UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Event Correlation in Ciências

Nikhil Ravin Tulcidas

Mestrado em Segurança Informática

Trabalho de Projeto orientado por:
Prof. Doutor Hugo Alexandre Tavares Miranda

2024

Acknowledgments

I would like to express my sincere gratitude to Pedro Botas and Professor Hugo Miranda for allowing me to create this project, continuously supporting my choices, and teaching me over its course. Thank you to Susana Pereira for incentivizing and integrating me into CIÊNCIAS ULisboa's other cybersecurity components, allowing me to have the opportunity to improve my knowledge and gain experience. Additionally, a thank you to Paulo Bastos for helping me troubleshoot issues and permitting me to install agents into his systems for the project to gather real, rather than simulated, data.

My parents and sister have my heartfelt gratitude for their unwavering support and encouragement throughout my pursuit of a Master's degree.

A very special thank you to the friends made along the way: Beniamin Tott, Miguel Matias, Rodrigo Guerreiro, and Tiago Muxagata who have had to put up with me over the years.

Thank you to my friends I met in Ciências, especially Rodrigo Branco, António Silvestre, and Francisco Amaro for supporting and holding me to a high standard. Your jokes have constantly pushed me to continue improving myself. An extra thank you to Duarte Santos, one of the hardest-working individuals I have met, who motivated everyone around him, including me, to do better.

And lastly, thank you to the video game industry and its developers for creating worlds, stories, and characters that entertained me so much. They ignited the spark that gave me my never-ending interest in IT.

To those who had my back.

Resumo

A Faculdade de Ciências da Universidade de Lisboa (CIÊNCIAS ULisboa) é uma instituição de ensino superior pública, com mais de 700 funcionários e 5.600 estudantes. Anualmente, cerca de 1.000 alunos concluem os seus cursos na CIÊNCIAS ULisboa, enquanto outros 1.000 novos estudantes ingressam. Este projeto é, por isso, implementado num ambiente único onde não é possível controlar as ações que podem ser realizadas pela maioria dos utilizadores, que são estudantes. Monitorizar e gerir meticulosamente mais de 400 pontos de acesso, uma vasta gama de mais de 10,000 dispositivos (muitos dos quais não pertencem ou não são geridos por CIÊNCIAS ULisboa), numerosos servidores *web* e múltiplos controladores de domínio com diversas funções está além das possibilidades da instituição.

Uma grande quantidade de eventos é produzida e registada a cada segundo pela infraestrutura de Tecnologia da Informação (IT) de CIÊNCIAS ULisboa. Estes eventos são potencialmente relevantes para a cibersegurança, pois contêm informações que permitem compreender o que aconteceu ou está a decorrer. Exemplos de informações incluem endereços IP e o *input* utilizado, que pode conter conteúdo malicioso destinado a explorar vulnerabilidades. Estas informações são úteis para compreender de onde vêm os ataques e que técnicas são utilizadas pelos adversários de forma a mitigar tentativas de explorações futuras. No entanto, não é viável inspecionar manualmente estas informações. Por isso, utilizam-se ferramentas como os *Intrusion Detection Systems* (IDS) para automatizar a análise e alertar sobre possíveis atividades maliciosas.

Um problema que advém da utilização de ferramentas de segurança é que, tradicionalmente, não utilizam as informações ou eventos gerados por outras ferramentas de segurança, nem dos eventos gerados pela infraestrutura de IT. Ao correlacionar as informações dos eventos das ferramentas de segurança com outras fontes, como os eventos gerados pelos serviços em execução, é possível determinar se um ataque detetado é legítimo ou um falso positivo, qual foi o alvo, se afetou o serviço e até interromper o ataque antes que cause mais danos.

Em CIÊNCIAS ULisboa, o *Snort* é utilizado como o IDS principal. Os alertas gerados pelo *Snort* fornecem informações como endereços IP, portas e algum contexto sobre o que foi feito ou tentado. Para identificar os serviços afetados e confirmar se algo foi comprometido, é necessário complementar as informações fornecidas pelo *Snort* com fontes adicionais para obter mais contexto e melhorar a compreensão da situação.

O projeto começou com a atualização do *Snort* para a versão mais recente e a atualização dos *scripts* em funcionamento que serviam para uma correlação básica, enriquecimento de eventos e

notificação de utilizadores sobre os alertas gerados. O plano original era continuar a melhorar a capacidade de correlação desses *scripts* com a integração de fontes de informação adicionais. Após a atualização do *Snort* e dos *scripts*, foram identificados vários problemas com esta solução, problemas que, após avaliação, levaram a alterações no projeto.

Essas alterações lidaram à investigação de plataformas que poderiam ajudar a atingir os objetivos do projeto. Foram encontradas e avaliadas soluções de *Security Information and Event Management* (SIEM) com base em critérios que têm em conta o enquandramento na instituição. Os fatores decisivos para CIÊNCIAS ULisboa foram: funcionalidades, flexibilidade de implementação, custo e modelo de licenciamento.

Das plataformas avaliadas, como *Splunk Enterprise Security*, *QRadar*, *SumoLogic*, *InsightIDR*, *Security Onion* e *Wazuh*. O *Wazuh* foi selecionado, configurado e utilizado para analisar e normalizar dados. Através da centralização de eventos de segurança de rede e sistemas numa única plataforma para visualização, enriquecimento e correlação, é facilitado o processo de monitorização e resposta de segurança. O Wazuh foi também configurado para atuar como uma solução *Extended Detection and Response* (XDR) que deteta e age sobre ameaças em tempo real com o uso de agentes instalados nos sistemas integrados.

O processo de configuração do *Wazuh* começou com a personalização do ficheiro de configuração do servidor *Wazuh* para ativar ou desativar funcionalidades específicas com base nos requisitos operacionais. Seguiu-se a geração de uma chave de acesso para autenticar os agentes e a configuração da taxa máxima de eventos enviados por segundo pelos agentes para otimizar o fluxo de dados baseado no nível de atividade de cada máquina.

Para a integração de fontes de informação, como *logs*, foram utilizados *decoders* que decompõem a informação recolhida através de expressões regulares bem definidas. Os dados decompostos relevantes dos *logs* são atribuídos aos campos apropriados no *Wazuh*. A normalização dos campos é crucial para permitir uma correlação e análise eficazes. Após a configuração de várias fontes de informação, incluindo o *Snort*, foram identificados os alertas mais pertinentes e aqueles que poderiam ser silenciados para reduzir a fadiga dos alertas.

Para o enriquecimento de eventos foi utilizado o módulo *Wazuh Integrator* que permitiu utilizar *scripts*. Este método adotado proporcionou maior flexibilidade no tratamento de alertas.

Ao analisar o tráfego gerado, várias observações foram feitas com base nos IPs de origem e destino. Em particular, foi observado que um número significativo de alertas teve origem em IPs internos, especificamente entre o *reverse proxy* e os servidores internos, pois o IP do *reverse proxy* substitui o IP original nos pedidos efetuados.

A solução implementada passou por utilizar os *scripts* de enriquecimento para consultar outro sistema de monitorização existente. A partir dele, o domínio atacado e o IP de origem real do atacante podem ser recuperados ao fornecer os IPs e portos utilizados na comunicação alertada. Isso permite substituir os dados iniciais do alerta, criando eventos com informações mais precisas.

Para CIÊNCIAS ULisboa, caso sejam detectados ataques pelos IDSs ou outros eventos, o bloqueio destes ao nível do sistema individual ainda permitiria que fossem desencadeados eventos na

camada de rede. Uma solução ideal seria enviar os IPs para o firewall do perímetro, cortando completamente a comunicação, e foi exatamente essa a solução avaliada. Para determinar a reputação de um IP, foi utilizada a plataforma *AbuseIPDB*.

A implementação do projeto revelou comunicações anormais entre sistemas internos que desviavam das boas práticas e que deveriam ser desativadas. Além disso, facilitou a descoberta de um serviço público não utilizado que era alvo de ataque e, que em resultado, foi desativado.

Em suma, o *Snort* foi configurado com sucesso e continua a servir como o IDS primário para CIÊNCIAS ULisboa, melhorando significativamente a segurança da rede. Aproveitando o enriquecimento adicional implementado, o *Snort* permitiu a CIÊNCIAS ULisboa detectar ameaças e identificar comunicações irregulares na rede, que poderiam ter sido ignoradas, expondo serviços a potenciais explorações.

O *Wazuh* foi implementado e integrado com sucesso na infraestrutura existente. Embora os resultados imediatos possam não ser totalmente tangíveis, a cibersegurança é um esforço contínuo, e os benefícios mais significativos provavelmente surgirão a longo prazo com novas melhorias e otimizações desenvolvidas ao longo do tempo através da monitorização e análise contínua dos dados obtidos.

A maior visibilidade proporcionada pelo projeto também permitiu a CIÊNCIAS ULisboa cumprir os requisitos das suas políticas de segurança interna, estabelecendo-o efetivamente como a plataforma de segurança da instituição para a deteção de ameaças cibernéticas.

Palavras-chave: Segurança da informação, Sistema de deteção de intrusões, *SIEM*, *Logs*, *Wazuh*

Abstract

A large number of events are produced and recorded each second in any reasonably sized Information Technology (IT) infrastructure. These events are potentially interesting for security, detailing things such as the domains visited by a certain Internet Protocol (IP) address and the payloads that may have been injected in the existing fields. However, having to inspect entries in log files to identify suspicious activity manually is not a viable solution. Instead, tools like Intrusion Detection Systems (IDS) are used to automate the analysis and signal potential malicious activity.

One of the most popular open-source IDSs is Snort. Snort alerts only provide IP addresses, ports, and some contextual information. To identify targeted services and confirm if anything was affected, one should complement the information provided by Snort with additional sources to understand what happened. By correlating the information received from the alert and other sources such as the logged information of running services, it is possible to determine what was affected and its cause, or even determine the legitimacy of attacks and stop them before they do any harm.

In this project, multiple Security Information and Event Management (SIEM) solutions were evaluated. One was selected, configured, and used to analyze and normalize data to allow for the correlation of multiple events to verify a threat's legitimacy and centralize security data from both the network and systems into one platform for easy enrichment and visualization of data. The chosen solution, Wazuh, also served as an extended detection and response (XDR) solution to detect and act on threats in real-time with the use of agents installed on endpoints.

Keywords: Information Security, Intrusion Detection System, SIEM, Logs, Wazuh

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Organizations and institutions provide many services to the public through their own customized platforms. Cyber-criminals, people who use technology to commit malicious activities on digital systems or networks, take advantage of this opportunity as each platform has its own security issues that can be exploited. However, most of the time, these exploitation attempts go unnoticed by organizations. Why?

The usage of these platforms presents various challenges for organizations, diverting resources away from cybersecurity management. Challenges include efficiently managing these platforms to avoid operational issues like downtime and continuously developing new technologies or services to stay ahead.

Organizations often store events generated by their services for regulatory compliance and debugging purposes. These events contain information such as timestamps to identify when something happened, if performed operations were successful or unsuccessful, issues that may have occurred, and sometimes even a possible cause for a particular issue. Dedicated tools to protect organizations from cyber threats such as firewalls, intrusion detection/prevention systems, and endpoint protection also generate events.

Traditionally, these tools generate events by analyzing data in their own context. To be specific, these tools don't work together to identify threats and generated events may even contain different information depending on the tool used. Combining the information that all tools provide or just centralizing the incoming information is already beneficial as trying to determine whether there is a real threat amid different tools becomes difficult, especially when the information is spread out across multiple platforms, as you have to dedicate time to manually analyze and understand if the threat is real or the alerts generated from multiple tools are related to the same threat.

Another challenge is knowing how to determine when an event generated from a non-security-oriented tool, is indicative of something malicious happening. Most of the time, these events are not relevant. Furthermore, it is a huge challenge to manage them as they are constantly generated by the IT infrastructure and are very diversified. Trying to differentiate between normal and abnormal events is a difficult endeavor as there are different types of devices and countless applications that generate distinct events that may even be unique to some users and their privileged accounts.

To understand this, event correlation is used. It connects the dots between events and lets

individuals get the full scope of a threat, leading to the reduction of false positives and time spent on them. This allows the information security team to handle incidents more rapidly instead of trying to find the connections between events themselves leading to loss of time in situations where time is of the essence. This is why organizations have been increasingly recognizing the important role of Security Information and Event Management (SIEM) tools in fortifying their cybersecurity capabilities. Now, CIÊNCIAS ULisboa, the higher education institution where this project is developed, is also embracing this trend, actively participating in the integration and implementation of SIEM solutions to improve the security of its environment and assets.

1.1 Motivation

The evolution of threats has become worrisome in the past few years. Simple malware that annoyed users, like adware, has evolved into highly sophisticated programs that are capable of crippling entire systems. As such, cybersecurity has become the utmost priority in today's digital landscape, rapidly evolving into a multi-billion-dollar industry.[1]

In response to this escalating threat landscape, the cybersecurity industry has developed advanced tools and technologies to improve its defenses to detect and deal with intrusions swiftly. Security Information and Event Management (SIEM) tools are an essential part of the cybersecurity defense arsenal since they work as solutions that can analyze and respond to security incidents in real-time as the absence of robust cybersecurity measures exposes organizations to significant risks, ranging from financial losses and reputational damage to legal liabilities and regulatory penalties.

Thus, integrating sophisticated cybersecurity tools like a SIEM in CIÊNCIAS ULisboa is essential to bolster its security posture and serve as a foundation for future improvements as attacks on the education sector have been steadily increasing over time.[2] With more than 5600 students and 700 staff members, over 400 access points and thousands of connected devices, all of which could be potentially exploited, CIÊNCIAS ULisboa is a prime target for malicious actors.

1.2 Objectives

This project encompasses several key objectives. By integrating a SIEM into its existing infrastructure, CIÊNCIAS ULisboa aims to enhance its cybersecurity posture and establish a more robust defense against threats. Furthermore, this integration provides a foundation for continuous improvement.

Event correlation and enrichment play a central role in this endeavor, as it allows CIÊNCIAS ULisboa to gain a comprehensive understanding of threats by connecting the dots between various security incidents across its network and systems.

[1] https://www.fortunebusinessinsights.com/industry-reports/cyber-security-market-101165

[2] https://www.microsoft.com/en-us/security/blog/2024/10/10/cyber-signals-issue-8-education-under-siege-how-cybercriminals-target-our-schools/

Additionally, configuring proper endpoint monitoring is essential to detect potential breaches as network security tools and log files may not always indicate potential attacks. By implementing robust endpoint security monitoring measures, it is possible to safeguard individual devices.

In summary, the main objectives of this project are to:

- Implement a SIEM tool to meet the specific needs of CIÊNCIAS ULisboa;
- Centralize security alerts in a single platform to facilitate correlation;
- Enrich existing event data with additional context;
- Configure endpoint monitoring;
- Correlate security events from the network and systems to understand the full scope of an attack;
- Enhance threat detection and response capabilities;
- Notify users about alerts generated from their own devices and encourage self-resolution.

1.3 Methodology

Throughout the project, ideas exchanged played a crucial role in shaping the evolving objectives and methodologies. As challenges arose and insights were gained, the project objectives were changed. The project started by upgrading Snort and enhancing existing alert information enrichment scripts as it would facilitate the inclusion of other information sources with additional context with the help of correlation. This initial solution had issues, that will be detailed in the chapters ahead, which led to the implementation of a SIEM tool.

Implementing and configuring the chosen SIEM solution involved a process of fine-tuning settings, integrating data sources, and customizing rules to monitor and protect the most targeted or vulnerable assets effectively. The development of event enrichment and custom blocklist generation scripts further improved the capabilities of the SIEM, enabling more accurate threat detection. As data accumulated throughout the project lifecycle, an analysis was performed to understand and measure the effectiveness of the implemented solutions.

1.4 Structure of the document

This document is structured into eight chapters, each dedicated to a specific aspect of the project. The introduction sets the primary motivations behind the project, its necessity, utility, and goals. It provides a comprehensive understanding of the project's importance and scope.

