Users**: The "net users" command allows an attacker to gather information about the user accounts configured on a target system. By executing this command, an attacker can obtain a list of user accounts, including privileged accounts, local accounts, and potentially domain accounts if the system is part of a domain.

```
C:\Users\denni>net users

User accounts for \\DESKTOP-TNC39G1

-------------------------------------------------------------------------------
Administrator            DefaultAccount           denni
Guest                    test                     WDAGUtilityAccount
The command completed successfully.


C:\Users\denni>whoami
desktop-tnc39g1\denni

C:\Users\denni>
```

Figure 50: Whoami/Net Users commands

Figure 51 clearly shows that HELK was able to detect the execution of both of these commands, and could also identify that they originated from a remote host.

System Owner/User Discovery	whoami	whoami.exe	whoami.exe
Remote System Discovery	c:\windows\system32\net1 users	net1.exe	net1.exe
Remote System Discovery	net users	net.exe	net.exe
Remote System Discovery	c:\windows\system32\net1 users	net1.exe	net1.exe
Remote System Discovery	net users	net.exe	net.exe
Remote System Discovery	c:\windows\system32\net1 user	net1.exe	net1.exe
Remote System Discovery	net user	net.exe	net.exe
System Owner/User Discovery	whoami	whoami.exe	whoami.exe

Figure 51: HELK detected both commands

Additionally, we tried executing other system-information related commands, which HELK detected with great success and raised the corresponding Alerts. The rules triggered by our use of these commands can be seen in Figure 52.

PowerShell	-	amsi.dll	tasklist.exe
Windows Management Instrumentation	-	wmiutils.dll	tasklist.exe
Process Discovery	tasklist	tasklist.exe	tasklist.exe
DLL Side-Loading	-	MpOAV.dll	systeminfo.exe
PowerShell	-	amsi.dll	systeminfo.exe
System Owner/User Discovery	systeminfo	sysinfo.exe	systeminfo.exe
System Network Configuration Discovery	-	-	nslookup.exe
System Network Configuration Discovery	-	-	nslookup.exe

Figure 52: Additional Discovery Commands detected by HELK

6.2.3 MALWARE INJECTION

Malware injection, also known as code injection, refers to the technique of injecting malicious code or malware into a legitimate process, application, or system component. The goal of malware injection is to execute the injected code within the target process or system, allowing the attacker to gain unauthorized access, control, or perform malicious actions.

The purpose of malware injection can vary depending on the attacker's objectives. It can be used for various malicious activities, including:

- Gaining unauthorized access or control over a system.
- Evading detection by antivirus or security solutions.
- Stealing sensitive information, such as passwords, credentials, or personal data.
- Executing additional malicious payloads or modules.
- Creating persistence mechanisms to maintain long-term access.
- Modifying or bypassing security mechanisms, such as firewalls or access controls.

To test how well HELK would detect attempts at malware injection we set up a localhost server in our Kali machine which would contain the malware we want to ship to our victim.

This process is shown in Figure 53. By taking advantage of the access we gained on the victim in the previous step of Brute-Force, we can download malicious software from the Kali Localhost server by using Powershell commands.

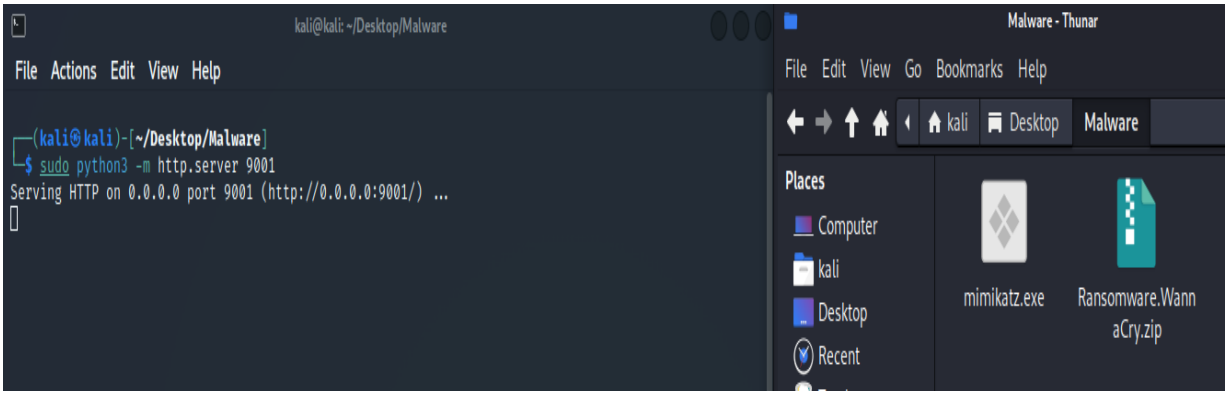


Figure 53: Setting up Server for Malware Injection in Kali

MIMIKATZ

Mimikatz[32] is a powerful post-exploitation tool that is commonly used in penetration testing and hacking scenarios. It is primarily designed to extract sensitive credentials and perform various credential-based attacks in Windows environments.Here are some key functions and capabilities of Mimikatz:

- **Credential Extraction**: Mimikatz can extract various types of credentials from Windows systems, including passwords, hashes, and Kerberos tickets.
- **Pass-the-Hash**: Mimikatz allows the attacker to leverage extracted password hashes to authenticate and gain unauthorized access to other systems without knowing the actual plaintext password.
- **Pass-the-Ticke**t: Similar to pass-the-hash, Mimikatz can also exploit Kerberos tickets obtained from compromised systems to impersonate users and gain unauthorized access to resources.
- **Golden Ticket**: With the help of Mimikatz, an attacker can generate a forged Kerberos "Golden Ticket" using the domain controller's KRBTGT account password.

After gaining elevated access to the Windows Host through RDP, we execute a Powershell command, shown in Figure 54, in order to download Mimikatz from the server hosted on the Kali.

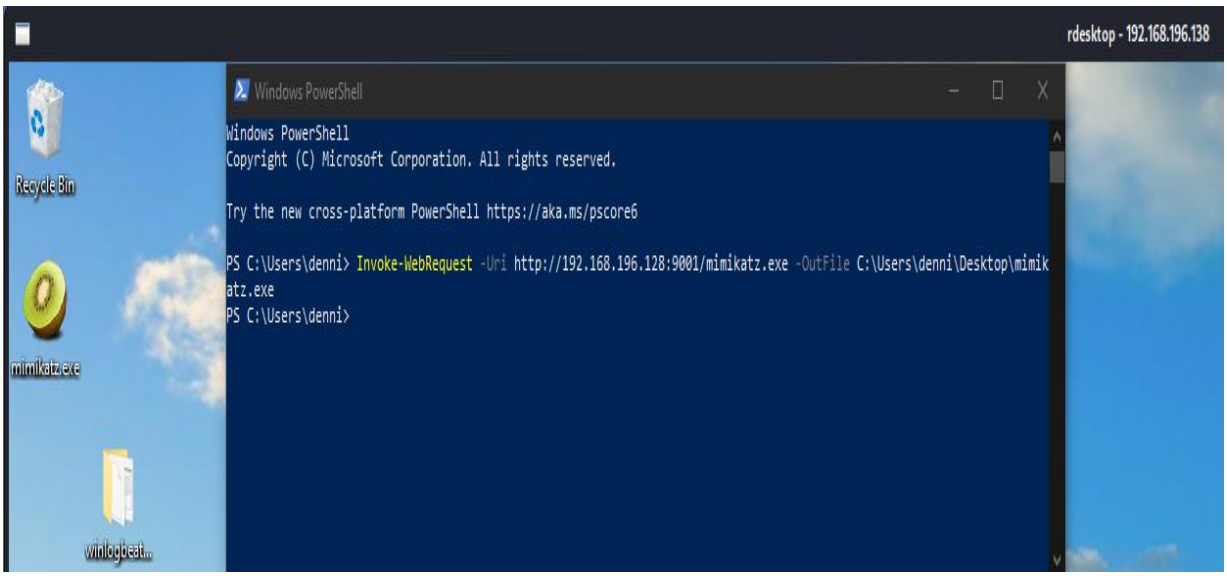


Figure 54: Downloading Mimikatz through Powershell

We proceeded to run Mimikatz and list all of the passwords stored in memory by using the command *sekurlsa::logonpasswords* as depicted in Figure 55.