Chapter two serves as a foundation, it will explain concepts that are important to the project's understanding. It delves into various aspects of cyberattacks, their methodologies, and countermeasures, while also elucidating the functioning of key protective tools and technologies.

The third chapter transitions to the organizational context, exposing existing mechanisms, technologies, and operations, and highlighting some of the cybersecurity measures adopted.

In the fourth chapter, the focus shifts toward the initial plan that included the configuration of Snort, the revamp of existing scripts , and how some of the challenges faced prompted significant modifications to the project.

Chapter five constitutes an in-depth exploration of multiple SIEM solutions, providing a comparative analysis of their strengths, weaknesses, and suitability for CIÊNCIAS ULisboa' needs and objectives.

Chapter six delves into the details of project implementation, encompassing requirements gathering, architectural design, configuration processes, encountered obstacles, and corresponding solutions to said obstacles.

The implemented solution's evaluation takes place in chapter seven, where the effectiveness of the deployed SIEM solution is assessed.

Finally, chapter eight serves as a summary, highlighting the key findings, outcomes, and lessons learned throughout the project. Additionally, it explores potential future research and development to further augment and refine the implemented solution.

Chapter 2

State-of-the-art

This chapter aims to provide an overview of the state-of-the-art Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) systems. It covers key concepts such as current SIEM/XDR technology, along with necessary background information on cybersecurity attacks, methodologies, vulnerabilities, and Intrusion Detection Systems (IDS) to understand commonly faced challenges.

It also explores some tools and information sources used to feed these systems, detailing their nature, formats, and the types of information they provide. Overall, the chapter motivates the need to have a centralized platform for the observability of security events to allow for the detection of attacks, anomalies, and so forth, including allowing for the existence of correlation between the different information sources.

2.1 Cyber Attacks

Cyber attacks are malicious and deliberate attempts carried out by individuals or groups aiming to compromise the confidentiality, integrity, and availability of information systems, networks, or devices of individuals or organizations by stealing data, causing damage, or seeking some benefit from disrupting the victim's network [14, 28].

Attacks occur frequently due to their low cost, the abundance of freely available learning resources, and easy-to-use automated tools that can launch sophisticated payloads without needing someone to be vastly knowledgeable. Figure 2.1 schematically depicts the perceived evolution of the sophistication of attacks with time.

These attackers vary from inexperienced "script kiddies" to state-sponsored entities, considered Advanced Persistent Threats (APT), employing advanced Tactics, Techniques, and Procedures (TTPs) to infiltrate and achieve their objectives against targets. Despite the varying skill levels of these attackers, the potential rewards of successful attacks are often significant, driving their continual prevalence.

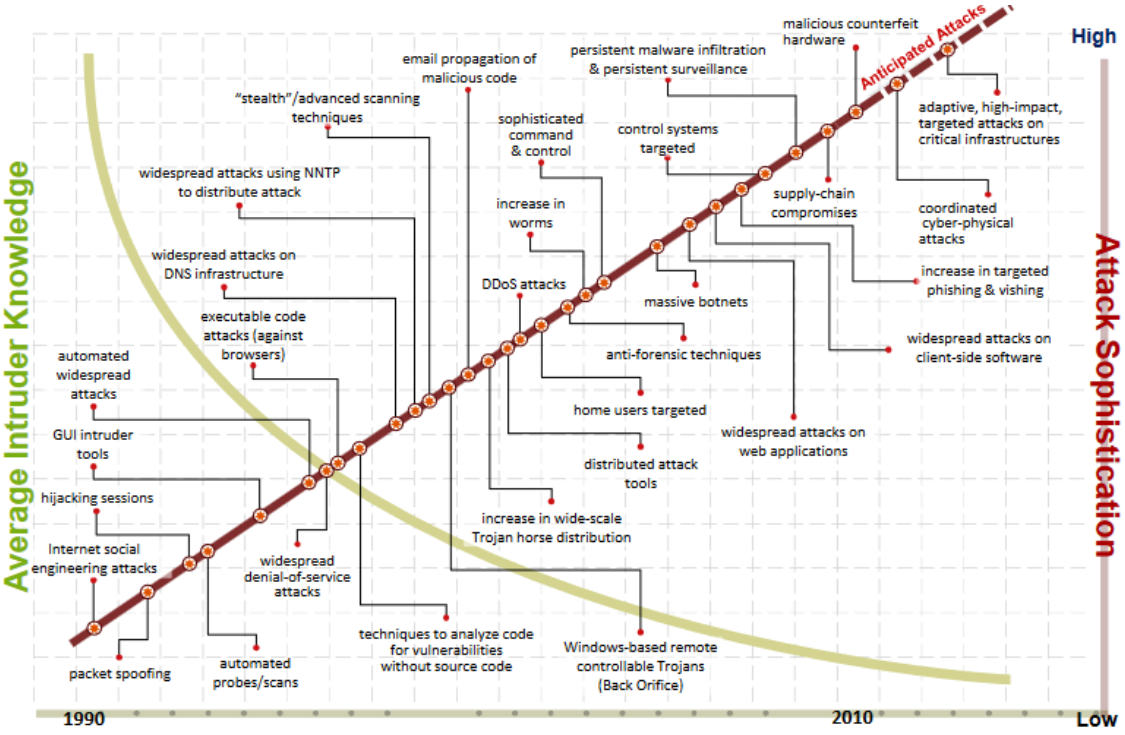

Figure 2.1: Average intruder knowledge and attack sophistication as a function of time. [13]

2.1.1 Vulnerabilities

In every successful attack, a vulnerability must first be present and exploited to gain access to a network or system. Vulnerabilities are weaknesses or flaws in a system, network, or application that attackers can exploit to gain unauthorized access, disrupt services, or cause other forms of damage [25]. These vulnerabilities can manifest in several ways and different components of the IT infrastructure such as software, hardware, and even humans. They can originate from bugs, misconfigurations, lack of proper security controls, design flaws, insider threats, or lack of security awareness training.

For many types of vulnerabilities, there exists a way to exploit them. To give a few examples of generic methods that can be used against common vulnerabilities:

Phishing This is a method that targets human vulnerabilities by using fake information to deceive individuals into revealing sensitive information.

Malware Some malicious software can exploit software vulnerabilities to damage or gain unauthorized access to systems. Types include viruses, worms, trojans, ransomware, etc.

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) These attacks exploit vulnerabilities in networks or systems by overloading them with traffic, making them unavailable to legitimate users.

SQL Injection This is an attack that targets web application vulnerabilities by inserting malicious

SQL queries into input fields to manipulate the backend database into altering, deleting, or retrieving data.

Brute Force Attacks This type of malicious activity systematically tries a large number of possible usernames/passwords to gain access to a system. It targets vulnerabilities in authentication mechanisms, such as weak or guessable passwords, or a lack of controls such as limiting successive authentication attempts.

2.1.2 Advanced Persistent Threats

An Advanced Persistent Threat (APT) is a sophisticated and continuous cyberattack where an attacker infiltrates a network and remains undetected for an extended time, potentially months or even years. Their primary goal is to steal sensitive data or disrupt critical infrastructures through a carefully planned and executed operation tailored to breach a targeted organization's defenses and evade existing security measures [28].

Key characteristics of APTs include:

Customization and Sophistication APTs require a higher level of customization and sophistication compared to traditional cyberattacks. They are meticulously crafted to exploit specific vulnerabilities within the target organization.

Well-Funded, Experienced Adversaries The attackers behind APTs are typically well-funded and highly skilled, often state-sponsored.

High-level Targets These adversaries often target high-value organizations, such as government agencies, financial institutions, multinational corporations, and critical infrastructures like power plants.

Extensive Research and Planning Before launching an APT, attackers spend a considerable amount of time and resources researching their target. This includes identifying and understanding the organization's vulnerabilities and security measures to ensure a successful infiltration and prolonged undetected presence.

Compared to APTs, traditional cyber attacks are almost the opposite. They are not specifically targeted but instead opportunistic, directed at any available target, whether it's an individual or a large corporation. These attacks are typically not carried out by well-funded groups but rather by individuals or small groups as the tools used are often publicly available or are recognized as Malware-as-a-Service (MaaS).

The primary objectives of these attackers are usually financial gain or skill development. When they succeed in breaching a system, their actions tend to be "loud" and conspicuous. For instance, they might deploy ransomware on a single machine, which quickly alerts the security teams. This is in contrast to APTs, where attackers carefully navigate the internal network, compromise multiple systems, and coordinate the deployment of their malware payload simultaneously across all endpoints to avoid detection and maximize impact.

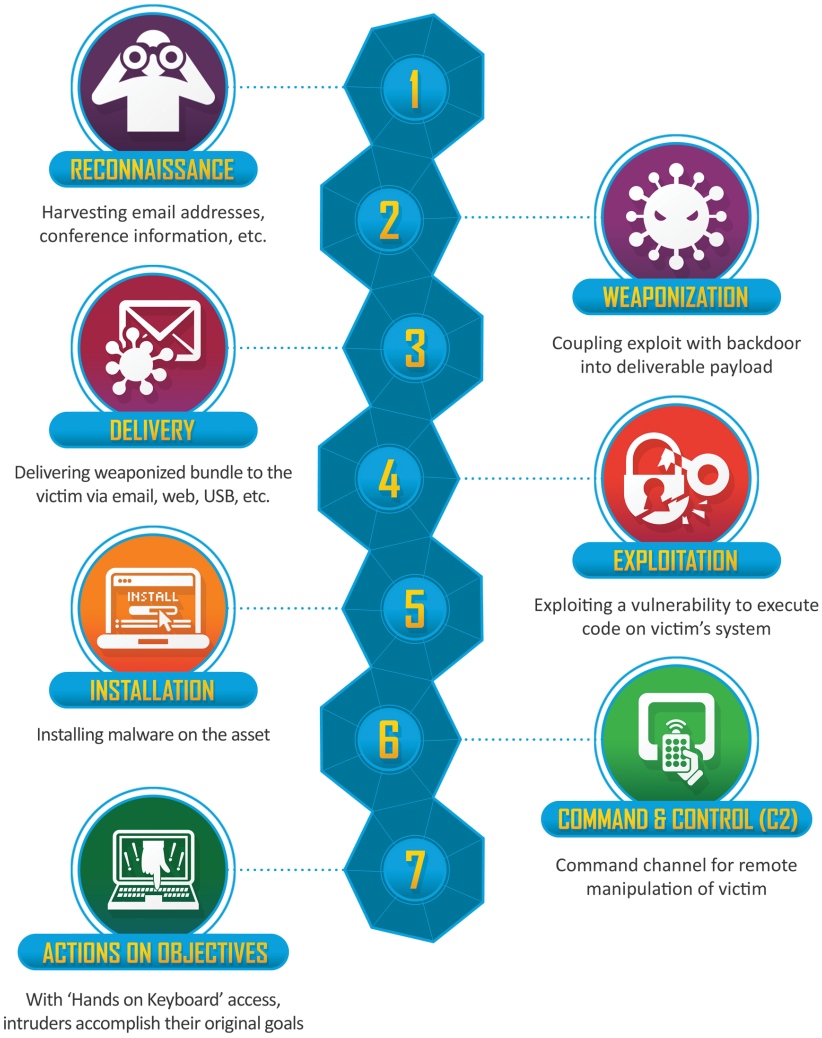

Figure 2.2: Cyber Kill Chain. [21]

2.1.3 Attack Methodology

Traditional cyber attacks and APTs use the structured stages of a cyber attack methodology depicted in Fig. 2.2. The difference between them rests in the sophistication, persistence, and objectives of the attack rather than the stages themselves.

Reconnaissance Involves thoroughly gathering detailed information about the target for a long time by looking for vulnerabilities, using open-source intelligence (OSINT), social engineering, and active probing with minimal detection.

Weaponization Focuses on developing custom payloads tailored to the specific target, employing malware, zero-day exploits, and tailored phishing campaigns.

Delivery Aims to carefully and covertly deliver the payload to the target through spear phishing (targeted phishing), water-holing attacks (compromising a website used by a target), USB flash drives, social media, etc.

Exploitation Consists of exploiting the target's vulnerabilities while avoiding detection. The trigger can be time, human interaction, or other factors.

Installation Involves installing malware that provides long-term access while evading detection, often through backdoors, rootkits, or living-off-the-land techniques (using legitimate tools and processes).

Command and Control (C2) Seeks to open and maintain covert communication channels, using encrypted communications to avoid detection from security tools.

Action on Objectives These are carried out to achieve long-term goals such as intelligence gathering, sabotage, data theft, long-term espionage, and manipulation of critical data or systems.

2.2 Detection and Prevention

The standard approach to detect malicious activity is for security teams to look for Indicators of Compromise (IoC) or Indicators of Attack (IoA). Although interconnected, IoCs and IoAs have distinct uses.

IoCs are forensic data collected after an attack has occurred. They provide insights into how an attack happened. IoCs can help end an ongoing incident and, by understanding the specific vulnerabilities exploited in past attacks, organizations can better safeguard themselves against similar future attacks.

On the other hand, IoAs enable security teams to respond to threats in real-time, before they can compromise their target. IoAs focus on identifying malicious behaviors and tactics as they occur, allowing for a proactive defensive response [2]. Together, IoCs and IoAs form a comprehensive approach to threat detection and prevention.

Effective detection and prevention methods are important in protecting against various cyber threats, particularly APTs. Since they are more advanced than traditional attacks, fortifying defenses against them also implies resilience against traditional ones. As such, some strategies for the detection and prevention of APT campaigns include:

Threat Intelligence Integrate threat intelligence feeds to gain insights into evolving APT Tactics, Techniques, and Procedures (TTPs).

Endpoint Protection Deploy Host-based Intrusion Detection System (HIDS) with features like behavioral analysis, file integrity monitoring, and periodic scans for deployed obscured malware.

Network Protection Segment the network into distinct zones with restricted access controls to hinder the lateral movement of APTs. Implement Network-based Intrusion Detection Systems (NIDS) to monitor malicious traffic.

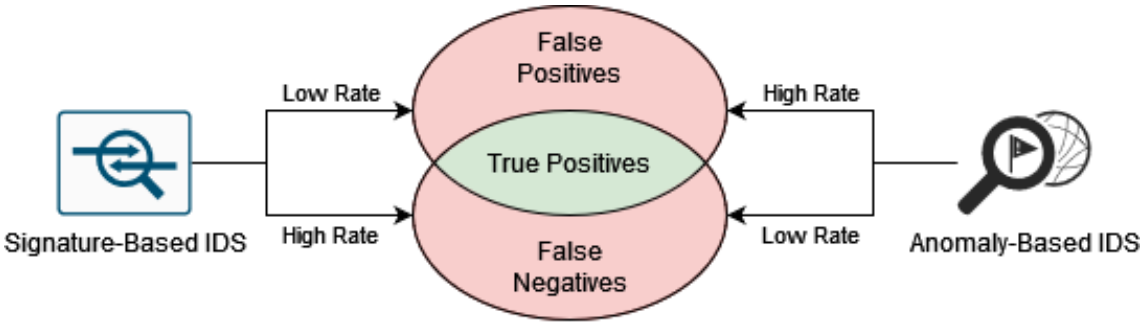

Figure 2.3: Signature vs Anomaly Based IDS. Based on [34]

Role-Based Access Control Implement strict access controls and enforce the least privilege principle to limit unauthorized access to critical systems and data based on roles given to accounts.

2.2.1 Intrusion Detection/Prevention Systems

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) have the objective of observing network or system data and respectively alerting or blocking malicious activity that may occur in a system or network [38]. They are necessary for organizations as they form a critical line of defense against threats that might bypass the firewall. While firewalls are designed to minimize the attack surface by controlling incoming and outgoing network traffic, IDS and IPS provide additional layers of security by detecting and, in the case of IPS, blocking attacks that target any remaining open surface. As depicted in Fig. 2.3, IDS and IPS operate using two primary methods: signature-based detection (or knowledge-based) and anomaly-based (or behavior-based) detection.