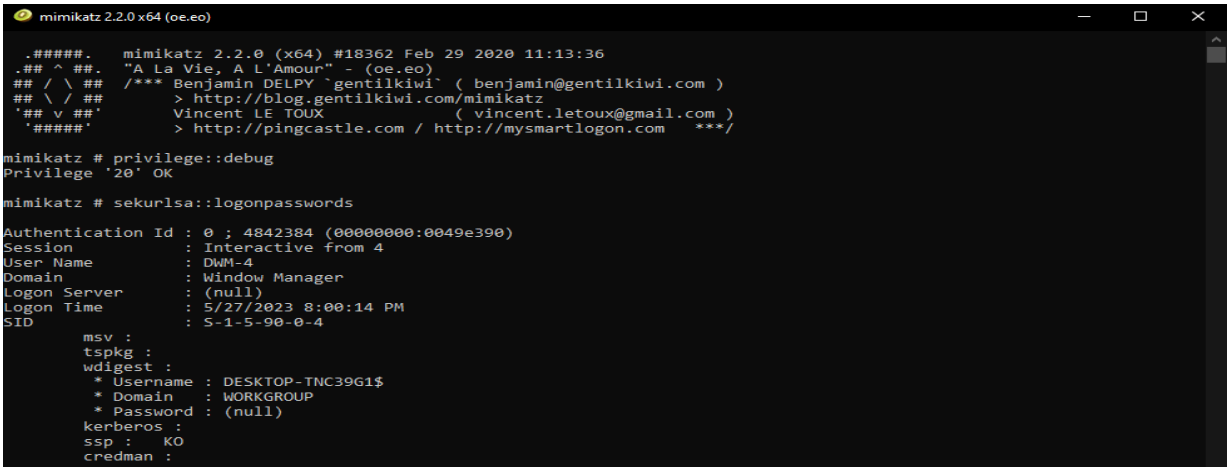


Figure 55: Running Powershell

In the context of HELK, we expect to receive an alert related to the process Lssas.exe. Mimikatz and lsass are connected in the context of credential dumping and password retrieval. Lsass.exe is a critical Windows system process responsible for security-related operations, including authentication and handling of user credentials. It stores sensitive information such as password hashes and plaintext passwords in its memory.

Mimikatz, as a tool, can interact with lsass.exe and extract various forms of credentials from its memory. It leverages security vulnerabilities and weaknesses to bypass security mechanisms and access the sensitive data stored in lsass.exe.

Indeed, HELK has detected both the execution of Mimikatz by the user as well as the dumping of credentials from the memory in the form of Lsass.exe as we can see in Figure 56.

Moreover, we can see the interactions between Mimikatz and Lsaas.exe can be seen in the log depicted in Figure 57.

〉	May 27, 2023 @ 20:35:22.815	Credential Dumping	T1003	mimikatz.exe	lsass.exe
〉	May 27, 2023 @ 20:35:1	Service Creation	T1543	dllhost.exe	-
〉	May 27, 2023 @ 20:35:09.065	-	-	lsass.exe	mimikatz.exe
〉	May 27, 2023 @ 20:35:08.875	-	-	conhost.exe	mimikatz.exe
〉	May 27, 2023 @ 20:35:08.813	-	-	csrss.exe	mimikatz.exe
〉	May 27, 2023 @ 20:35:08.728	User Execution	T1204	mimikatz.exe	-
〉	May 27, 2023 @ 20:35:08.717	-	-	csrss.exe	mimikatz.exe

Figure 56: Mimikatz detected by HELK

```
Process accessed:
RuleName: -
UtcTime: 2023-05-27 17:35:09.065
SourceProcessGUID: {47ebcb63-2d4a-6472-0c00-000000001e00}
SourceProcessId: 676
SourceThreadId: 6400
SourceImage: C:\Windows\system32\lsass.exe
TargetProcessGUID: {47ebcb63-3f4c-6472-e901-000000001e00}
TargetProcessId: 6008
TargetImage: C:\Users\denni\Desktop\mimikatz.exe
GrantedAccess: 0x1478
CallTrace: C:\Windows\SYSTEM32\ntdll.dll+9d524|C:\Windows\system32\lsasrv.dll+ac12
|C:\Windows\system32\lsasrv.dll+b030|C:\Windows\system32\lsasrv.dll+b3fb|C:\Window
s\SYSTEM32\SspiSrv.dll+1a82|C:\Windows\System32\RPCRT4.dll+7b253|C:\Windows\System
32\RPCRT4.dll+de77b|C:\Windows\System32\RPCRT4.dll+5bc7c|C:\Windows\System32\RPCRT
4.dll+58f88|C:\Windows\System32\RPCRT4.dll+3a1a6|C:\Windows\System32\RPCRT4.dll+39
af8|C:\Windows\System32\RPCRT4.dll+472bf|C:\Windows\System32\RPCRT4.dll+466c8|C:\W
indows\System32\RPCRT4.dll+45cb1|C:\Windows\System32\RPCRT4.dll+4571e|C:\Windows\S
ystem32\RPCRT4.dll+49e32|C:\Windows\SYSTEM32\ntdll.dll+20330|C:\Windows\SYSTEM32\n
tdll.dll+52f76|C:\Windows\System32\KERNEL32.DLL+17614|C:\Windows\SYSTEM32\ntdll.dl
l+526a1
SourceUser: NT AUTHORITY\SYSTEM
TargetUser: DESKTOP-TNC39G1\denni
```

Figure 57: Mimikatz interacting with lsass.exe

WannaCry

WannaCry[33] is a highly impactful ransomware that first emerged in May 2017. It spread rapidly across the globe, infecting hundreds of thousands of computers and causing significant disruptions. Its main characteristics are:

- Propagation: WannaCry spread rapidly by exploiting a vulnerability in the Windows operating system called EternalBlue.
- Encryption and Ransom: Once a system was infected, WannaCry encrypted files on the targeted computer and demanded a ransom in Bitcoin cryptocurrency for the decryption key.
- Impact: The WannaCry attack had a widespread impact, affecting hundreds of thousands of systems worldwide. It targeted various organizations, including healthcare institutions, government agencies, and businesses, causing disruptions to critical services and operations.
- Mitigation: The WannaCry attack highlighted the importance of promptly applying security patches and updates. Microsoft released an emergency patch to address the EternalBlue vulnerability following the attack.

The WannaCry ransomware attack served as a “wake-up call” for organizations and individuals to enhance their cybersecurity practices, including timely patching, implementing robust security measures, and maintaining backups to mitigate the impact of ransomware attacks.

As before with Mimikatz, we downloaded the executable file of Wannacry with a powershell command from Kali. By running the executable, our victim Host was effectively locked from use for further use while displaying the infamous Wannacry Ransom depicted in Figure 58.

Figure 58: Wannacry Ransom Message

Again, HELK was able to detect the Malware that we infected our host with. As we can see on the Figure 59 below, the **Timestomping** technique rule was triggered.
Ransomwares may employ timestomping techniques to manipulate or alter file timestamps on a compromised system. By changing the timestamps, they can make it more challenging for investigators to determine when the ransomware was initially deployed or to trace the activities leading up to the attack. Timestomping can act as a means to evade detection by security software or intrusion detection systems by altering the timestamps of ransomware executables or related files, to make them appear as legitimate or benign files that were created or modified in the past.

>	May 27, 2023 @ 21:05:55.567	-	-	@wanadecryptor@.exe	-	c:\users\denni\desktop\ransomware.wannacry\taskdata\tor\tor.exe
>	May 27, 2023 @ 21:05:55.566	Timestomp	T1070.006	@wanadecryptor@.exe	-	c:\users\denni\desktop\ransomware.wannacry\taskdata\tor\ssleay32.dll
>	May 27, 2023 @ 21:05:55.560	-	-	@wanadecryptor@.exe	-	c:\users\denni\desktop\ransomware.wannacry\taskdata\tor\ssleay32.dll
>	May 27, 2023 @ 21:05:5	Timestomp	T1070.006	@wanadecryptor@.exe	-	c:\users\denni\desktop\ransomware.wannacry\taskdata\tor\libssp-0.dll
>	May 27, 2023 @ 21:05:55.554	-	-	@wanadecryptor@.exe	-	c:\users\denni\desktop\ransomware.wannacry\taskdata\tor\libssp-0.dll
>	May 27, 2023 @ 21:05:55.536	Timestomp	T1070.006	@wanadecryptor@.exe	-	c:\users\denni\desktop\ransomware.wannacry\taskdata\tor\libevent_extra-2-0-5.dll

Figure 59: Wannacry detected by HELK

6.2.4 MAINTAINING PERSISTENCE

Maintaining persistence, in the context of cybersecurity, refers to the ability of an attacker or malicious software to maintain a presence or control over a compromised system or network for an extended period of time, even after initial access has been gained. It involves ensuring that unauthorized access or control can be retained across system reboots, software updates, and other changes.

Some of the characteristics of maintaining persistence are explained below:

- **Long-Term Access**: Once an attacker or malware gains initial access to a system, they aim to establish mechanisms that allow them to maintain ongoing access and control.

- **Counteracting Remediation:** Maintaining persistence involves implementing techniques or tools that allow the attacker to regain access even if the original entry point is discovered and remediated.

- **Techniques:** There are several techniques used to maintain persistence. Examples include modifying system configuration settings, adding unauthorised user accounts, installing rootkits or backdoors, leveraging scheduled tasks or cron jobs, modifying startup processes or system services etc.

SCHEDULED TASKS

Utilizing scheduled tasks for achieving persistence provides attackers with distinct advantages. By strategically timing the execution of tasks, they can minimize the risk of detection by running them during off-peak hours or periods of low system activity. Furthermore, scheduling these tasks with elevated privileges, such as administrator or root access, grants the attacker the ability to execute privileged actions without the need for additional authentication. This combination of timing and privilege allows attackers to maintain prolonged access to the system, evading detection and facilitating their malicious activities.

In the example pictured in Figure 60, we will configure a scheduled task that pings our Kali Linux Host every Sunday. This setup replicates a real-life scenario where a compromised system aims to verify communication with the attacker's system by initiating a ping request. This initial step ensures the establishment of a communication channel between the two systems, enabling the attacker to potentially deploy additional malicious payloads or malware in the future.

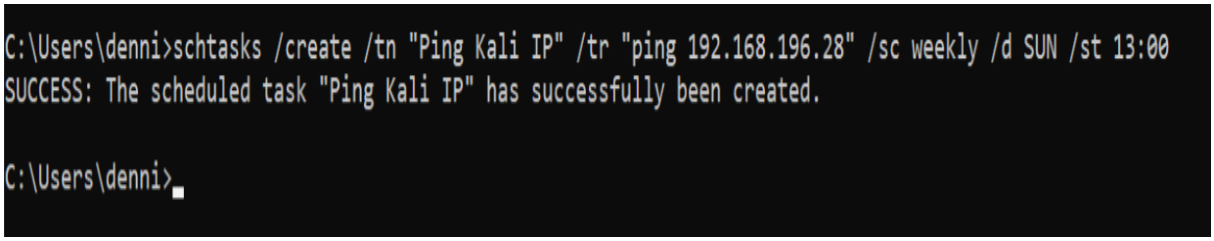


Figure 60: Scheduled Task Creation in CMD

HELK was able to detect the creation of this Scheduled task as we can see in Figure 61.

Time	rule_technique_name	process_path	process_name
Jun 5, 2023 @ 15:14:18.335	Scheduled Task	c:\windows\system32\schtasks.exe	schtasks.exe
Jun 5, 2023 @ 15:14:18.313	Command-Line Interface	c:\windows\system32\schtasks.exe	schtasks.exe
Jun 5, 2023 @ 15:12:49.879	Command-Line Interface	c:\windows\system32\schtasks.exe	schtasks.exe
Jun 5, 2023 @ 15:12:44.032	Command-Line Interface	c:\windows\system32\schtasks.exe	schtasks.exe

Figure 61: Scheduled Task Creation detected by HELK

LOLBins

LOLBins, also known as "Living Off the Land Binaries," refer to legitimate executables or tools present on a system, which attackers can exploit for malicious purposes. These binaries are typically bundled with operating systems, providing attackers with a means to bypass security measures by leveraging trusted and commonly used tools. In the context of maintaining persistence, attackers often utilize LOLBins to execute malicious code or achieve their objectives, all while camouflaging their activities within normal system operations. The usage of LOLBins can evade detection by traditional security monitoring since these binaries are already deemed trustworthy and are often whitelisted by security tools.

Notable examples of LOLBins include utilities like PowerShell, Windows Management Instrumentation (WMI), regsvr32, schtasks, among others. These utilities offer attackers various means to execute malicious scripts, download additional payloads, alter system configurations, create scheduled tasks, establish persistence mechanisms, and perform an array of harmful actions on a compromised system.