Signature-Based Detection This method relies on a database of known malicious patterns or signatures. The system compares network traffic and system activities against these signatures or patterns to identify potential threats. When a match is found, the IDS triggers an alert to be analyzed by the security team while an IPS can automatically block the malicious activity. This method is effective against known threats. However, it fails to detect new or evolving threats leading to a large number of false negatives (undetected real intrusions).

Using existing data can help reduce the number of false positives in signature-based IDS/IPS. However, as technology evolves, new legitimate patterns or signatures may still trigger reactions from these systems. Therefore, it is critical to keep IDS/IPS up-to-date to ensure the capability of detecting new attacks while also minimizing false positives (falsely detected intrusions) [38].

Anomaly-Based Detection This method involves monitoring normal network behavior and flagging deviations from this baseline as potential threats. The system uses Machine Learning (ML) and statistical analysis to distinguish between regular activities (noise) and suspicious activities. This approach is useful for identifying novel or sophisticated attacks that do not

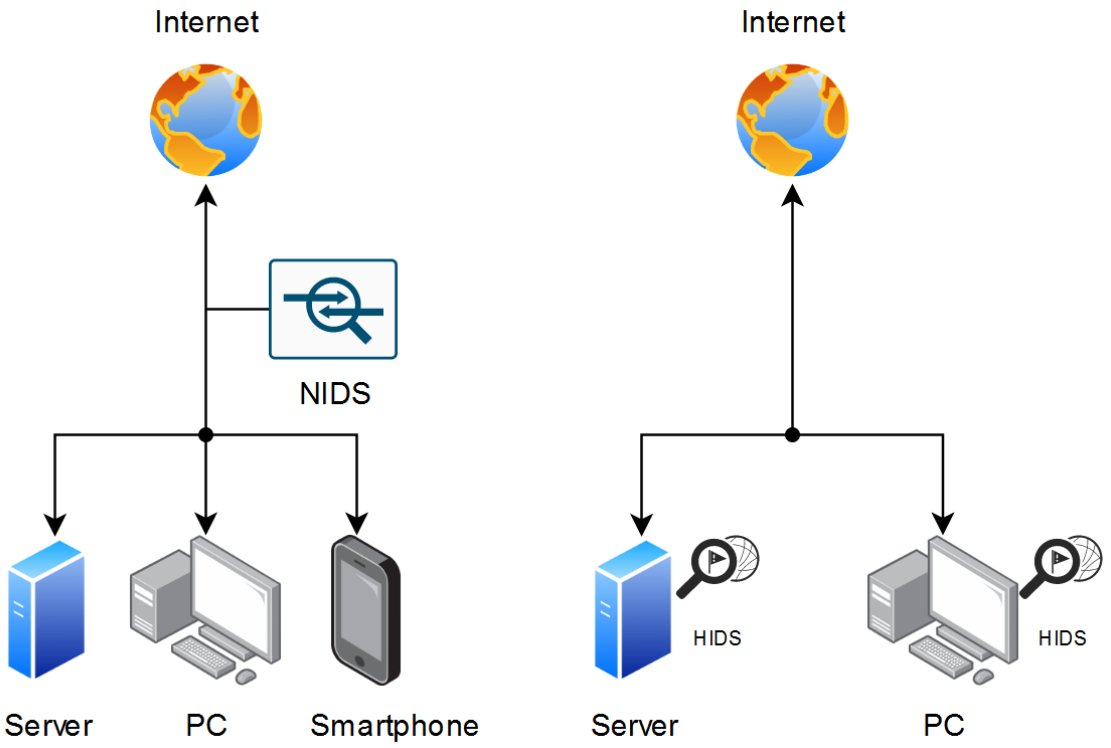

Figure 2.4: NIDS vs HIDS.

match any known signatures. As anything that deviates from normal behavior triggers a response, it is prone to a high number of false positives [38].

IDS/IPS can be deployed to monitor and detect malicious activity at the network or host levels:

Network-based Intrusion Detection System (NIDS) These systems operate on the network and analyze traffic for signs of malicious activity.

Host-based Intrusion Detection System (HIDS) These systems are installed on individual hosts. They monitor and analyze the internal behavior of a single host, such as system logs, file integrity, and process activity to identify potential threats.

Figure 2.4 illustrates the placement of the IDS/IPS depending on their operation method. Both can be used in conjunction to ensure full coverage of the infrastructure. By monitoring all hosts individually and the network as a whole, organizations can achieve a more thorough detection capability. To facilitate better observability and analysis, it is recommended to feed the incoming data from these systems into a centralized monitoring tool like a SIEM to ease the process of correlating events, identifying patterns, and responding to potential threats across the entire infrastructure.

2.2.2 Logs

Logs are records that contain event data from a variety of devices and applications. These can be used for a multitude of purposes such as debugging, monitoring, auditing, and security analysis [17].

Many different types of information can be obtained from the events stored in logs. Typical fields include:

Timestamp The date and time when the event occurred.

Log Level The severity or importance of the event (e.g., INFO, WARNING, ERROR).

Source : The origin of the log entry, such as the name of the specific application.

Message : A short description of the event.

Context/Metadata : Additional information that provides more context about the event, such as IDs and IP addresses.

Because of the volume and variety of generated logs from an IT infrastructure, managing them is very complex as not all of them prove to be useful, especially for security monitoring. For security, the most valuable data to have are events that might indicate potential attacks or allow security teams to identify who caused the event to be generated. Events that indicate problems with the application or requests being made also have their uses to diagnose issues. Key fields that are particularly useful for security teams include IP addresses, ports, URLs, and domain names.

2.2.3 Open-source intelligence

Open-source intelligence (OSINT) is the collection of data that can be gathered by using publicly available sources and processing that data to gain a deeper understanding of a particular subject, concept, or individual. By using publicly accessible resources, everyone can freely use this data to their advantage without breaching any copyright or privacy laws while also avoiding expenses.

Attackers use OSINT to gain intelligence about targets, especially individuals who might be employed on those specific organizations to exploit via social engineering, as nowadays, you can gain a lot of information about people with social media platforms.

These targeted individuals can also use OSINT to their advantage to determine what type of information is available about themselves. With this knowledge, they can better protect themselves from certain attacks.

For organizations, OSINT can be used to gain insights into the market and competitors. It can also improve cyber defense capabilities by gaining data about threats, i.e. cyber threat intelligence [33]. This intelligence helps them gain and understand a multitude of information about threats such as:

Threat Actors Information about individuals, groups, or organizations that may pose a threat.

Threat Indicators Data like IP addresses, domain names, and file hashes associated with malicious activity.

Tactics, Techniques, and Procedures Methods used by attackers to infiltrate systems and networks.

Vulnerabilities Weaknesses in systems or software that can be exploited by attackers.

Impact Analysis Understanding the potential consequences of specific threats to the organization.

2.2.4 Security Information Event Management

Security Information Event Management (SIEM) is a centralized platform designed to aggregate and analyze data in real-time to detect threats, as illustrated in Fig. 2.5, and ensure that the systems in use comply with an organization's security policies [36]. SIEM platforms provide a plethora of functionalities such as [15]:

Log Collection and Aggregation The tool gathers logs from various sources such as servers, applications, intrusion detection systems, etc. This centralized log aggregation ensures teams have full visibility across the entire IT infrastructure.

Event Correlation The centralization of information allows the tool to analyze and correlate events from different sources to identify patterns and gain more insight into a threat.

Alerting and Notification SIEM systems generate real-time alerts when potential threats are detected, notifying security personnel through channels such as email or SMS.

Dashboards and Reporting Security teams can create customizable dashboards and detailed reporting features to provide better visibility for certain aspects. The reporting feature of a SIEM system allows for the creation of automatic reports that facilitate audits, compliance documentation, and management reviews.

User and Entity Behavior Analytics Utilization of user and entity behavior to detect anomalies by using ML, statistical analysis, and data modeling techniques like an anomaly-based IDS.

Compliance Management Ensures that an organization's systems are compliant with relevant laws, regulations, industry standards, and internal policies to avoid legal penalties and maintain their reputation.

Log Retention and Archiving Allow for proper data management and safeguard for compliance. It is the process of storing, managing, and preserving log data to ensure that it is available for future reference and analysis.

Threat Intelligence Integration Integration with external threat intelligence allows for enhanced detection capabilities. Having access to real-time data on known threats and contextual information helps to adjust detection rules and response actions effectively.

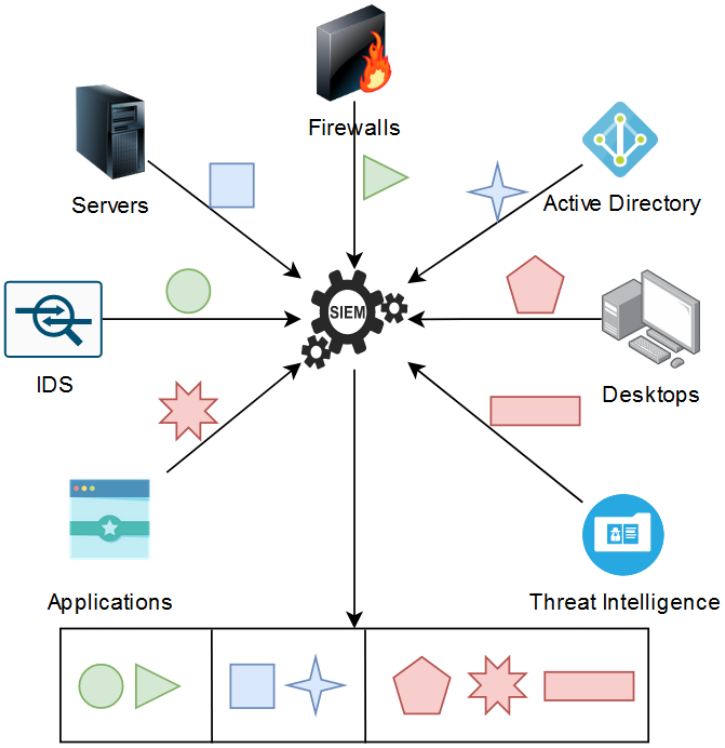

Figure 2.5: SIEM Correlation from multiple data points. [34]

Although these tools are very useful and important for organizations, they do have their limitations. SIEM solutions often come with substantial upfront costs for hardware, software licenses, and ongoing maintenance. The need for skilled personnel to operate and maintain the system also increases the cost. Without proper personnel, in environments with high volumes of security events, the amount of generated alerts overwhelms security teams that don't know what they are looking for. This can result in alert fatigue, where important alerts may be overlooked. Correlation helps with this aspect but the tool's correlation capabilities rely heavily on the quality and normalization of the events ingested.

SIEM tools can be deployed either on-premises or in the cloud. Table 2.1 created from information in [3, 32, 37, 18] show that both options have their advantages and disadvantages in various categories.

2.2.5 Extended Detection and Response

Extended Detection and Response (XDR) is a more novel approach to cybersecurity and an evolution of Endpoint Detection and Response (EDR). It protects endpoints by detecting and reacting to malware in real-time and using ML-driven analytics to automate responses, providing a more proactive measure for threat mitigation. It also extends current SIEM capabilities by utilizing Artificial Intelligence (AI) to perform behavioral analysis to detect anomalies and correlate events across multiple data points [24, 22].

Features	On-premises	Cloud-based
Deployment model	Requires hardware infrastructure, software installation, and ongoing maintenance by the organization's IT team.	No need to acquire hardware and integrate it into the existing infrastructure.
Scalability	Limited by on-premises hardware and infrastructure. Scaling up requires the acquisition of additional hardware and configuration.	Easily scale up or down providing a more flexible and cost-effective approach depending on needs.
Customization	Flexibility to be tailored to the specific requirements and needs of an organization. SIEMs can be "fully" customizable to adhere to custom security policies and configurations.	Less control and customization as most of the setup is managed by the service provider.
Cost structure	Requires upfront expenses for hardware, software licenses, and implementation costs. Ongoing operational expenses include maintenance, upgrades, and staffing.	Typically follows a monthly or annual subscription-based pricing model with the amount of data ingested, or assets monitored taken into consideration. Over time, costs are superior to an on-premises deployment.
Maintenance and updates	Organizations are responsible for maintaining and updating the SIEM software as well as managing their hardware upgrades, patching, and backups.	Maintenance tasks such as software updates, patches, and backups are managed by the cloud service provider.
Data Privacy	Sensitive data never leaves the organization's data centers.	Storing sensitive security data in the cloud can raise concerns.

Table 2.1: On-premises SIEM vs Cloud-based SIEM

2.3 Used tools

Tools with specific functionalities were used to achieve the desired result. This includes tools like:

Python Python is a high-level programming language, particularly useful for fast application development, and scripting. It is also easy to learn due to its syntax which improves code readability and maintainability [26].

Rsyslog Rsyslog is an open-source utility that is used to rapidly process logs from a wide variety of sources. It can be used to enrich incoming data with existing Rsyslog modules and output them to a desired destination. If existing modules prove to be insufficient, logs can be forwarded to other programs for additional processing [27].

Cron Cron is a job scheduler on Unix operating systems that is used to execute commands or scripts periodically. It is used to automate system maintenance or administration making it useful for scheduling repetitive tasks such as executing a script every day at a defined period that compiles yesterday's alerts summary [11].

MySQL MySQL is an open-source relational database management system used to manage and organize large sets of data. It is commonly utilized in applications to store and retrieve data rapidly and efficiently. Due to its popularity, there exist many types of attacks that can exploit its usage, therefore it is very advisable to implement rigorous security measures on the applications that use the database and manage user permissions with care [23].

AbuseIPDB AbuseIPDB is a web-based platform that enables users to contribute to and access a shared database of potentially malicious IP addresses. By leveraging crowd-sourced data, AbuseIPDB provides valuable insights into the reputation and trustworthiness of IP addresses, aiding organizations in their cybersecurity efforts and proving to be a valuable OSINT source [1].

Chapter 3

Host Institution

This chapter provides a comprehensive overview of the current IT infrastructure, network configuration, systems that support operations, and some of the existing security measures at the host institution of this project. By understanding the current infrastructure and security posture, a critical assessment can be made to implement enhancements that protect against present and future cyber threats.

3.1 Background

CIÊNCIAS ULisboa is a a public higher education institution, part of the Universidade de Lisboa, with more than 5600 students and 700 staff members with around 1000 or so students graduating and entering each year.[1] It "has a strong technological component and was one of the first places in Portugal with Internet connectivity. Until 2005, CIÊNCIAS ULisboa was the connection hub of the Universidade de Lisboa to the Internet. Since then, the connection of the Universidade de Lisboa to the FCCN (Fundação para o Cálculo Científico Nacional), the institution that interconnects all the national Higher Education Institutions, was handed over to the Rectorate of Universidade de Lisboa" [7].

This project is implemented in a unique environment where control over the actions that can be performed by a majority of users, which are students, is not attainable. To meticulously monitor and manage over 400 access points, a vast array of over 10,000 devices (many of which are not owned or managed by the CIÊNCIAS ULisboa), numerous web servers, and multiple domain controllers with diverse functions is beyond the faculty's possibilities.

3.2 Infrastructure & Network

The faculty's computer network is organized around Virtual Local Area Networks (VLANs) that divide traffic and devices according to the users and purposes. As illustrated in Fig. 3.1, the list of VLANs include:

Development Servers for software development.

[1] `https://ciencias.ulisboa.pt/pt/estatisticas`

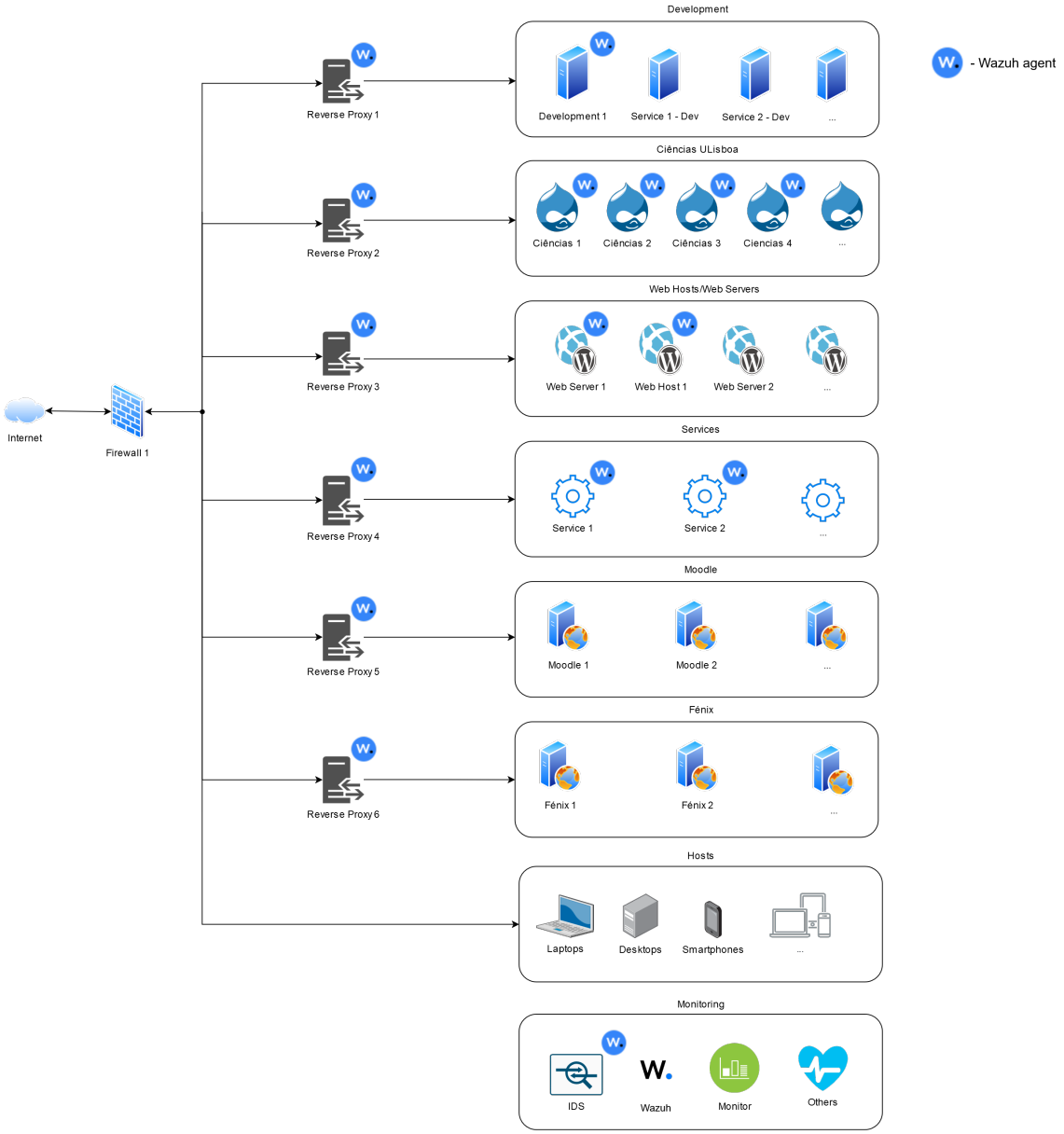

Figure 3.1: CIÊNCIAS ULisboa partial network architecture.

Portal Drupal for CIÊNCIAS ULisboa's main web portal.[2]

Web hosts/Web servers CIÊNCIAS ULisboa's and its members' websites and web applications.

Services CIÊNCIAS ULisboa's main services available to its members and the public.

Moodle Learning management system.

Fénix Academic management system.

Hosts User devices.

Monitoring Internal tools for the observation of systems and networks.

According to [8], the same physical network is used to serve not only traditional data but also various other services such as access control to buildings, surveillance systems, attendance control, and Voice Over IP (VoIP) telephone systems. Buildings on the campus are interconnected by a network based on optical fiber. CIÊNCIAS ULisboa has a backbone with a 1 Gbit Full-Duplex distribution network with the data center backbone using 10Gbit/s links. Overall, CIÊNCIAS ULisboa uses more than 100 switches, connecting a few thousand network sockets. Access to the wired network is controlled, with each socket being individually activated upon request to the IT Department. Computers can be authenticated by MAC address or using the 802.1x protocol. By December 2015, approximately 80% of services provided connectivity using a dual IPv4/IPv6 stack [6].

3.3 Systems

CIÊNCIAS ULisboa has a wide variety of systems, each with distinct hardware, software, and permission configurations tailored to their specific use cases. Virtualization technology is heavily used as it allows for the creation of multiple hosts that can have their individual operating systems and applications on a single physical server, thereby maximizing hardware usage and reducing costs while also facilitating the creation of systems that meet the specific needs of some types of software.

3.3.1 Reverse Proxies

CIÊNCIAS ULisboa's web applications are behind reverse proxy servers that distribute incoming traffic across the backend servers. Reverse proxies act as load balancers, ensuring that no one server is overburdened with requests. There are also mechanisms that cache frequently requested resources to avoid communicating with the backend, effectively lowering response times and using server resources more efficiently. Additionally, management of transport layer security and secure sockets layer (TLS/SSL) is made easier since the reverse proxy servers can deal with the encryption and decryption of data while the internal servers, where web applications are hosted,

[2] `https://ciencias.ulisboa.pt`

only need to provide the resource requested, further improving efficiency. Used software includes Apache[3], Nginx[4], and HAProxy[5].

3.3.2 Web Applications

CIÊNCIAS ULisboa has developed and maintains a variety of web applications to manage different aspects of its infrastructure. These applications handle a variety of functions, from the activation of physical network sockets to user password recovery.[6] These internally developed applications undergo rigorous testing to mitigate security flaws and bugs. However, CIÊNCIAS ULisboa offers web hosting services to its members. In these cases, security and software updates are managed by the respective web application creators which include, students, investigators, and staff. This often results in increased vulnerabilities, as the respective web application creators frequently neglect to update their software or implement additional security measures [5].

3.3.3 Databases

CIÊNCIAS ULisboa employs multiple databases to securely store data that is used for internal tools. These databases are protected by restricting access to only authorized systems, ensuring that only the proper systems with mature code can access them. CIÊNCIAS ULisboa ensures that data storage solutions are optimized for performance, security, and scalability. Each database is maintained, with regular updates and backups to safeguard against data loss and ensure continuous availability. The technologies used include MariaDB, Microsoft SQL, MySQL, and PostgreSQL [10].

3.3.4 Domain Controllers

CIÊNCIAS ULisboa utilizes Microsoft Active Directory (AD) to manage and secure its IT resources. The infrastructure includes multiple domain controllers catering to both student (@alunos.ciencias.ulisboa.pt) and non-student (@ciencias.ulisboa.pt) domains. These domain controllers are responsible for multiple tasks such as:

DNS (Domain Name Service) Provide name resolution services essential for locating network resources and services.

DHCP (Dynamic Host Configuration Protocol) Automate the assignment of IP addresses to devices on the network.

GC (Global Catalog) Facilitate searches across the AD forest by storing a partial, read-only replica of all objects in the directory.

[3] `https://httpd.apache.org/`
[4] `https://nginx.org/en/`
[5] `https://www.haproxy.org/`
[6] `https://passwd.ciencias.ulisboa.pt`

DFS (Distributed File System) Enable access to distributed files across multiple servers, presenting them as a unified file system.

Authentication Verify user and device credentials to ensure secure access to the network.

Access Control Manage permissions and policies to control access to resources, ensuring that only authorized users can access sensitive information.

3.4 Existing Security Measures

Various security measures have been implemented over time to protect CIÊNCIAS ULisboa and ensure compliance with regulations. This section will cover some of the existing measures, providing context about the state of CIÊNCIAS ULisboa's cybersecurity and how this project fits in to address existing limitations.

3.4.1 Firewalls

Two independent firewalls filter traffic. One of these is a Next-Generation Firewall (NGFW) that includes an IDS, which analyzes the traffic entering and exiting CIÊNCIAS ULisboa's network and blocks malicious communication attempts. Having distinct firewalls covering various segments of the network raises the difficulty for attackers to successfully breach CIÊNCIAS ULisboa as payloads have to be specially crafted to avoid getting blocked by each firewall. This also ensures that in the case some segments of the network are compromised, others remain secure. Additionally, a Web Application Firewall (WAF) is implemented to protect backend servers at the application layer.

3.4.2 Intrusion Detection Systems

Snort is CIÊNCIAS ULisboa's main IDS. It has three primary uses: as a packet sniffer, packet logger, and as an alert system. Among other locations, Snort monitors the traffic incoming from the Internet to the reverse proxies and from the reverse proxies to the web servers, firing an alert for potentially malicious traffic, and saving the packet(s) associated with the alert for later analysis. The IDS included with the NGFW also serves as a welcome measure to validate threats identified by Snort while also detecting additional threats that Snort might miss, and vice versa.

3.4.3 Event enrichment and correlation

CIÊNCIAS ULisboa uses scripts that enrich incoming alerts from Snort with additional information by using correlation. This correlation has the goal of aiding in the identification of systems that are being attacked to determine if threats pertaining to different systems are related. However, the integrated information sources are insufficient to obtain useful insights by enriching and correlating data.

3.4.4 Vulnerability Management

To ensure the security and compliance of its critical systems, CIÊNCIAS ULisboa performs vulnerability scans to identify and mitigate potential vulnerabilities. This proactive approach effectively assures that systems are up-to-date with security standards and ensures the integrity and security of software across various distinct systems.

3.4.5 Data Loss Prevention

CIÊNCIAS ULisboa offers a data safeguard service (backups) for the servers managed and/or housed in the data center. Backups are performed on tape using Linear Tape-Open (LTO5) technology. The storage of backup information is distributed between 2 geographically dispersed locations [9].

3.4.6 Monitoring

A monitoring solution observes systems and network devices, providing alerts for critical events such as process crashes, resource exhaustion, and network device malfunctions. This allows for rapid identification and repair of issues to avoid prolonged downtime and ensure optimal performance.

3.4.7 Limitations

Of the currently existing measures, there are various limitations such as the inexistence of a centralized security system to aggregate events from various systems and analyze its data for irregularities or malicious activity. Although endpoints are monitored, the focus hinges more on system-related issues rather than security threats. Additionally, there is a lack of proper active response mechanisms to automatically stop attackers in real time. These are the main limitations that will be addressed in this project.

Chapter 4

Snort

This chapter addresses the aspects of this project that involved a revamp of the existing Snort service at CIÊNCIAS ULisboa. It covers aspects like configuration, architecture, and script rewrite as the service has become outdated and is not providing the best threat detection possible. The limitations found with Snort described in this chapter will provide greater insight as to why the initial plan, which had Snort as its central component, would not be the best for CIÊNCIAS ULisboa's current cybersecurity situation. This analysis led to a better overall solution which will be addressed in the following chapters.

Snort is one of the most popular open-source IPS and it is used in CIÊNCIAS ULisboa. It uses signatures, similar to patterns, to create rules that identify malicious packets and generate alerts [29]. Snort can be deployed in either passive mode acting as an Intrusion Detection System (IDS) or active mode, blocking incoming and outgoing packets and acting instead as an Intrusion Prevention System (IPS). Useful information that can be obtained from alerts include: Timestamp; Signature ID; Message; Classification; Priority Level; Network Interface; Protocol; Source IP; Source Port; Destination IP; Destination Port.

4.1 Configuration

The configuration process started by updating the server's operating system (OS) from an outdated CentOS Linux 6 to Centos Stream 9. Snort was also upgraded from a version 2.x to 3.x and was installed using the first six chapters of a guide.[1] However, it was necessary to modify some steps as the guide is outdated.

Snort configuration was done with the following steps:

1. Preparation (Configure repositories, paths and update OS).
2. Install dependencies with a package manager.
3. Install Snort3 from the git repository following instructions on the web repository and guide.
4. Install Snort3 Extras from the git repository.

[1] `https://snort.org/documents/snort-3-1-0-0-on-centos-stream`

5. Download and install the Snort ruleset from the Snort website.
6. Download and install the IP blocklist from Talos Intelligence.
7. Update Snort configuration to encompass specific needs.
8. Test Snort configuration by verifying if it is producing the expected output.

4.2 Scripts

Scripts developed in house are used to update Snort rules and blocklists. They are also utilized to enrich and correlate Snort alerts as referenced in Sec. 3.4.3. The existing Bash scripts were converted to Python for enhanced readability, exception handling, and, maintainability. Additionally, it provided an opportunity to identify and address any bugs in the original Bash scripts, with the subsequent fixes being implemented in their Python counterparts.

Scripts are arranged in the "Updaters" and "Alerting" groups to highlight their different objectives.

4.2.1 Updaters

These scripts serve to update Snort rules and blocklists. The updater scripts revised were:

ruleUpdater Rule updates allow for the detection of new threats and the updating of old rules as some trigger false positives. Before making any changes, the script creates a backup of the existing rules file. The newest Snort rules are downloaded and the data from these files is concatenated into a single file. This is to ensure that when new files are added by the Snort team, there is no need to manually update the Snort configuration file as it requires the enumeration of the rule files used by the service. In the event of an exception during the update process, these are captured, logged and the rule files are reverted to their original state, if altered, ensuring the integrity of the rule set. Rules are updated every 2 weeks.

blocklistUpdater Snort incorporates a valuable blocklist feature enabling the detection of incoming packets from IPs on the blocklist. Blocklist updates allow security teams to detect if there are ingoing/outgoing traffic from/to potentially malicious sources/destinations that emerge over time. This script retrieves the newest publicly available IP blocklist[2] from Snort and updates the existing blocklist. Blocklists are updated daily as they are constantly updated by their creators.

4.2.2 Alerting

These scripts serve to process Snort alerts and notify users of alerts generated by their devices. The scripts can be divided into 3 stages and their full architecture can be observed in Fig. 4.1:

[2] `https://www.snort.org/downloads/ip-block-list`

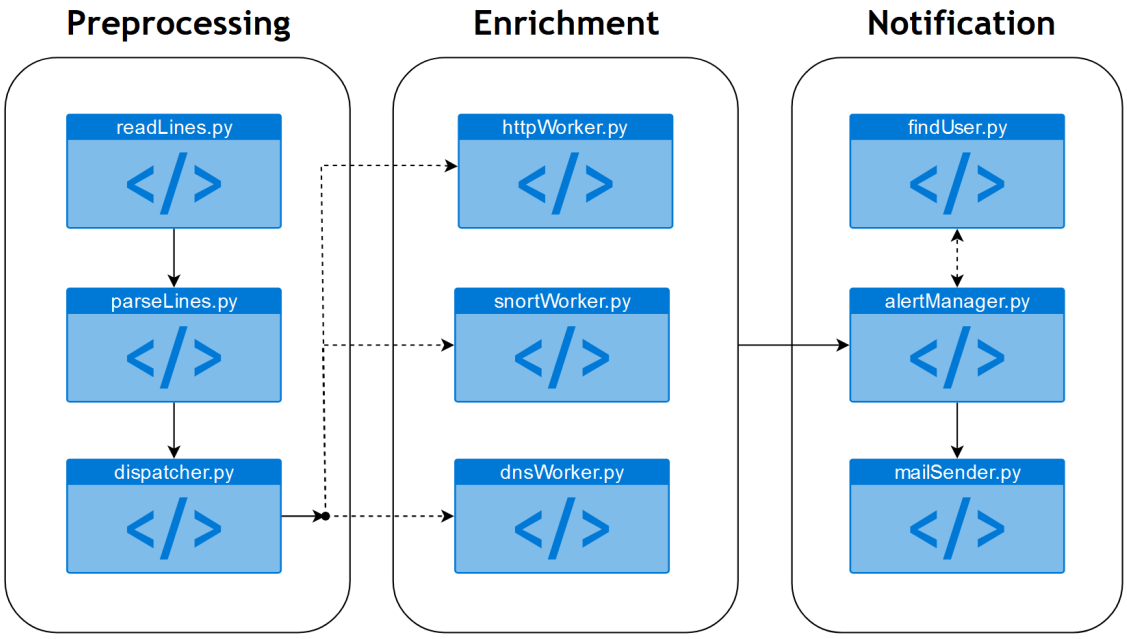

Figure 4.1: Architecture of alerting scripts.