In this scenario, depicted in Figure 62, we will make use of the rundll32.exe utility and JavaScript in order to execute a Powershell command that downloads and runs a malicious script from the Kali Host (http://192.168.196.128:9001/)

```
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\denni>rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();new%20ActiveXObject("WScript.Sh
ell").Run("powershell -nop -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('http://192.168.196.128:9001/')
;"
```

Figure 62: Malicious Script Download from Kali Host

HELK detected this suspicious user activity in the command console and raised multiple alerts related to JavaScript and rundll32.exe as we can see in Figure 63 below:

>	Jun 5, 2023 @ 17:14:04.304	rundll32.exe	c:\windows\system32\rundll32.exe	rundll32.exe
>	Jun 5, 2023 @ 17:14:0	Service Creation	c:\windows\system32\cmd.exe	cmd.exe
>	Jun 5, 2023 @ 17:13:57.258	Service Creation	c:\windows\system32\cmd.exe	cmd.exe
>	Jun 5, 2023 @ 17:13:57.258	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2txyewy\searchapp.exe	searchapp.exe
>	Jun 5, 2023 @ 17:13:57.214	User Execution	c:\windows\system32\cmd.exe	cmd.exe
>	Jun 5, 2023 @ 17:13:57.213	Service Creation	c:\windows\explorer.exe	explorer.exe
>	Jun 5, 2023 @ 17:13:56.854	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2txyewy\searchapp.exe	searchapp.exe
>	Jun 5, 2023 @ 17:13:56.854	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2txyewy\searchapp.exe	searchapp.exe
>	Jun 5, 2023 @ 17:13:56.838	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2txyewy\searchapp.exe	searchapp.exe
>	Jun 5, 2023 @ 17:13:56.838	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2txyewy\searchapp.exe	searchapp.exe
>	Jun 5, 2023 @ 17:13:56.822	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2txyewy\searchapp.exe	searchapp.exe
>	Jun 5, 2023 @ 17:13:56.807	JavaScript	c:\windows\systemapps\microsoft.windows.search_cw5n1h2tx	searchapp.exe

Figure 63: Suspicious Activity detected by HELK

6.2.5 COVERING TRACKS

The "covering tracks" phase, refers to the activities carried out by an attacker to hide or remove any evidence of their presence or actions within a compromised system or network. The primary objective of this phase is to make it difficult for investigators or system administrators to detect and attribute the attack.

During the covering tracks phase, attackers typically perform various actions to erase their footprints, remove traces of their activities, and minimize the chance of being discovered.

Techniques like tampering with system logs, event logs, or audit trails to remove any indications of malicious activity are one of the most common ways to cover an attacker's tracks. This can involve deleting or modifying log entries, altering timestamps, or disabling logging mechanisms altogether.

In the context of our scenario, deleting Application and Security related logs to hide our actions is fairly straightforward. By executing the following 2 commands on the command prompt, we essentially make use of the *wevutil* utility in order to clear any Application and Security event logs.

- **wevtutil cl Application**
- **wevtutil cl Security**

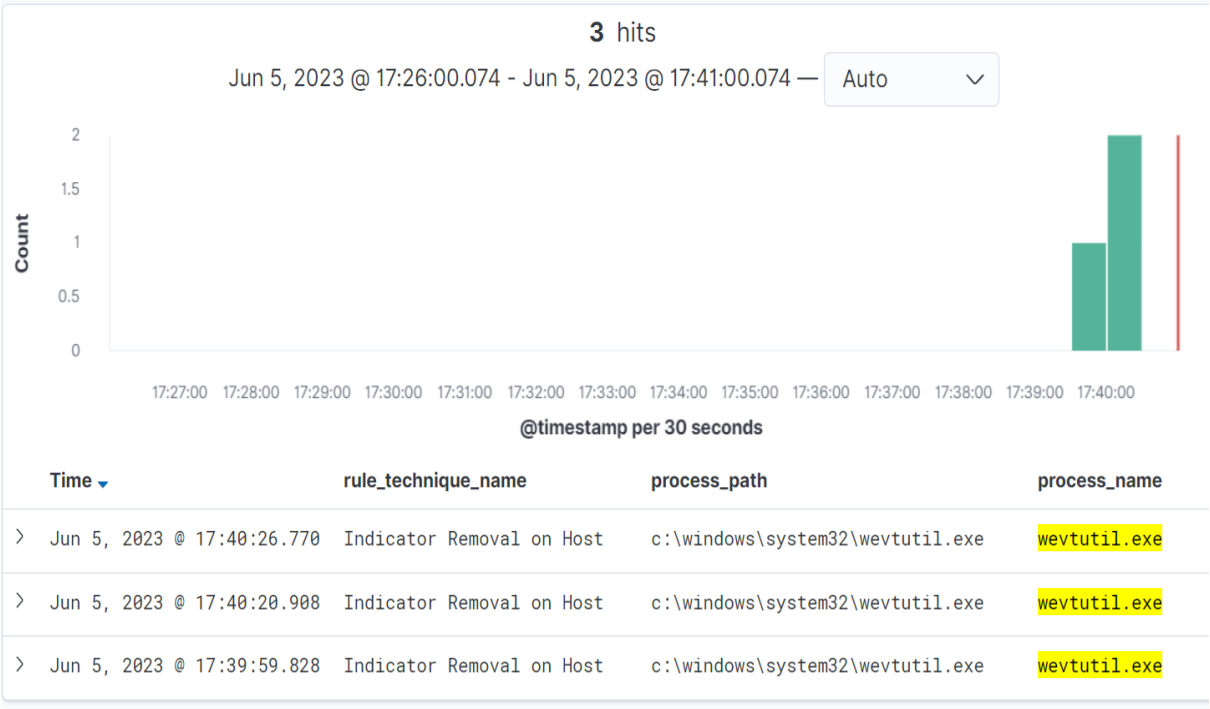


Figure 64: Log Deletion detected by HELK

The deletion of Application and Security event logs from our Victim Host, triggered the appropriate alerts in our Kibana interface, meaning that HELK was successful in detecting this suspicious activity as well. These alerts are visible in Figure 64.

Finally, having completed our tests on the HELK SIEM, we can review the results of all of the aforementioned techniques on Table 1 shown below:

Phase	**Technique**	**Outcome**
Reconnaissance	• NMAP Scan • Nessus Scan	Partially Detected Partially Detected
Brute Force	• Hydra	Detected
Malware Injection	• Mimikatz • WannaCry	Detected Detected
Maintaining Persistence	• Scheduled Tasks • LoLBins	Detected Detected
Covering Tracks	• Deleting System Logs	Detected

Table 1: Results HELK detection on different simulated attacks

CHAPTER 7: Experimental Results & Discussion

Throughout the entire process of establishing our HELK test environment and conducting a comprehensive evaluation of its capabilities, as detailed in Chapters 4 and 5, we have acquired a profound understanding of the platform's functionality, its advantages and limitations, as well as areas where enhancements could be implemented. These findings enable us to draw meaningful conclusions regarding the suitability of HELK as a SIEM solution for the SME sector.

7.1 BENEFITS

Small and medium-sized businesses (SMEs) face unique challenges when it comes to implementing effective cybersecurity measures. Limited resources, budget constraints, and a lack of dedicated security personnel often pose significant hurdles. However, the need for robust security solutions remains crucial. Below we will explore some of the reasons as to why HELK (The Hunting ELK) can be a viable SIEM solution for SMEs.

- **Open-Source and Cost-Effective**:
  HELK is an open-source platform that provides powerful security analytics capabilities at no licensing cost. For SMEs with limited budgets, this presents a cost-effective alternative to commercial SIEM solutions. By leveraging open-source technologies like Elasticsearch, Logstash, and Kibana, HELK allows organizations to allocate their resources more efficiently without compromising on security.

- **Customizability and Scalability:**
  HELK offers a high degree of customizability, allowing SMEs to tailor the solution to their specific needs and requirements. It supports the integration of various data sources, including logs from network devices, servers, and endpoints. This flexibility enables organizations to scale their security operations as they grow, adapting to evolving threats and compliance requirements.