Preprocessing Snort-generated alerts are transmitted to the Rsyslog daemon, as defined in the Snort configuration file. Rsyslog is a tool to process and forward logs as explained in Sec. 2.3. The tool subsequently forwards the received alerts to the "readLines.py" script, as illustrated in Fig. 4.2. Since Rsyslog cannot execute another instance of a script before the previous is terminated, "readLines.py" continuously reads lines of input, and data is sent to the next script for processing, allowing for the processing of multiple alerts almost simultaneously. Even though this approach could be made more efficient, for example, by implementing an asynchronous I/O model to process all incoming alerts in real-time, it would eliminate the need to send data to a separate script for processing and allows multiple alerts to be processed in parallel without waiting for the previous one to complete. The implemented solution proves to be sufficient to deal with the current amount of alerts being triggered by Snort.

The following script uses regular expressions to extract all the information available from the alert such as the "Timestamp", "Signature ID", "Message", "Classification", etc. Data is then sent to "dispatcher.py" which stores the data in a local MySQL database to be used by another service that creates custom blocklists. Depending on the captured port numbers, such as 53 for DNS or 80/443 for HTTP/S, the alert data is then sent to different scripts for enrichment.

Enrichment This stage begins by discarding the alerts whose "Classification" does not match the monitored ones. The remaining are enriched with other information sources. For example, the domain name of the visited website for HTTP/S alerts can be obtained by inspecting the packet that triggered the alert.

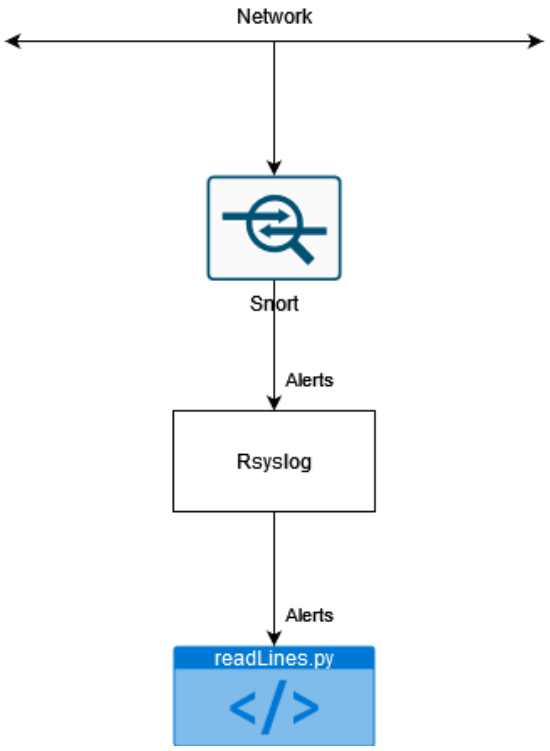

Figure 4.2: Initial flow of Snort alerts.

Notification The stage begins by determining which of the source or destination IP addresses of the alert belongs to CIÊNCIAS ULisboa and its corresponding sub-network. Following that, the IP, "Signature ID" and "Timestamp" are stored in a MySQL database, different from the previous one. The database is then queried to retrieve the total number of entries that share the same IP address, "Signature ID", and occurred within the last minute. If the number of matching entries exceeds a predefined threshold and the sub-networks aligns with those used by faculty members, attempts are made to identify the user whose device has been assigned the troubled IP.

In the case of the Wi-Fi sub-network, the script sends SNMP queries to the Wireless Local-Area Network (WLAN) controller to retrieve the email of the user associated with that IP at that time. If identified as the VPN sub-network, the VPN server is queried to obtain the email that is currently using that IP address. If the sub-network is used for physically connected devices, the script queries a MySQL database using the MAC Address of the device to determine its physical location in CIÊNCIAS ULisboa and its owner. The MAC Address is obtained by inspecting local files containing the Address Resolution Protocol (ARP) table for IPv4 and the Neighbor Discovery Protocol (NDP) table for IPv6 exported from the firewall.

If the IP does not belong to the aforementioned networks, the script employs an exhaustive approach. It systematically uses every query method available, regardless of whether the IP belongs to their respective sub-network. If the user is discovered, its record in the MySQL

database is updated and all the relevant information is sent to the last script in the pipeline.

"MailSender.py" creates an LDAP query and sends it to the correct domain controllers (depending on the domain extracted from the username), to retrieve the user's real name. Subsequently, a message is constructed containing the details of the alert and communicates this information via email to the user, administrators, and support team.

4.3 Benefits & Issues

The move from Snort version 2 to version 3 delivers more robust and advanced features.[3] Many improvements are gained with this upgrade such as, improved logging and packet analysis capabilities. The new version of Snort immediately started to produce a large amount of alerts. As a result, email notifications were disabled to prevent spam and alert fatigue.

To investigate if Snort was operating correctly, a manual analysis was performed, utilizing key parameters such as the "Signature ID", IP addresses, and alert frequency. Results led to adjustments in the configuration of Snort with the objective of lowering the quantity of less relevant alerts being produced by either suppression or rate-limiting mechanisms. This would be beneficial in reducing the "noise" and processing costs of duplicate or irrelevant data.

The conclusion of the Snort configuration was shown to be unsatisfactory concerning the objectives of the project. The more worrisome limitations found were:

Alert fatigue The overwhelming number of alerts being triggered leads to alert fatigue.

Verification mechanisms With only Snort data available, there is no reliable method to differentiate between true and false positives.

Lack of observability The data is only viewable as text offering no comprehensive visualization.

No filtering mechanisms No options to filter data by user-defined criteria.

Correlation difficulties If the main focus is correlating events with Snort, it is challenging to identify which events from other sources are relevant. For example, the existence of a status code 200 in a weblog is normal as it indicates a successful request and is no cause for an issue. However, if the URL utilized had an SQL injection, this would pose a risk. To detect these types of anomalies in distinct events, a new program would have to be developed that efficiently detected and extracted relevant data for correlation purposes.

The lack of observability and filtering mechanisms is largely due to the absence of a dedicated platform, like a Security Information and Event Management (SIEM) tool, to serve as an interface for Snort alerts. This is highlighted in a previous dissertation by João Calado, a former student who worked on integrating Snort into CIÊNCIAS ULisboa, "(...) we believe that this tool or a similar one is definitely a must, and suits perfectly, to perform as an interface to IDSs alerts" [4].

[3] `https://www.snort.org/snort3`

To address these issues, research was conducted on potential solutions, leading to the consideration of SIEM tools.

Chapter 5

SIEM Solutions Comparison

This chapter will explore existing SIEM solutions and compare them to determine which one best addresses the cybersecurity monitoring limitations identified in CIÊNCIAS ULisboa in the previous chapter.

Utilizing existing and well-documented tools instead of developing one in-house is a better choice. This approach offers numerous benefits, including access to additional functionalities developed over time and community support. Often, encountered issues have already been faced and resolved by others, providing solutions or workarounds.

These tools are designed to be scalable to handle enterprise data volume, have an extensive feature set, ensure interoperability within an existing security ecosystem, and are regularly updated to address emerging threats.

5.1 Evaluation Criteria

When evaluating and comparing solutions, it's critical to consider various factors to ensure that the chosen solution aligns with the organization's security and operational needs. The main deciding factors for CIÊNCIAS ULisboa are:

- Functionalities;
- Deployment flexibility;
- Cost and licensing model.

5.1.1 Functionalities

Evaluating the presence of certain functionalities is crucial for making informed decisions about their suitability and effectiveness in meeting an organization's security needs. What is mainly needed by CIÊNCIAS ULisboa are "Event Correlation" and "Alerting and Notification" which most solutions provide. However, with the additional XDR integration, endpoints can be better protected. These endpoints should primarily be the most critical components of CIÊNCIAS ULisboa infrastructure as others, such as desktops in students' laboratories, already have strict access control mechanisms and are not high-value targets.

The common functionalities provided by SIEMs have already been detailed in Ch. 2.2.4. However, any distinct features that are highlighted in each solution will be identified and explained ahead.

5.1.2 Deployment Flexibility

The tool must be integrated into the organization's existing infrastructure without significant disruption while also allowing for future upgrades to be made. An on-premises deployment is the preferred choice, given that it eliminates the need for data to pass through third-party systems and CIÊNCIAS ULisboa's existing infrastructure can provide the needed processing power and storage.

5.1.3 Cost and licensing model

The cost can influence the feasibility and sustainability of adopting a particular tool. Especially in government-funded entities. On-premises solution is a better long-term investment for CIÊNCIAS ULisboa as the cost of acquiring and maintaining the hardware is amortized over time compared to the cost of using external hardware. Having a perpetual license to the software also benefits CIÊNCIAS ULisboa, as implemented technologies into its infrastructure rarely change and stay in production for a large number of years, thus choosing an on-premises deployment with a perpetual license model for the chosen tool is the more appealing option in terms of cost.

5.2 Solutions

Many of the researched tools have the expected core features commonly associated with SIEM solutions. However, a few stood out, garnering attention for their recognition and reputation.

- Splunk Enterprise Security;
- QRadar;
- SumoLogic;
- InsightIDR;
- Wazuh;
- Security Onion.

5.2.1 Splunk Enterprise Security

Splunk Enterprise Security[1] (Splunk ES) is a SIEM built onto Splunk which is a data analytics platform. According to Gartner[2], it is one of the top leaders and visionaries in the SIEM industry as observed in Fig. 5.1. Each quadrant has a different evaluation which is detailed in Fig. 5.2.

[1] https://www.splunk.com/en_us/products/enterprise-security.html
[2] https://www.gartner.com/en

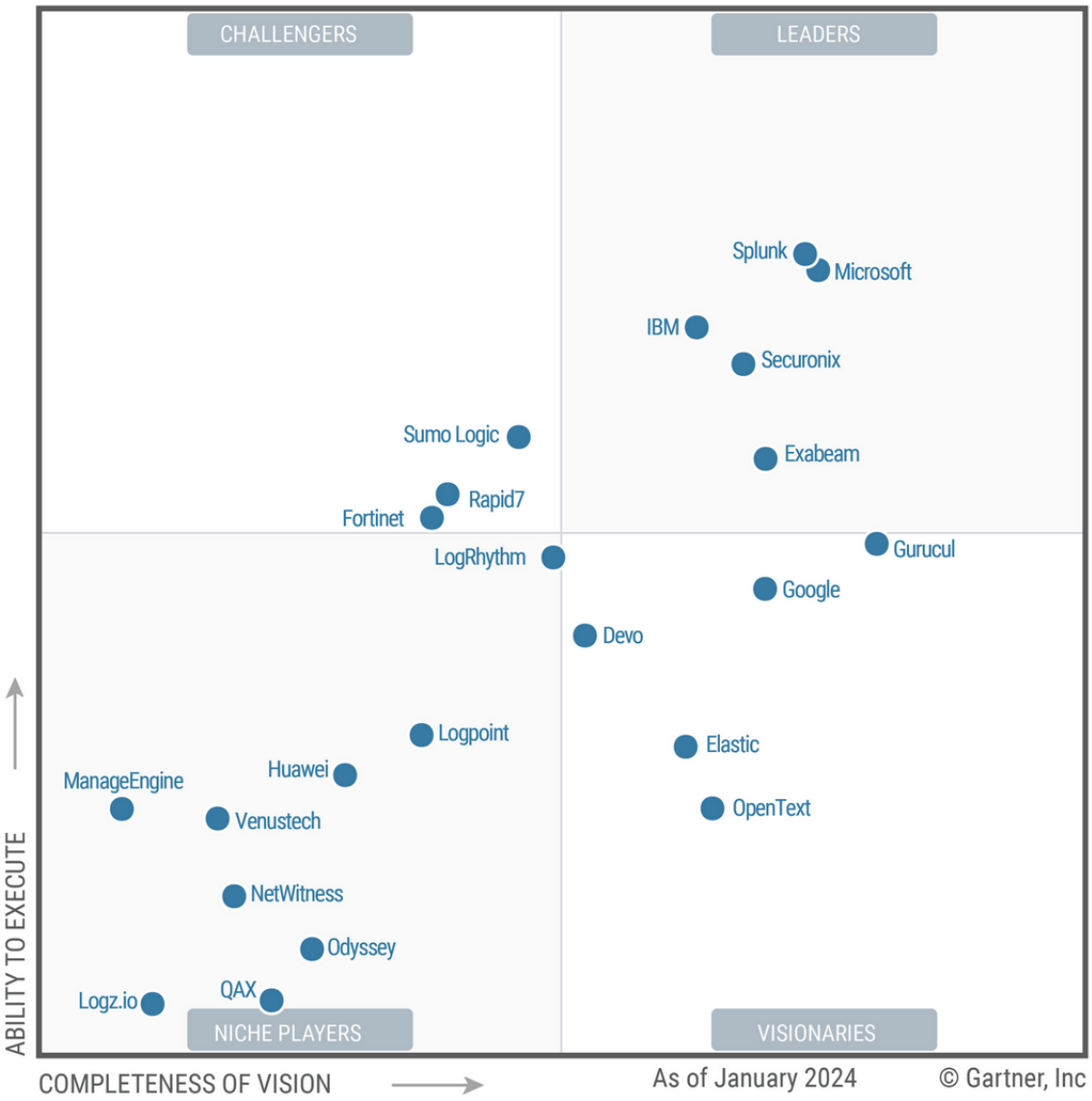

Figure 5.1: Gartner Magic Quadrant for SIEM 2024. [12]

Figure 5.2: Explanation of each quadrant of the Gartner Magic Quadrant.

From its several features, the most prominent ones that are displayed in Splunk's information sources are [31]:

Threat Topology Threat topology allows for the visualization of data in topology graphs, linking threats to risk objects such as hosts or users. For CIÊNCIAS ULisboa it is not a must-have feature. However, in the future, data can be taken and ingested into other platforms to achieve this functionality.

MITRE ATT&CK Framework Matrix Threats are categorized by using the MITRE ATT&CK Framework[3] which would help CIÊNCIAS ULisboa's IT team to determine what a certain threat attempted to accomplish and build situational awareness.

Risk Based Alerting (RBA) Risk-based alerting improves out-of-the-box Splunk detections by attributing risk levels to assets which include users and systems. If the risk level of certain assets exceeds defined thresholds, alerts are generated. This mechanism reduces false positive rates significantly and would be a a helpful feature as CIÊNCIAS ULisboa has a multitude of assets to manage.

Adaptive Response Actions Splunk ES has actions that can be taken against threats either manually or automatically which greatly help in reducing staff work for repetitive tasks.

ES Content Updates and Use Case Library The Splunk Threat Research Team releases pre-packaged security content that can be utilized by organizations to improve detections and

[3] `https://attack.mitre.org/`

responses to the latest threats.

Splunk ES can be deployed both on-premises and in the cloud, offering flexibility and scalability to meet different organizational. The adopted model influences the pricing for Splunk ES [30]:

Ingest pricing Based on the amount of data ingested into Splunk products.

Workload pricing Based on the compute and storage resources required to run workloads.

Entity pricing Based on the number of hosts using Splunk observability products.

Activity-based Pricing Based on metric time series (MTS), traces analyzed per minute, sessions, or uptime requests.