- **Advanced Threat Detection and Hunting:**
  The core strength of HELK lies in its ability to perform advanced threat detection and hunting. By leveraging its robust data analysis and correlation capabilities, SMEs can identify and respond to security incidents more effectively. HELK enables proactive threat hunting, allowing security teams to detect and mitigate potential risks before they escalate. This proactive approach is particularly valuable for SMEs with limited incident response capabilities.

- **Enhanced Visibility and Compliance:**
  HELK provides a centralized and comprehensive view of security events and logs, offering improved visibility into an organization's security posture. This visibility helps identify anomalies, detect unauthorized access attempts, and monitor compliance with

regulatory requirements. SMEs can leverage HELK's visualization capabilities to generate meaningful reports and demonstrate compliance to auditors or stakeholders.

- **Community Support and Active Development:**
  HELK benefits from an active and supportive community of security professionals and developers. This community-driven approach ensures regular updates, bug fixes, and the availability of new features, however as we will explain below, community support can be lacking at times especially when it comes to production deployments and not testing environments. SMEs could rely on this vibrant community to seek assistance, share knowledge, and stay up-to-date with emerging security trends.

- **Compliance Considerations:**
  Small and medium-sized businesses often have compliance requirements to meet. HELK can assist in meeting these requirements by providing basic compliance features and the ability to configure log retention settings.

Overall, HELK's cost-effectiveness, flexibility and scalability, integration possibilities as well as customization options can make it a viable SIEM solution for small and medium-sized businesses seeking to enhance their security monitoring and incident response capabilities. As we will explain in further detail, however, there are limitations with this platform, especially when it comes to support and longevity.

7.2 LIMITATIONS

While HELK has its strengths and advantages as were detailed above, there are several factors that a SME might want to consider before choosing such a solution.

- **Complexity and Technical Expertise**:
  HELK, being an open-source platform, requires a certain level of technical expertise to set up, configure, and maintain. SMEs often have limited IT resources and may lack the necessary skills and knowledge to effectively deploy and manage a complex SIEM solution like HELK. This can result in increased implementation and maintenance costs or potential misconfiguration, diminishing the overall value of the solution. Installing HELK and configuring it to work in tandem with one Windows Log Source was a *mostly* straightforward process, SMEs however, might require up to hundreds of different devices and multiple different Log Source types like firewalls, IDS, Servers, etc. Configuring each different Log Source to ship logs to the HELK platform requires significant IT and networking proficiency. Moreover, maintaining and troubleshooting such deployments will require specialized staff which can ensure log delivery, system health and retention is maintained at all times.

- **Cost Considerations:**
  While HELK is open source and free to use, SMEs need to consider the total cost of ownership, including hardware requirements, storage, and maintenance. Implementing

and maintaining the necessary infrastructure for HELK can be financially burdensome for SMEs with limited budgets. Additionally, the costs associated with training staff or hiring specialized personnel for managing the HELK environment can further strain limited resources. While small deployments, like the one we tested extensively in chapter 5, are completely free of charge, we continuously had to delete old logs to maintain smooth operation. Log Retention for up to a year required for investigation of old incidents is a fundamental feature of SIEM Solutions, while requiring large amounts of storage capabilities and specialized server hardware or in other cases cloud storage subscriptions.

- **Scalability and Performance:**
  While setting up the HELK SIEM and configuring it to work with different types of log sources might be a cost-free affair, the more that are added, the more computing power and storage will have to be allocated for this task.
  HELK's performance and scalability then must be addressed when dealing with growing data volumes. As the number of security events and logs increases, the resources required by HELK may become inadequate, impacting system performance and responsiveness. SMEs may find it challenging to allocate the necessary resources to scale HELK effectively, potentially leading to degraded performance and delayed event analysis.

- **Lack of Vendor Support:**
  HELK is an open-source project maintained by the community, which means there is no official vendor providing dedicated support and assistance. SIEM deployments often rely on vendor support to address technical issues, troubleshoot problems which will surely arise, and receive timely updates and patches. Moreover, reputable and established SIEM vendors provide ever growing support for integration of their SIEM products with hundreds of security solutions like Firewalls, IDS's, Antiviruses and many others in the form of on-boarding guides as well as real time support. Without this vendor support, SMEs may face challenges in resolving issues, staying up to date with the latest security enhancements as well as on-boarding and accessing expert guidance when needed.

- **Time and Resource Constraints:**
  SMEs typically have limited staff and time available to devote to security operations, let alone maintaining a dedicated in-house security team. HELK, with its extensive customization options and fine-grained analysis capabilities, requires significant time and effort to configure and fine-tune to match specific business requirements. For example, throughout our testing scenario, we had to constantly fiddle with rules which were creating hundreds of false positive alerts which disrupted our visibility.
  SMEs may struggle to allocate the necessary resources, time and staff to effectively leverage HELK's capabilities such as creating custom SIEM rules that address their security needs, on-boarding new devices and expanding on its capabilities, leading to underutilization or improper configuration.

While HELK offers advanced features and extensive customization options for security event monitoring and analysis, its complexity, post deployment cost considerations, scalability limitations and lack of vendor support can hinder its prospects as a viable SIEM solution for small and medium-sized businesses. SMEs require SIEM solutions that are easy to deploy, cost-effective, scalable, and well-supported to meet their specific security needs without straining their resources and requiring a large number of specialized staff.

7.3 HELK COMPARED TO OTHER SIEM PRODUCTS

Selecting the right Security Information and Event Management (SIEM) solution is crucial for organizations to effectively monitor and analyze security events and protect their digital assets. In this comparison, we will evaluate the HELK SIEM, alongside other well-known commercial SIEM products like QRadar, Splunk, ArchSight and others.

The goal of this comparison is to provide insights into the features, capabilities, and considerations associated with each SIEM solution. While HELK SIEM offers a cost-effective and flexible option, commercial SIEM products bring additional functionalities, support, and integration options. Understanding the strengths and weaknesses of each solution will help organizations make informed decisions based on their specific needs, resources, and objectives.

Comparison Overview:

The comparison encompasses several categories, such as Features, Scalability, Ease of Use, Integration Capabilities, Support, and Cost. Each category highlights key factors that organizations typically consider when evaluating SIEM solutions. The comparison will first be presented in an easy-to-read table format depicted in Table 2, and then will be expanded for each category. It's important to note that the information presented is a general overview and may vary based on specific versions, configurations, and individual business needs. Organizations should conduct their own research, evaluate demos, and engage in discussions with vendors to gather more detailed information before making a final decision. The information shown below was validated primarily within a corporate environment and is by no means a definitive and comprehensive list. Extensive research and evaluation must be made before choosing a SIEM for a production environment.

CATEGORY	HELK	QRADAR	Splunk	ArcSight	LogRhythm
Cost	Open-Source, Free SIEM Solution	Commercial solution with licensing fees	Commercial solution with licensing fees	Commercial solution with licensing fees	Commercial solution with licensing fees

Deployment	On Premises deployment	On Premises or Cloud deployment options	On Premises or Cloud deployment options	On Premises or Cloud deployment options	On Premises or Cloud deployment options
Log Collection	Collects logs via Beats for data ingestion	Support various data sources for log collection	Support various data sources for log collection	Support various data sources for log collection	Support various data sources for log collection
Data Analysis	Elasticsearch and Kibana for log data analysis	QRadar SIEM engine for log data analysis	Splunk Search Processing Language (SPL)	ArcSight ESM analytics engine	AI engine for log data analysis
Threat Intel	Limited built-in Threat Intel capabilities	Robust Threat Intel capabilities through first party apps	Robust Threat Intel capabilities	Integration with 3rd party Threat Intel Feeds	Integration with 3rd party Threat Intel Feeds
Scalability	Scalable Architecture for large volumes of data	Scalable to handle enterprise-scale log data	Scalable to handle enterprise-scale log data	Scalable to handle enterprise-scale log data	Scalable to handle enterprise-scale log data
Compliance	Basic compliance features for meeting regulatory standards	Strong compliance features for meeting regulatory standards	Compliance modules for meeting regulatory requirements	Compliance modules for meeting regulatory requirements	Compliance modules for meeting regulatory requirements
Alerting	Basic Alerting Features for generating notifications	Advanced alerting capabilities with customizable rules	Advanced alerting capabilities with customizable rules	Advanced alerting capabilities with customizable rules	Advanced alerting capabilities with customizable rules
User Interface	Kibana UI for log data exploration and analysis	QRadar console for user interaction and monitoring	Splunk Web for user interaction and monitoring	ArcSight console for user interaction and monitoring	LogRhythm console for user interaction and monitoring
Vendor Support	Community support through forums and online resources	Vendor support available with various support options	Vendor support available with various support options	Vendor support available with various support options	Vendor support available with various support options
Integrations	Limited integrations with other tools and platforms	Extensive integration with various security and IT tools	Extensive integration with various security and IT tools	Extensive integration with various security and IT tools	Extensive integration with various security and IT tools

Machine Learning	No built-in ML capabilities	Built-in ML capabilities for advanced threat detection	Built-in ML capabilities for advanced threat detection	No built-in ML capabilities	Built-in ML capabilities for advanced threat detection
Data Privacy	Self-Managed data privacy and compliance considerations	Compliance features for data privacy regulations	Compliance features for data privacy regulations	Compliance features for data privacy regulations	Compliance features for data privacy regulations

Table 2: High Level Comparison between HELK and other SIEM Tools

Cost: HELK is open-source and free to use, making it highly cost-effective for small and medium-sized businesses (SMBs). However, the total cost of ownership (TCO) may vary based on infrastructure and customization needs.
Commercial SIEM tools like QRadar, Splunk, Arcsight, and Sentinel typically involve significant licensing, operational, and maintenance costs, making them less budget-friendly for SMBs.

Deployment: HELK's deployment can be more complex and time-consuming as the scale of the infrastructure in question increases, as it requires setting up and configuring multiple open-source components such as Elasticsearch, Logstash, Kibana, and Beats.
Commercial SIEM solutions often provide more streamlined deployment processes, professional support, and user-friendly installation wizards, reducing deployment complexity.

Log Retention: HELK offers customizable log retention options, allowing users to define their log retention policies. However, the scalability and retention period depend on the underlying hardware and storage capacity.
Commercial SIEM solutions typically provide more robust log retention capabilities with scalable storage options, ensuring long-term data retention and compliance with regulatory requirements.

Data Analysis: HELK's data analysis capabilities are based on the ELK Stack (Elasticsearch, Logstash, Kibana), offering robust searching, indexing, and visualization features. Users can create custom dashboards and visualizations tailored to their needs.
Commercial SIEMs often have advanced analytics and correlation engines that enable more comprehensive data analysis, threat detection, and incident response, with predefined use-case templates.

Threat Intel: HELK can integrate with threat intelligence feeds, but the availability and quality of feeds may vary. Users can manually configure threat intelligence sources and indicators of compromise (IOCs).

Commercial SIEMs often come with built-in or easily integrated threat intelligence features from reputable sources, simplifying the threat detection and enrichment process.

Scalability: HELK's scalability depends on the capacity of the underlying hardware, infrastructure, and Elasticsearch cluster configuration. Users can scale horizontally by adding more Elasticsearch nodes.
Commercial SIEM solutions are designed with scalability in mind, offering options for distributed deployments, load balancing, and automatic scaling to handle growing data volumes.

Compliance: HELK can be configured to support compliance requirements, but achieving compliance may require more manual configuration and customization based on specific regulatory standards.
Commercial SIEM tools often come with predefined compliance templates, reporting features, and audit trails, simplifying compliance efforts and reporting for SMBs.

Alerting: HELK offers alerting capabilities through Elasticsearch's Watcher feature, allowing users to create custom alerting rules. However, setting up and fine-tuning alerts may require additional effort.
Commercial SIEMs provide comprehensive alerting with predefined rules, real-time alert notifications, and advanced alert correlation, enabling faster incident detection and response.

User Interface: HELK's user interface is Kibana-based, offering visualization, dashboards, and search capabilities. While it provides flexibility, customization may be needed to create tailored dashboards.
Commercial SIEM tools often have user-friendly, customizable interfaces with predefined dashboards and reports, making it easier for SMBs to get started without extensive customization.

Vendor Support: HELK relies on community support, forums, and documentation, lacking official vendor support. Users may need to rely on the open-source community for assistance.
Commercial SIEM vendors offer official support packages with guaranteed assistance, service-level agreements (SLAs), and access to dedicated support teams, ensuring prompt issue resolution.

Integrations: HELK supports various data sources and integrations through Beats and Logstash, but configuring and maintaining integrations may require custom scripts or configurations.
Commercial SIEM solutions often come with extensive integrations and connectors for popular platforms, simplifying data ingestion and integration efforts for SMBs.

Machine Learning: HELK can leverage machine learning capabilities through the ELK Stack, but advanced machine learning models for anomaly detection may require additional development and tuning.

Commercial SIEMs often include built-in machine learning algorithms for anomaly detection, reducing the complexity of implementing ML-based threat detection.

Data Privacy: HELK's data privacy controls depend on how it's configured and secured by the user. Users must implement proper access controls, encryption, and data masking to protect sensitive information. Commercial SIEM solutions often provide more granular data privacy controls, including role-based access control (RBAC), data anonymization, and encryption, ensuring compliance with data privacy regulations.

7.4 PROPOSED IMPROVEMENTS

In the fast-paced world of cybersecurity, small and medium-sized businesses (SMEs) face unique challenges in defending against evolving threats. Deploying a robust Security Information and Event Management (SIEM) solution is essential for effective threat detection and response. HELK can be a powerful open-source platform that offers a solid foundation for SMEs. While HELK has its strengths, there are certain considerations and challenges when deploying it in small and medium-sized businesses. This essay aims to explore various methods and strategies to improve the capabilities of HELK, making it a viable and effective SIEM solution for SMEs.

1. Streamlined Deployment and Configuration:

SMBs often have limited resources and technical expertise, making a simplified deployment and configuration process essential. This can involve developing user-friendly installation scripts that automate the setup, providing clear documentation with step-by-step instructions, and offering automated configuration options,developing comprehensive installation guides, providing pre-configured virtual machine images, and creating user-friendly interfaces or wizards that guide users through the setup process. By reducing the complexity and time required for initial deployment, SMBs can quickly adopt and start leveraging the benefits of HELK.

2. Customized Use Case Development:

To maximize the value of HELK for SMEs, it's important to provide pre-built use cases and rules that align with their specific industry and security needs. These use cases can cover common attack vectors, compliance requirements, and specific threats relevant to the SMB environment. Additionally, guidance on customizing and creating new use cases and rules tailored to their unique circumstances will enable SMEs to effectively leverage HELK's capabilities in detecting and responding to threats specific to their business.