Although concrete values were not found in their official publicly available resources, there is information on the Internet from individuals who claim their organizations use Splunk products and reveal their annual cost. Still, as the sources can't be verified, these values will not be considered.

5.2.2 QRadar

QRadar[4] is the SIEM solution of International Business Machines (IBM) that provides situational awareness and compliance support. Like Splunk, it is also considered to be a leader and visionary in the SIEM industry as seen in Fig. 5.1. The most notable features of QRadar are:

Risk-based alert prioritization Like Splunk, risk values are assigned to assets with the help of AI, and alerts are based on this value, ensuring that security analysts focus first on the most critical alerts.

Sigma community rules QRadar supports the usage of thousands of Sigma rules[5] which is a good source of OSINT.

Federated search Federated searches allow for a cost-effective measure to choose which data to ingest into the SIEM but still have the option to search from all data wherever it is stored from a single interface.

Similarly to Splunk ES, QRadar can be deployed either on-premises or on multiple public cloud platforms. QRadar offers a choice of subscription or perpetual licensing models. The cost of perpetual licensing is not displayed on their publicly available information sources. If a subscription model is adopted, the price range is €940 - €1,400[6] per month taking into account the number of workstations/employees and servers in the infrastructure. This increases to €4,039

[4] `https://www.ibm.com/products/qradar-siem`

[5] `https://graylog.org/post/the-ultimate-guide-to-sigma-rules/`

[6] `https://www.ibm.com/products/qradar-siem/pricing` - Calculations made for 1000 workstations and 100 servers

- €5,770 if deployed using the cloud. There is also a third option of using their cloud-native SIEM which calculates the price based on the amount of data ingested. The estimated cost is €6300/month for 100GB of daily ingestion.[7] The cost per Gigabyte (GB) decreases as the daily ingestion volume increases.

5.2.3 SumoLogic

Sumologic[8] is a cloud-based data analytics tool with a focus on security that also has its own cloud-native SIEM. It is also present in the Gartner Magic Quadrant. The more prominent features of SumoLogic's SIEM are:

MITRE ATT&CK coverage explorer Like Splunk, it maps adversary Tactics, Techniques, and Procedures (TTP) covered by detection rules to the MITRE ATT&CK framework to identify strengths and weaknesses in organizations' networks.

Signals and Insights An adaptive Signal[9] clustering algorithm is used to automatically group related Signals to accelerate alert triage. If the risk of aggregated Signals surpasses a threshold, an Insight[10] is generated to help the security team determine the most important threats.

Entity Relationship Graph Correlated events are available for viewing through an interconnected graph that facilitates understanding of threats as investigating isolated Signals is difficult.

Built-in automation and playbooks Built-in automation allows the automatic enrichment of data to add further context and the use of playbooks allow teams to quickly respond to threats.

Sumo Logic's cloud SIEM, being cloud-native, can only be deployed in the cloud. There is no price for ingested data as the cost model is based on the volume of data scanned during queries performed for data analysis. For a medium-sized enterprise, the estimated price is approximately $3.09 per TeraByte (TB) of scanned data.[11]

5.2.4 InsightIDR

InsightIDR[12] is Rapid7's cloud SIEM solution. Just like the previous solutions, it is also recognized in the Gartner Magic Quadrant for SIEM. Its existing features include:

EDR InsightIDR has its own EDR solution which is easily integrated with its SIEM solution and can be used to increase visibility and protect assets in the infrastructure.

Enhanced Network Traffic Analysis (ENTA) With its ENTA feature, critical network visibility is gained by gathering traffic metadata (both at the edge and inside the network), in a human-readable format and using it in conjunction with its own IDS to detect threats.

[7] https://www.ibm.com/products/qradar-cloud-native-siem/pricing
[8] https://www.sumologic.com/
[9] Individual security event.
[10] Higher-level, aggregated view that consolidates multiple related signals.
[11] Sumologic Pricing
[12] https://www.rapid7.com/products/insightidr/

Integrations InsightIDR offers an extensive range of third-party integrations, like AWS[13], that enhance its built-in capabilities for monitoring endpoints, networks, and user activities.

MITRE ATT&CK Alignment Just like the former options, threats are mapped to the MITRE ATT&CK framework.

Deception technology With its deception technology, organizations can create their own honeypots, honey users, honey credentials, and honey files to capture the attention of attackers and identify them.

Response and automation Like some of the former solutions, InsightIDR, provides various automation features, such as out-of-the-box workflows for containing endpoint threats, suspending user accounts, and integration with ticketing systems.

InsightIDR is fully cloud-based, meaning it is impossible to deploy on-premises. Its pricing model works as an annual Software as a Service (SaaS) subscription with three different tiers. InsightIDR Essential begins at $3.82 per asset per month, InsightIDR Advanced begins at $6.36 per asset per month and InsightIDR Ultimate begins at $8.21 per asset per month, in the United States (US). The price is based on environments with 250k assets.[14] No price was found for Europe.

5.2.5 Wazuh

Wazuh[15] is an open-source unified SIEM+XDR solution that is recognized for being a fork of OSSEC[16] HIDS and being a solution that organizations can use and modify freely to fit their needs. Its capabilities have won them an award[17] for being the best SIEM solution of 2023. Among Wazuh's capabilities, one can find:

Vulnerability detection With the use of agents, information about installed packages in the host system can be analyzed to check for vulnerabilities that are categorized in severity, by using CVSS 3.0, to help teams patch critical systems and vulnerabilities promptly.

Integrations Wazuh can be integrated with other platforms to aid in a variety of purposes such as notification (Slack[18]), threat intelligence, improved data visualization platforms (Grafana[19]) or event enrichment.

Security Configuration Assessment Wazuh is capable of analyzing system configurations to ensure they meet security standards using out-of-the-box policies or custom policies defined

[13] https://aws.amazon.com/
[14] https://www.rapid7.com/products/insightidr/packages/
[15] https://wazuh.com/
[16] https://www.ossec.net/
[17] https://www.scmagazine.com/news/sc-award-winners-2023-wazuh-best-siem-solution
[18] https://slack.com/
[19] https://grafana.com/

by organizations. They can be used to help identify and address vulnerabilities, misconfigurations, or deviations from best practices and security standards.

Regulatory compliance Wazuh can simplify regulatory compliance by including rules that align with PCI DSS, NIST 800-53, GDPR, TSC SOC2, and HIPAA guidelines. This allows to verify which rules are impacting compliance with these regulations.

Automated response Wazuh agents can automatically respond to threats with out-of-the-box response mechanisms at the host level. Additional responses can be developed to fit the organizations' needs.

XDR With its XDR capabilities, Wazuh improves existing HIDS mechanisms to detect malware, perform file integrity monitoring, read endpoint telemetry, and automatically respond to threats further increasing the visibility of the infrastructure,

As Wazuh is a free open-source solution, deploying it on-premises has no cost in terms of acquiring a license for the software. If deployed in the cloud, for a small environment of 100 active agents the plan costs $571/month. Up to 250 active agents cost $923/month. A large environment handles up to 500 active agents and the plan costs $1449/month.

5.2.6 Security Onion

Security Onion[20] is another open-source framework that can be used as a SIEM. However, Security Onion acts as an aggregator that utilizes multiple different tools like Suricata[21] IDS, Elastic Agents[22] and Zeek[23] into a combined security platform. Its features include:

Detections Teams can manage all detection rules (Suricata, YARA, and Sigma) in a single interface and have the ability to import publicly available community rules or create custom rules.

PCAP Analysis Captured PCAP files can be analyzed in security onion to gain further insight into a threat without relying on other tools, although the option is available.

Cases Incident response dashboard is available to respond to threats efficiently.

Like Wazuh, Security Onion is a free open-source solution, having no costs associated with the acquisition of a license to use the software on-premises. It can also be deployed on a variety of clouds like Amazon, Azure, and Google with varying costs and usage models dependent on each cloud provider.

[20] https://securityonionsolutions.com/
[21] https://suricata.io/
[22] https://www.elastic.co/elastic-agent
[23] https://zeek.org/

Functionalities/SIEM	Splunk Enterprise Security	QRadar	SumoLogic	InsightIDR	Wazuh	Security Onion
Log Collection and Aggregation	✓	✓	✓	✓	✓	✓
Event Correlation	✓	✓	✓	✓	✓	✓
Alerting and Notification	✓	✓	✓	✓	✓	✓
Dashboards and Reporting	✓	✓	✓	✓	✓	✓
User and Entity Behavior Analytics	✓*	✓	✓	✓*		
Threat Intelligence Integration	✓	✓	✓	✓*	✓	✓
Compliance Management	✓	✓	✓	✓	✓	
Log Retention and Archiving	✓	✓	✓	✓	✓	✓
Native EDR/XDR integration		✓*		✓*	✓	

Table 5.1: Common functionalities SIEM tools provide. *additional cost

5.3 Comparison between Solutions

Table 5.1 compares the availability of the most common features in SIEMs that were stated in Sec. 2.2.4.

Both QRadar and InsightIDR gain the advantage on the most common functionalities provided by the tools. However, this is insufficient to make a decision. Since each solution has different additional functionalities or the same functionality with a different name, these will also be accounted for in the decision-making process. From each solution, the more attractive features of each solution are:

- Splunk
  - Threat Topology
  - Risk Based Alerting
  - Adaptive Response Actions
  - Enterprise Security Content Updates and Use Case Library
- QRadar
  - Risk-based alert prioritization
  - Federated search
- SumoLogic
  - Signals and Insights
  - Entity Relationship Graph
  - Built-in automation and playbooks
- InsightIDR
  - Endpoint Detection and Response
  - Enhanced Network Traffic Analysis
  - Deception technology

SIEM/Deployment Type	On-premises	Cloud
Splunk Enterprise Security	✓	✓
QRadar	✓	✓
SumoLogic		✓
InsightIDR		✓
Wazuh	✓	✓
Security Onion	✓	✓

Table 5.2: Deployment types of SIEMs

SIEM/Deployment Type	On-premises	Cloud
Splunk Enterprise Security		
QRadar	€940 - €1,400/month*	€4,039 - €5,770/month*
SumoLogic		$3.09/TB
InsightIDR		$6996/month*
Wazuh	$0	$2900/month*
Security Onion	$0	Cloud Provider dependent

Table 5.3: Deployment types cost of SIEMs. *Calculations made with 1100 assets.

  - Response and automation

- Wazuh
  - Vulnerability detection
  - Security Configuration Assessment
  - Automated response
  - Extended Detection and Response

- Security Onion
  - Network packets Analysis
  - Cases

Each solution offers functionalities that can greatly benefit CIÊNCIAS ULisboa in the future. However, it is important to manage expectations, as leveraging these features effectively requires staff to learn and operate them proficiently. While many features are valuable, attempting to use multiple functionalities without mastering any of them can become a weakness.

Table 5.2 summarizes the deployment types that can be made for each solution. The advantage goes to Splunk, QRadar, Wazuh, and Security Onion as on-premises is preferred.

Concerning cost, as observed in Table 5.3, Security Onion and Wazuh have the advantage as they are free to use. This promotes longevity in their usage.

After this comparison of various SIEM tools, assessing their features, capabilities, and longevity for CIÊNCIAS ULisboa, Wazuh was proposed and selected as the most suitable option that aligns with CIÊNCIAS ULisboa's current organizational needs and fits well for a project without having

significant downsides. If this project demonstrates good results, it could pave the way for acquiring more advanced solutions or improving other aspects of existing systems, with a focus on their integration with Wazuh. This success could lead to further investments in enhancing the overall security infrastructure.

Chapter 6

Implementation

This chapter will provide a comprehensive overview of the implementation of the Wazuh Security Information and Event Management (SIEM) framework on the infrastructure of CIÊNCIAS ULisboa. It will cover the framework's initial capabilities and the upgrades that have been made to enhance it, the challenges encountered during implementation, and the solutions employed to overcome these challenges.

6.1 Wazuh

6.1.1 Architecture

As depicted in Fig. 6.1, Wazuh has 4 main components:

Agent Installed on the endpoints, protects them against threats and sends information to the server.

Server Analyzes the data received from agents and processes it with decoders, rules, and threat intelligence to find indicators of attack/compromise. It also updates, manages and configures agents.

Indexer Indexes alerts produced by the server(s) for fast and accurate information retrieval.

Dashboard User interface for data visualization, analysis, and Wazuh configuration. Includes several dashboards depending on the functionality used and allows for the creation of custom ones.

This architecture allows for Wazuh to be scalable. For additional processing power, server nodes can be added, with a load balancer placed between the agents and nodes of the Wazuh cluster, to ensure even distribution of the workload. To improve indexing and search speeds, additional indexer nodes can be integrated into the architecture. Agents communicate with the servers using a proprietary protocol. By default, messages are encrypted by AES with 256-bit keys.

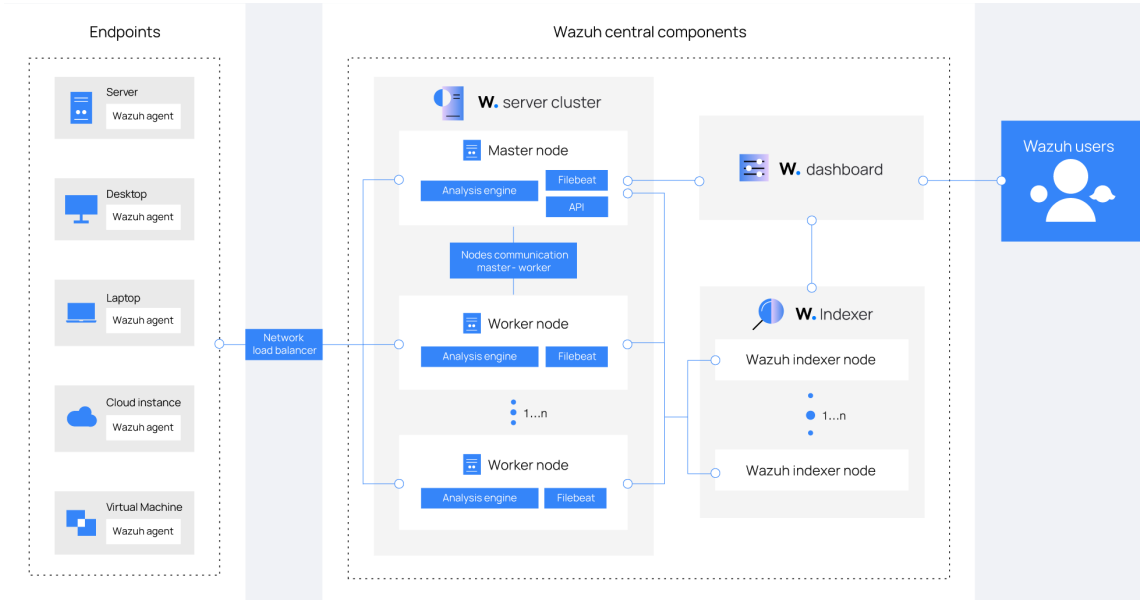

Figure 6.1: Wazuh architecture. [35]

Devices such as firewalls, routers, and switches, where it is not possible to install agents, can also send information to the Wazuh server via Syslog[1] or be monitored via Secure Shell[2] (SSH) by executing commands and sending the output to the server.

6.1.2 Configuration

Wazuh was configured, by following its extensive documentation.[3] Installation and configuration can be done manually or by using the Wazuh-provided script, which automates the initial setup. To ensure a comprehensive understanding and identify any differences between the methods, both approaches were tested. Ultimately, the script was used to minimize human error and expedite the process, leveraging its efficiency for a smoother setup experience. After the initial setup, to tailor the setup to our specific needs and requirements, the following additional configuration steps were undertaken:

1. Configure server configuration file to enable/disable features. In particular, it was decided to enable agent authentication to ensure that only agents with the correct passkey could communicate with the Wazuh server.

2. Configure agent groups. In our configuration, agents were grouped according to the objectives of their host as presented in Fig. 3.1 (p.18).

3. Create passkey for agent authentication.

4. Open firewall ports required for communication with Wazuh agents.

[1] https://www.logicmonitor.com/blog/what-is-syslog
[2] https://www.cloudflare.com/learning/access-management/what-is-ssh/
[3] https://documentation.wazuh.com/current/index.html

5. Change default Wazuh indexes and logs storage directory to one on a separate partition to avoid filling space on the root partition that could halt server execution.
6. Install agents.
7. Configure the maximum events per second (EPS) sent by agents based on the activity level of each host. For high-traffic hosts like reverse proxies, the maximum EPS was increased from the default 500 to 1000. For less active hosts such as the development server, the EPS limit was reduced to 100.
8. Configure information sources and File Integrity Monitoring (FIM) in agents.
9. Configure decoders and rules on the Wazuh server.

Wazuh's configuration is primarily managed through XML files, using various tags and attributes to define settings and behavior. The more relevant tags and attributes will be explained in detail within their respective subsections.

6.1.3 Information sources

Most of the information sources were integrated using Wazuh agents. Work for the integration of information sources continues but in the scope of this project, the primary focus was on collecting data produced by web services and security tools. The former because they are the most vulnerable and active services in CIÊNCIAS ULisboa. The latter because security tools are expected to produce the most relevant alerts.

The current CIÊNCIAS ULisboa information sources integrated into Wazuh have been graphically depicted in Fig. 3.1 (p.18). Agents were placed in the reverse proxies, as most of the traffic goes through them, and some of the internal systems as a test to verify if Wazuh can make use of its data sources. Current information sources include:

Web server logs Produced by web server software such as Apache and Nginx at the reverse proxies and backend servers. Their access logs provide important information such as the client IP, requested resource, and payload data.

Web Application Firewall (WAF) logs WAF logs contain key information such as the IP addresses, attacked domains, and context for packet rejection, such as missing headers. WAFs monitor traffic at the reverse proxies.

Custom web application logs Custom logs generated by the web applications running in the backend servers. They allow CIÊNCIAS ULisboa to identify malicious activity at the application layer and complement the information received from the web server logs.

Snort alerts Provide detailed information about potential malicious activity, including the actions attempted and the IP addresses involved which contribute in the identification of systems that were targeted. These alerts are generated by the IDS whose implementation was reported in Ch. 4.

Next-Generation Firewall (NGFW) IDS alerts Alerts generated by the IDS of CIÊNCIAS ULisboa's NGFW. The alerts content and structure is similar to the fields present in Snort alerts. Since an agent cannot be placed in the NGFW, alerts generated from its IDS are delivered to the Wazuh server via Syslog.

System logs Various logs from Linux and Windows services, such as authentication logs, audit logs, and Windows Registry keys. These logs are used to detect anomalies, such as unusual login attempts, suspicious processes, or the installation of malicious software. These are retrieved from every host a Wazuh agent is deployed in.

Several configuration options can be used for each information source to ensure that Wazuh collects and interprets the information correctly. The ones used for this project are, "location" and "log_format"; which are grouped under a "localfile" tag to set each information source.

location Specifies from where to read events at the monitored host. It can be a path to a log file, a Windows event channel, the journald system etc.

log_format Indicates to the agent the format of the log to be read. Examples include syslog, JSON, eventlog, journald, etc.

Listing 6.1 shows the XML excerpt that integrates Snort alerts after being processed by the "Alerting" scripts. These write the data in a syslog-like format into a log file, "/var/log/snort/snortScripts/complete_data.log", to facilitate integration.

```
<localfile>
    <location>/var/log/snort/snortScripts/complete_data.log</location>
    <log_format>syslog</log_format>
</localfile>
```

Listing 6.1: Integration of Snort log file.

6.1.4 Decoders

Configured in the Wazuh server, decoders are used to process data incoming from information sources. Decoders act as parsers, using regular expressions to extract relevant data and assign it to the appropriate fields. Normalizing these fields is crucial for enabling effective correlation and analysis. For certain data sources, such as JSON, the decoding step can be bypassed since JSON is inherently structured. Decoders also use a parent-child mechanism in which information is passed to child decoders only if it matches the parent decoder's pattern. This allows multiple log formats to be processed for the same program. At the XML configuration file, the various tags and attributes used for decoders are:

decoder Acts as the root element of a decoder file, encompassing the full definition of a decoder. This includes its name, type, and the specific attributes that determine how it processes and extracts information from log messages.

program_name Specifies the program name that must be present in the log header for the decoder to be applied. This only applies to logs with Syslog-like headers.

parent Used to link a subordinate decoder to its parent. A parent decoder can have many child decoders but a child decoder cannot be a parent.

regex Serves to program regular expressions to extract information from events.

order Specifies the sequence of fields to where the extracted data from events is inserted.

name Assigns a unique name to the decoder.

offset Allows to discard some of the content of the event and only apply the regular expression after a certain point. "after_parent" applies the regular expression to the rest of the event after the matches made in the parent decoder.

Listing 6.2 presents the configuration file excerpt used to decode information from enriched Snort logs. The "program_name" instructs Wazuh to use the "snort" decoder only if the program name in the event matches "Snort". Once this condition is met, the data is passed to the corresponding child decoder, "snort_child". The "snort_child" decoder uses regular expressions to extract data from the log and assigns the extracted values to the correct fields in the order specified by the "order" tag.

```
<decoder name="snort">
    <program_name>^Snort</program_name>
</decoder>

<decoder name="snort_child">
    <parent>snort</parent>
    <regex offset="after_parent">(\d+)\|(\d+)\|(\d+)\|(\.+)\|(\.+)\|(\.+)\|(\S
        +)\|(\S+)\|(\S+)\|(\S+)\|(\S+)\|(\S+)\|(\.*)\|(\.*)\|(\S*)\|(\S*)</
        regex>
    <order>snort.gen_id, snort.sig_id, snort.rev_id, message, classification,
        priority, protocol, srcip, srcport, dstip, dstport, interface, snort.
        url, snort.cve, domain, url</order>
</decoder>
```

Listing 6.2: Snort Decoder Configuration.

6.1.5 Rules

Rules are a set of conditions that specify the criteria for triggering various alerts. They can search for patterns or information in events to evaluate which alert to trigger. These alerts and their data will then be logged for visualization and analysis.

These rules are programmed in the Wazuh server where the following tags and attributes are used:

rule Tag to start the block defining a rule.

decoded_as Used as a requisite to trigger a rule. It will be triggered if the event has been decoded by a certain decoder.

description Specifies a human-readable description of the rule to provide context to each alert.

if_sid Used as a requisite to trigger a rule. This option matches if the log has previously matched a rule with the specified ID. Rules can have as many descendants as necessary.

if_matched_sid Matches if an alert of the specified ID has been triggered within a predefined frequency and timeframe.

same_srcip Specifies that the decoded source IP address must be the same as the rule matched in "if_matched_sid". This option is used in conjunction with frequency and timeframe.

frequency Number of times the rule must match before generating an alert.

timeframe A value specified in seconds that defines the timeframe within which alerts are matched.

id Assigns unique ID to identify the rule.

level Assigns a severity level to an alert. The higher the level, the more severe. Severity levels help convey the criticality of each alert more effectively. Rules from 0 to 2 do not generate alerts but can be used to trigger other rules. Wazuh has its own model[4] to define the severity of its alerts which is also used by CIÊNCIAS ULisboa.

As a standard practice for rules and decoders, a main parent is used to group all types of alerts related to a specific program as seen in the decoder with the name "snort" depicted List. 6.2 and the rule with id 100601 in List. 6.3. This facilitates the management of writing multiple decoders and rules for the same program.

Listing 6.3 provides an example of a rule that triggers alerts in Wazuh for Snort logs decoded by the "snort" decoder. When a Snort log is received and is processed by the Snort decoder, it matches the criteria specified in rule 100601. Rule 100603 is designed to trigger when rule 100601 is matched. Once triggered, rule 100603 generates an alert with a level 6 severity and the description "Snort IDS Event."

```
<rule id="100601" level="0">
    <decoded_as>snort</decoded_as>
    <description>snort: Messages grouped.</description>
</rule>

<rule id="100603" level="6">
    <if_sid>100601</if_sid>
    <description>Snort IDS Event.</description>
</rule>
```

Listing 6.3: Snort Rule Configuration.

[4] https://documentation.wazuh.com/current/user-manual/ruleset/rules/rules-classification.html

Frequency counters can be employed to escalate the severity of repeated actions as demonstrated in List. 6.4. In this example, the rule with id 100607 is only triggered when rule 100603 is also triggered 10 times within an hour and the decoded source IP is the same.

```
<rule id="100607" level="10" timeframe="3600" frequency="10">
    <if_matched_sid>100603</if_matched_sid>
    <same_srcip />
    <description>Multiple Snort Scripts IDS Event from same source IP.</
        description>
</rule>
```

Listing 6.4: Snort Rule for Multiple Events from Same Source IP.

6.1.6 File Integrity Monitoring

Wazuh includes File Integrity Monitoring (FIM) capabilities, which are crucial for tracking changes to files, directories and registry keys on endpoints. This functionality is particularly important for detecting unauthorized modifications or hidden tools that attackers might deploy.

As discussed in Sec. 3.3.2, CIÊNCIAS ULisboa hosts several web applications for its members, which may be susceptible to vulnerabilities. FIM was configured to allow CIÊNCIAS ULisboa to monitor the files these web applications use, ensuring that any unauthorized changes are quickly identified. In the event of a modification, affected web applications can be rapidly identified for remediation.

The configuration is done at the agent level. Some configuration settings include:

syscheck Acts as the root element of a system check section in the agent configuration file.

directories List of directories to be monitored. Wildcard characters (? and *) can be used to monitor paths that fulfill the given pattern.

recursion_level Specifies the maximum depth of recursion allowed. If a monitored directory contains subfolders or additional directories, recursion_level determines the deepest level to be analyzed.

check_all Defines if all the available hashing algorithms in Wazuh for FIM (MD5, SHA-1 and SHA-256) should be used.

report_changes Defines if FIM should report file changes.

Listing 6.5 is one of the created configurations to monitor directories in one of the web hosting servers. It hashes all of the files in "/home/httpd/<domain>/public_html/" with a max depth of 2 with MD5, SHA-1 and SHA-256 and reports if any changes have been made to the files by recalculating the hash and comparing both values.

```
<syscheck>
    <directories recursion_level="2" check_all="yes" report_changes="yes">/home
        /httpd/*/public_html</directories>
</syscheck>
```

Listing 6.5: Syscheck Configuration

```
File '/etc/snort/snort3/intel/ip-blocklist' modified
Mode: scheduled
Changed attributes: size,mtime,md5,sha1,sha256
Size changed from '21833' to '21761'
Old modification time was: '1725058502', now it is '1725144902'
Old md5sum was: 'c74633eedf80261cec9df19ac54d52ef'
New md5sum is : '3b04de9f19bd41b2ff0bdbaf78447e66'
Old sha1sum was: 'b5ea2549ec33e2edff48ba59a97643897de06219'
New sha1sum is : 'e7d7828fcd827480f79fbfca2a7c1afe011ff885'
Old sha256sum was: '0f3b89c890f51c12589ff1c1094075f6dfbb3667a44a9b34225dae3f0bcdc249'
New sha256sum is : '4d88958b6ed422accfafb9c973a01abffed0a1f2e598767c2d1347771bf08cde'
```

Figure 6.2: Full log of an alert that detected the update of Snort's IP blocklist.

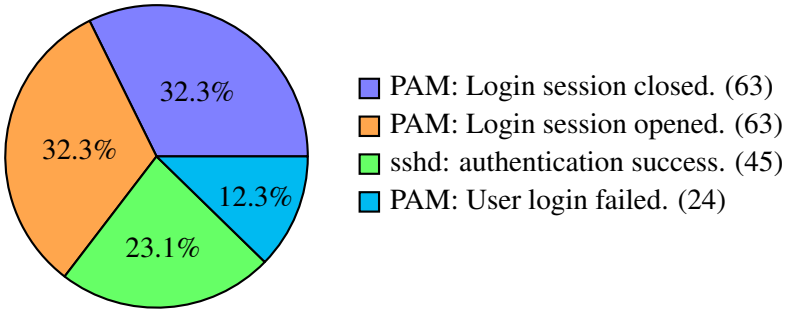

Figure 6.3: Alerts generated by the backup service in 1 month.

As an example of an alert, Fig. 6.2 demonstrates the full log of an alert triggered when Snort's IP blocklist was updated.

6.1.7 Initial observations

Alerts triggered by internal tools Determining what was relevant in the existing logs was initially challenging. However, over time, the more pertinent rules and which ones could be silenced to reduce alert fatigue started being identified. A notable instance involved alerts being triggered for authentication activities on Linux hosts, summarized in Fig. 6.3, that were not expected to be accessed frequently. The cause of these alerts was a software application that performed backups of the hosts during certain time frames. To address this, a rule was created so that no alert was triggered during these time frames and activity. This effectively silenced these alerts, ensuring that the backup-related authentications no longer trigger unnecessary alerts. While this decrease in alerts is not very noticeable compared to the total number of alerts generated daily, silencing specific alerts made it easier to detect genuine anomalies.

As a result of the previous observation, it was considered that internal hosts attacking each other pose significant risks. Testing revealed that the existing vulnerability management system, presented in Sec. 3.4.4, triggered multiple events on an internal host. These events triggered a variety of alerts as highlighted Fig. 6.4. This allowed CIÊNCIAS ULisboa to comprehend the type of alerts that should be looked out for if internal hosts start attacking each other.

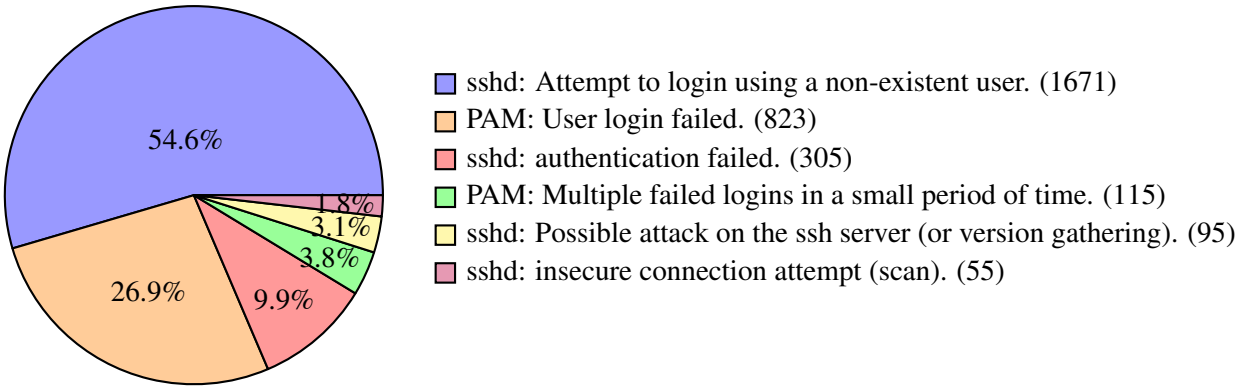

Figure 6.4: Alerts generated from system events on one host in 2 hours.

Alerts generated by external threats Figure 6.5 graphically represents multiple different alerts that originated from the same source IP. It should be noted that 1 information source, such as web server logs can trigger different types of alerts such as "SQL injection attempt.", "Suspicious URL access", and "XSS (Cross Site Scripting) attempt." depending on the payload.

From the many distinct alerts, a key one observed is "A web attack returned code 200 (success)". It could indicate successful data extraction or another type of successful attack. Investigating the utilized payload, that was also logged, in these situations helped assess if an attack was truly successful or benign.

Though not considered attacks, web scrapers/crawlers generate unwanted traffic that often goes unnoticed. This traffic usually occurs gradually over time, without triggering immediate red flags. However, it does generates a considerable amount of alerts as illustrated in Fig. 6.6.