3. Integration with SME-Friendly Data Sources:

SMEs rely on a range of security devices, applications, and platforms to protect their environment. Enhancing HELK's integration capabilities to support commonly used data sources in SME environments is crucial. This can include integrating with firewalls, endpoint protection solutions, cloud services, authentication systems, and

other relevant data sources. By expanding the list of supported integrations, HELK can provide comprehensive visibility into security events across the SME infrastructure.

4. Automated Threat Hunting and Alerting:
Given the limited resources and dedicated security teams in SMEs, automation plays a vital role in threat detection and response. Enhancements can focus on developing automated threat hunting capabilities within HELK, leveraging machine learning algorithms, behavioral analytics, and threat intelligence feeds. This enables HELK to proactively identify suspicious activities, detect anomalies, and generate timely alerts for potential threats. By automating these processes, SMEs can effectively mitigate risks without the need for constant manual monitoring.

5. Resource Optimization:
HELK's resource requirements can be optimized to accommodate the limitations of SMEs. This can involve optimizing the default configuration to strike a balance between performance and resource utilization, providing guidance on hardware specifications for different scale deployments, and implementing mechanisms for data retention and archiving to manage storage requirements effectively. By providing recommendations and best practices, SMEs can ensure the proper utilization of resources without compromising the effectiveness of HELK.

6. Enhanced Visualization and Reporting:
HELK's visualization tool, Kibana, serves as a powerful interface for analyzing security data. Enhancing its visualization capabilities to cater to SMEs needs is important. This can involve developing custom dashboards, reports, and visual analytics that provide SMEs with concise and actionable insights. Customizable reports and user-friendly templates can assist SMEs in meeting compliance requirements and conveying security insights to stakeholders effectively.

7. Cloud Integration and Scalability:
As SMEs increasingly adopt cloud services, HELK should enhance its integration and scalability capabilities. This includes seamless integration with popular cloud storage providers for efficient log management, allowing SMEs to store and access their security logs in the cloud. Additionally, HELK should support cloud-based elastic scaling to accommodate growing data volumes and ensure optimal performance. Integration with cloud-native security services can further enhance threat detection capabilities and provide SMEs with a holistic security solution.

8. Vendor-Support:
To address concerns about vendor reliability and long-term sustainability, a hybrid model can be considered, where commercial vendors provide enhanced versions of HELK along with dedicated support and maintenance services. This ensures SMEs have access to reliable technical assistance, security updates, and feature enhancements, while still benefiting from the core open-source nature of HELK. Vendor-supported

offerings can instill confidence in SMEs regarding the stability and future development of the SIEM solution.

CHAPTER 8: CONCLUSIONS AND RELATED WORK

8.1 RELATED WORK

In the landscape of cybersecurity, the quest for robust solutions to fortify systems against evolving threats is perpetual. This sub-chapter delves into two noteworthy works that resonate with the core theme of this thesis—evaluating the feasibility of the Hunting ELK as a SIEM open-source system. These works provide valuable insights into the broader realm of threat hunting, log analysis, and the strategic deployment of SIEM tools.

The initial study named "A Threat Hunting Framework for Industrial Control Systems"[34] introduces the Industrial Control System Threat Hunting Framework (ICS-THF), a comprehensive strategy designed to detect cyber threats in the early stages of attack life cycles within Industrial Control Systems (ICS). Utilizing the MITRE ATT&CK Matrix and a Diamond model for intrusion analysis, ICS-THF operates through three key stages: threat hunting triggers, threat hunting, and cyber threat intelligence. The preparation phase stresses the importance of a robust data collection and retention strategy, endorsing low-cost tools such as the Hunting Elastic Stack (HELK). This integration showcases HELK's prowess in monitoring network traffic, detecting anomalies, and correlating events. Threat hunting triggers, diverse in nature, can be initiated by cyber threat intelligence, news articles, third-party notifications, or data anomalies detected by security analysts. The paper underlines the integration of the MITRE ATT&CK Matrix and HELK for understanding adversarial Tactics, Techniques, and Procedures (TTPs), emphasizing the potential of HELK in detecting ATT&CK TTPs within an ICS network. Furthermore, the incorporation of Cyber Threat Intelligence (CTI) sharing through MISP and a call for future evaluation of CTI data round out the work.

Concerning log analysis automation, another work named "Automation of Log Analysis Using the Hunting ELK Stack"[35] delves into the core of log analysis and the role of HELK in streamlining this process. Positioned as a potent open-source utility stack, HELK automates log analysis and enhances security through machine learning techniques. Operating within the challenging landscape of log analysis, HELK emerges as a solution that automates log file processing. This integration, encompassing Elasticsearch, Logstash, and Kibana, seamlessly manages log data using "beats" for the efficient delivery of structured information. Integral components such as Sysmon, Kafka, and Suricata further amplify HELK's capabilities. The work underscores HELK's role in training threat hunting teams, as evidenced by a simulated test network where HELK and Suricata are strategically deployed. Simulated scenarios, mirroring real-world cyber threats, serve as a hands-on training ground for threat hunting teams. Results demonstrate the success of event capture and analysis in Kibana, highlighting HELK's prowess in monitoring network security events. The work concludes by positioning HELK as a dynamic asset in managing network log files and fortifying threat hunting capabilities. Beyond traditional log analysis, HELK emerges as a pivotal training platform, preparing teams to navigate the evolving landscape of cybersecurity threats.

8.2 CONCLUSIONS

In the contemporary landscape of escalating cyber threats, where the interconnectivity of digital infrastructures demands robust cybersecurity, the question of effective Security Information and Event Management (SIEM) solutions becomes paramount. Small and medium-sized enterprises (SMEs), often constrained by budget considerations, face challenges in adopting traditional commercial SIEM tools like QRadar and Splunk, which can impose significant financial burdens. This thesis meticulously explored the feasibility of SMEs embracing an open-source SIEM alternative, Hunting ELK (HELK). The study navigated through every aspect of HELK deployment, providing a comprehensive roadmap tailored to businesses with diverse technical proficiencies. The intricate process of configuring a Windows Host to seamlessly transmit logs to HELK is elucidated, ensuring accessibility for varying SME environments.

Moreover, the research conducted a detailed analysis of HELK's inner workings and subjected it to a series of carefully orchestrated cyberattacks to evaluate its efficacy in threat detection and mitigation. The findings illuminated HELK's strengths and weaknesses, offering valuable insights into potential enhancements. By traversing the complexities of deployment, testing performance, and scrutinizing practicality, this study can assist in empowering SMEs to make informed decisions about their cybersecurity strategies.

In conclusion, HELK can emerge as a promising SIEM solution for SMEs, yet significant enhancements are required to maximize its threat hunting potentials and compliance capabilities. Proactive measures to address deployment challenges can position HELK as a cost-effective and potent cybersecurity tool. With these improvements, HELK stands ready to deliver heightened security intelligence and robust defense mechanisms, enabling SMEs to fortify their security posture against modern threats. This study, combining hands-on experimentation with a thorough evaluation of HELK's capabilities, serves as a testament to the potential of open-source SIEM solutions in meeting the cybersecurity needs of SMEs while mitigating financial constraints.

REFERENCES

[1] "Cybersecurity for SMEs - Challenges and Recommendations," ENISA. Accessed: Oct. 15, 2023. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-report-cybersecurity-for-smes

[2] "First incident of cyber-espionage," *Guinness World Records*. https://www.guinnessworldrecords.com/world-records/612868-first-incident-of-cyber-espionage (accessed Oct. 19, 2022).

[3] "IoT connected devices worldwide 2019-2030," *Statista*. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed Oct. 19, 2022).

[4] M. Antonakakis *et al.*, "Understanding the Mirai Botnet," p. 19.

[5] "Kaspersky Security Bulletin 2021." https://securelist.com/ksb-2021/ (accessed Oct. 19, 2022).

[6] "Cost of a data breach in the U.S. 2022," *Statista*. https://www.statista.com/statistics/273575/us-average-cost-incurred-by-a-data-breach/ (accessed Oct. 19, 2022).

[7] T. Kellermann, "'Modern Bank Heists' Threat Report Finds Dramatic Increase in Cyberattacks Against Financial Institutions Amid COVID-19," VMware Security Blog. Accessed: Sep. 23, 2023. [Online]. Available: https://blogs.vmware.com/security/2020/05/modern-bank-heists-threat-report-finds-dramatic-increase-in-cyberattacks-against-financial-institutions-amid-covid-19.html

[8] M. Mazumder and A. Sobhan, "The Spillover Effect of the Bangladesh Bank Cyber Heist on Banks' Cyber Risk Disclosures in Bangladesh." Rochester, NY, Apr. 04, 2020. Accessed: Oct. 19, 2022. [Online]. Available: https://papers.ssrn.com/abstract=3771379

[9] "A SANS 2021 Survey: Security Operations Center (SOC) | SANS Institute." https://www.sans.org/white-papers/sans-2021-survey-security-operations-center-soc/ (accessed Oct. 19, 2022).

[10] "Personal Data Breach | European Data Protection Supervisor." https://edps.europa.eu/data-protection/our-role-supervisor/personal-data-breach_en (accessed Oct. 19, 2022).

[11] "Cyber Resilient Business | Accenture." https://www.accenture.com/us-en/insights/cyber-security-index (accessed Jan. 18, 2023).

[12] "IBM Security X-Force Threat Intelligence Index 2023 | IBM." https://www.ibm.com/reports/threat-intelligence#malware (accessed Aug. 05, 2023).

[13] "SolarWinds Cyberattack Demands Significant Federal and Private-Sector Response (infographic) | U.S. GAO." https://www.gao.gov/blog/solarwinds-cyberattack-demands-significant-federal-and-private-sector-response-infographic (accessed Aug. 05, 2023).

[14] "Cyber Kill Chain, MITRE ATT&CK, and Purple Team | SANS Institute." https://www.sans.org/blog/cyber-kill-chain-mitre-attack-purple-team/ (accessed Nov. 07, 2022).

[15] "Tactics - Enterprise | MITRE ATT&CK®." https://attack.mitre.org/tactics/enterprise/ (accessed Nov. 07, 2022).

[16] "trustwave-budgeting-for-SIEM.pdf." Accessed: Nov. 07, 2022. [Online]. Available: https://www.infopoint-security.de/media/trustwave-budgeting-for-SIEM.pdf

[17] R. Rodriguez, "HELK." Nov. 06, 2022. Accessed: Nov. 07, 2022. [Online]. Available: https://github.com/Cyb3rWard0g/HELK

[18] "What is Elasticsearch?," *Elastic*. https://www.elastic.co/what-is/elasticsearch (accessed Nov. 07, 2022).

[19] "What is Kibana?," *Elastic*. https://www.elastic.co/what-is/kibana (accessed Aug. 05,

2023).
[20] "How Logstash Works | Logstash Reference [8.9] | Elastic." https://www.elastic.co/guide/en/logstash/current/pipeline.html (accessed Aug. 05, 2023).
[21] "Beats: Data Shippers for Elasticsearch," *Elastic*. https://www.elastic.co/beats (accessed Aug. 05, 2023).
[22] "Apache Kafka," *Apache Kafka*. https://kafka.apache.org/ (accessed Aug. 05, 2023).
[23] "ElastAlert - Easy & Flexible Alerting With Elasticsearch — ElastAlert 0.0.1 documentation." https://elastalert.readthedocs.io/en/latest/ (accessed Aug. 05, 2023).
[24] "Command and Scripting Interpreter: PowerShell, Sub-technique T1059.001 - Enterprise | MITRE ATT&CK®." https://attack.mitre.org/techniques/T1059/001/ (accessed Jan. 14, 2023).
[25] markruss, "Sysmon - Sysinternals," Jan. 25, 2023. https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon (accessed Jan. 27, 2023).
[26] "Winlogbeat quick start: installation and configuration | Winlogbeat Reference [8.6] | Elastic." https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-installation-configuration.html (accessed Jan. 29, 2023).
[27] D. Stuttard and M. Pinto, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. John Wiley & Sons, 2011.
[28] "Nmap: the Network Mapper - Free Security Scanner." https://nmap.org/ (accessed Aug. 05, 2023).
[29] "Sigma." Sigma, Apr. 24, 2023. Accessed: Apr. 24, 2023. [Online]. Available: https://github.com/SigmaHQ/sigma
[30] "Tenable Multiproduct." https://www.tenable.com/lp/campaigns/22/try-nessus-multiprdct/free-trial/?utm_campaign=gs-{11596512905}-{118315026852}-{537515899088}_00026642_fy23&utm_promoter=tenable-hv-brand-00026642&utm_source=google&utm_term=nessus&utm_medium=cpc&utm_geo=emea&gclid=CjwKCAjw5remBhBiEiwAxL2M96rSNJbPoNuPPBduSMLchWrg1PpJ7w4leE3ZdjqeEHif4JdJII5cdxoCBWIQAvD_BwE (accessed Aug. 05, 2023).
[31] "hydra | Kali Linux Tools," *Kali Linux*. https://www.kali.org/tools/hydra/ (accessed Aug. 05, 2023).
[32] "Mimikatz, Software S0002 | MITRE ATT&CK®." https://attack.mitre.org/software/S0002/ (accessed May 27, 2023).
[33] "What is a WannaCry Ransomware Attack?," *Fortinet*. https://www.fortinet.com/resources/cyberglossary/wannacry-ransomware-attack (accessed Aug. 05, 2023).
[34] Z. Jadidi and Y. Lu, "A Threat Hunting Framework for Industrial Control Systems," IEEE Access, vol. 9, pp. 164118–164130, 2021, doi: 10.1109/ACCESS.2021.3133260.
[35] M. A. Stan, "Automation of Log Analysis Using the Hunting ELK Stack," 2021.

ABBREVIATIONS

SIEM	Security Information and Event Management
SOC	Security Operations Centre
UI	User Interface
DDoS	Distributed Denial of Service
SANS	SysAdmin, Audit, Network and Security
GDPR	General Data Protection Regulation
APT	Advanced Persistent Threat
EDR	Endpoint Detection and Response
IDS	Intrusion Detection System
API	Application Programming Interface
CPU	Central Processing Unit
RAM	Random Access Memory
VM	Virtual Machine
SME	Small & Medium Enterprises
FTP	File Transfer Protocol
OS	Operating System
RDP	Remote Desktop Protocol
CVE	Common Vulnerabilities & Exposures
WMI	Windows Management Instrumentation
IT	Information Technology
ML	Machine Learning
LOLBin	Living-off-the-Land Bins
SLA	Service Level Agreement
ICS-THF	Industrial Control System Threat Hunting Framework
CTI	Cyber Threat Intelligence