Based on the observations that confirm the occurence of attacks, it became clear that implementing an active response mechanism would be beneficial to protect systems. As a side effect, the active response mechanism would also reduce the number of triggered alerts, and consequently alert fatigue.

Enhanced Visibility By analyzing the traffic generated by the top source and destination IPs, it was observed that a significant number of alerts originated from internal IPs, specifically between the reverse proxy and the internal server. This occurs because traffic analysis primarily takes place between the reverse proxy and internal servers where the traffic is unencrypted. A key issue identified is that the reverse proxy IP replaces the original source IP. To address this, an existing logging and monitoring system addressed in Sec. 6.2.1, was utilized to enhance data enrichment.

With the enhanced visibility provided by Wazuh and the integrated information sources, monitoring more crucial events such as Snort alerts became more efficient. This improvement addressed numerous issues, detailed in Sec. 4.3. The ability to visualize and filter data in various ways allowed for the reactivation of email notifications to users, which had been previously disabled due to the aforementioned issues.

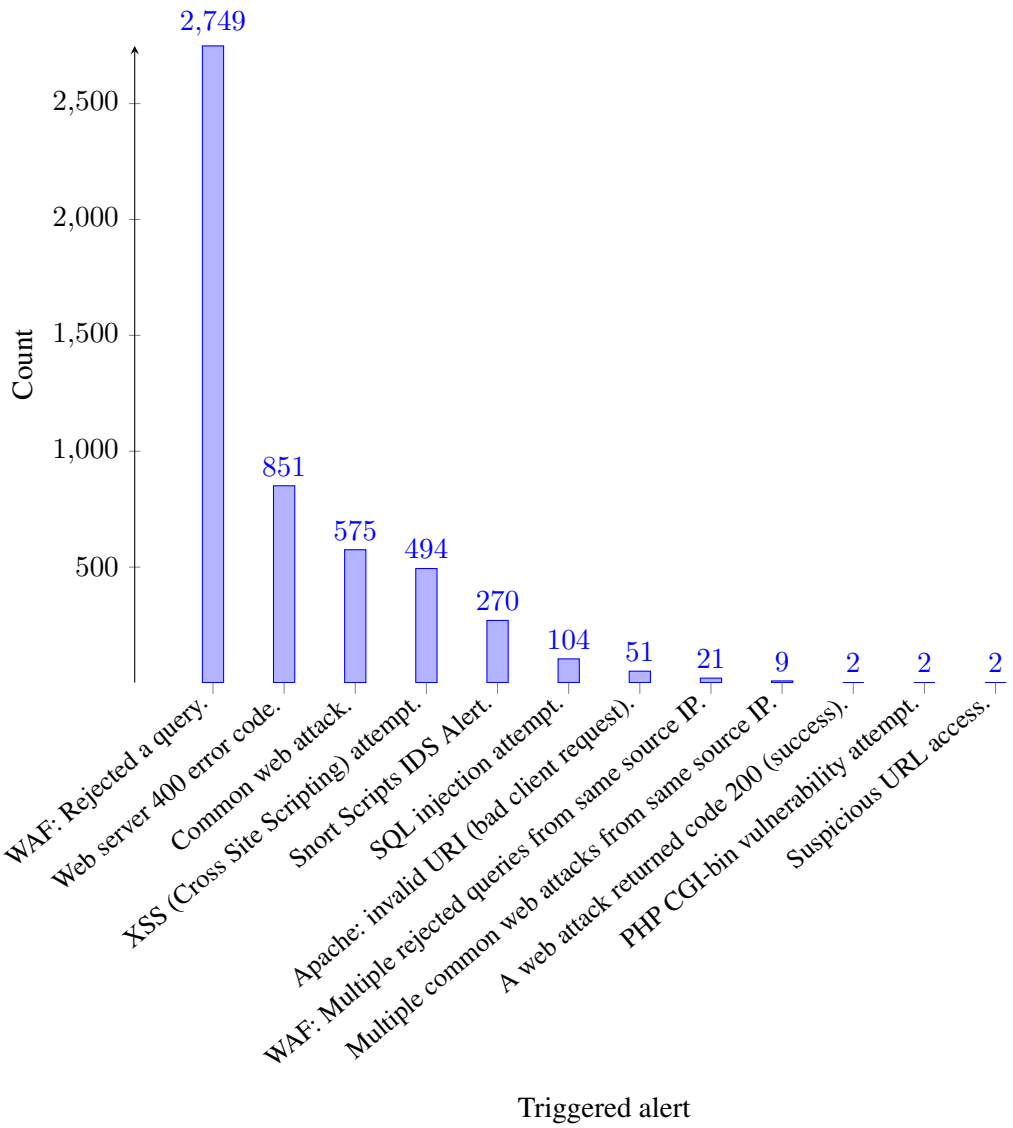

Figure 6.5: Alerts triggered by the IP 200.141.130.162 from February 16th to 20th.

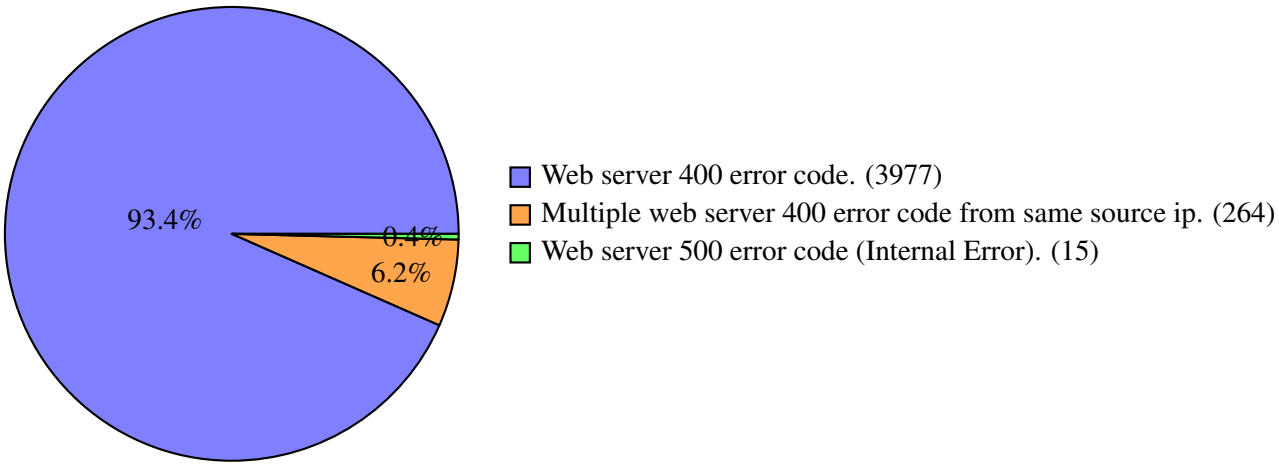

Figure 6.6: Total alert count produced by the most active web crawler in one week.

Correlation and Data Enrichment Although the same information source can trigger distinct alerts, as previously noted, improved context comes from using diverse sources. For example, Snort alerts might not provide the targeted domain of an attack, but by correlating them with data from the Web Application Firewall (WAF), CIÊNCIAS ULisboa can identify the target. The limitations of each information source highlight the importance of data enrichment as a valuable feature. It allows security teams to enrich events with further context which aid in correlation, filtering and sorting of information.

6.2 Scripts

Wazuh provides some scripts for active response mechanisms at the host level. However, they were considered insufficient for CIÊNCIAS ULisboa. In the scope of this project, some scripts were developed for alert enrichment and blocklist generation expanding the functionalities available in Wazuh.

6.2.1 Alert data enrichment

The goal of the alert data enrichment script is to add more data to an alert by using other information sources to improve filtering mechanisms and obtain more context for an alert. The script was implemented using the Wazuh Integrator module. This module forwards alert data to other programs or services, such as a ticketing system. It can also forward alerts to scripts for enrichment and then reintegrate the enriched alerts into Wazuh.

The main issue with this approach is that the initial alert, which is forwarded to the enrichment script, remains logged along with the enriched alert. This leads to redundant data. To solve this issue, a cron job was implemented to remove the initial alerts every 30 minutes, keeping only the enriched alerts in the Wazuh indexer for visualization and analysis purposes.

Currently, only Snort and NGFW alerts are being enriched. Some added information includes fields such as:

Hostnames Facilitates understanding of which systems are targeted without relying only on IP addresses, as a host can have multiple IP addresses, or domain name fields, which may not be available as only specific applications record this data in their generated events.

Network Segment Allows the filtering of specific source and destination networks to understand problematic or non-standard communications.

User Identifying users helps CIÊNCIAS ULisboa monitor its users for malicious activity and notify them directly.

Previously, some alerts incorrectly listed the reverse proxy IP as the source IP, complicating data correlation. An existing monitoring solution, “Monitor” (Fig. 3.1), already collects custom logs that include the actual source IPs and targeted domains. However, the extracted data from the logs was not properly normalized, making correlation difficult.

By querying "Monitor" and using the reverse proxy IP and source port from the alert as filters, the targeted domain and actual source IP can be accurately identified. The retrieved information from "Monitor" is placed in the correct fields of the Wazuh alert to normalize data, creating a more accurate alert that is able to be correlated.

6.2.2 Blocklist generation

As previously mentioned, Wazuh has an active response mechanism that uses existing scripts that operate at the host level. The Wazuh server instructs the hosts to invoke these scripts when specific alerts occur, performing a variety of actions, such as blocking IPs on the host firewall.

This type of functionality can prevent significant damage. When an attacker tries to exploit a vulnerability on a host, currently implemented measures can detect the attempt but cannot determine its success with certainty. Having a service that automatically blocks further exploitation attempts on a host significantly lowers the chance of successful exploitation.

For CIÊNCIAS ULisboa, if alerts are triggered by the IDSs or other tools, blocking these at the host level would still allow for alerts to be triggered at the network layer, contributing to alert fatigue.

A more effective solution would be to block the IPs at the perimeter firewall, completely cutting off communication. However, blocking IPs without first determining if they are malicious could result in inadvertently blocking critical services. A better approach is to verify the IP reputation before taking action.

To determine the reputation of an IP, AbuseIPDB[5] was utilized. As a test, all unique IPs that triggered alerts were sent to AbuseIPDB for reputation verification every hour, without blocking them. The findings from this test will be presented in the next chapter.

To avoid exceeding query limits to AbuseIPDB, a cache mechanism is also utilized. This enables the storage of previously queried IPs reducing the need to query AbuseIPDB for the same IPs repeatedly. This mechanism also proves useful to re-assess already verified IPs that have expired from the cache, as these can become malicious or benign over time.

[5] `https://www.abuseipdb.com/`

Chapter 7

Evaluation

This chapter discusses the most significant findings from the implementation of the measures outlined in this project. It will detail how results evolved throughout the project as a consequence of continuous fine-tuning, and it will highlight the improvements achieved in terms of security posture and observability.

7.1 Before & After Implementation

To measure the impact of this project on CIÊNCIAS ULisboa, it is essential first to determine whether the original objectives were attained and to identify the changes or observations that resulted from their accomplishment.

Implement a SIEM tool to meet the specific needs of CIÊNCIAS ULisboa Wazuh was implemented successfully and with collected data, various insights were obtained. It meets CIÊNCIAS ULisboa's current requirements as discussed in Sec. 5.3 and also provides versatility for improvements to meet future needs, as detailed in the conclusions.

Centralize security alerts in a single platform to facilitate correlation This was achieved and led to a better understanding of which events are important for cybersecurity monitoring. Over 8,000,000 alerts were generated in the course of 7 months and their analysis prompted the development of mechanisms, such as the blocklist generation scripts, that would reduce the amount of triggered alerts.

Enrich existing event data with additional context This was accomplished with the alert data enrichment scripts, providing versatility in structuring and filtering data to enable a focused analysis of specific aspects.

Configure endpoint monitoring Basic monitoring was implemented with extended features left as future work, as some options required changes at the endpoint. The existing monitoring from Wazuh and additional configuring done proved to be effective for detecting some attacks as illustrated in the previous chapter in Sec. 6.1.7 and future attacks that might provoke file/directory modifications (Sec. 6.1.6).

Correlate security alerts from the network and systems to understand the full scope of an attack By combining data from both sources, more context and understanding were achieved, as the information from one source complements the other. For example, both Snort and the Web Application Firewall (WAF) may detect an attack from the same source IP and port within the same timeframe. Although Snort cannot identify the targeted domain, the WAF can, consequently adding crucial context to the analysis.

Enhance threat detection and response capabilities Wazuh added more visibility and insights into threats and their operations. Although some basic response mechanisms were developed and results were simulated, the simulations show that they can be useful and serve as a strong foundation for the future, as discussed in the conclusions.

Notify users about alerts generated from their own devices and encourage self-resolution Email notifications for some alerts were implemented. The messages include possible solutions to ensure that the support team does not become overwhelmed with help requests.

7.2 Effectiveness

Wazuh, when used alongside various tools such as Snort, was observed to detect and report multiple attack attempts. The detections reported by Wazuh during its operation at CIÊNCIAS ULisboa in the month of July 2024 are shown in Table 7.1. Analyzing this data is crucial for identifying anomalies and understanding the prevalence of different types of attacks. It helps assess whether CIÊNCIAS ULisboa is adequately protected against the most common threats. By analyzing the most triggered alerts, it will aid in enhancing security if needed and provide insight into which alerts can be silenced if they are irrelevant or CIÊNCIAS ULisboa is already sufficiently protected. Additionally, it allows to differentiate between frequently occurring attacks and less common ones, such as "PHPMyAdmin scans (looking for setup.php)." By focusing on these niche attacks, we can enhance our security measures and address vulnerabilities that might otherwise be overlooked.

Knowing what types of attacks are happening is useful. However, identifying the attackers allows for the prevention of further attempts, effectively mitigating the threat in the most rapid way possible. Table 7.2 depicts the top 25 IP addresses involved in the alerts generated in July 2024.

The table shows that a few internal IP addresses (those in the 10.0.0.0/8 subnet) are generating numerous alerts. These alerts were found to be benign web errors caused by missing data in an endpoint. Additionally, most of the external IP addresses belong to the 85.208.96.0/24 and 185.191.171.0/24 subnets. These subnets are associated with web crawling activities that have been flagged as malicious due to excessive crawling (refer to Fig. 7.1 and Fig. 7.2.)

Shifting the focus to the hosts with the highest number of events, it can be observed in Fig. 7.3 that web hosts and reverse proxies lead. This is expected, as they handle the most traffic among all systems in CIÊNCIAS ULisboa. More critical security information sources, such as the IDS (Snort) and NGFW, offer deeper insights into actual threats rather than just crawlers or other less

Event Type	Count
Web server 4xx error code.	1152760
WAF: Rejected a query.	147324
Snort IDS Alert.	93005
Wordpress xmlrpc.php call.	79869
Multiple web server 4xx error codes from same source IP.	63610
Web server 500 error code (Internal Error).	44104
Web server 500 error code (server error).	35889
CMS (WordPress or Joomla) login attempt.	33663
NGFW IDS Alert.	18889
A web attack returned code 200 (success).	7676
SQL injection attempt.	6113
CMS (WordPress or Joomla) brute force attempt.	4003
Common web attack.	2995
Possible Brute-Force on Wordpress Site.	1947
Multiple Snort IDS Event from same source IP.	1389
WAF: Multiple rejected queries from same source IP.	1255
Multiple NGFW IDS Alerts from same source IP.	1197
Suspicious URL access.	866
Multiple web server 500 error code (Internal Error).	776
XSS (Cross Site Scripting) attempt.	721
URL too long. Higher than allowed on most browsers. Possible attack.	284
Multiple common web attacks from same source IP.	269
Multiple SQL injection attempts from same source IP.	184
PHPMyAdmin scans (looking for setup.php).	153
Multiple XSS (Cross Site Scripting) attempts from same source IP.	47

Table 7.1: Top 25 Alerts in July 2024

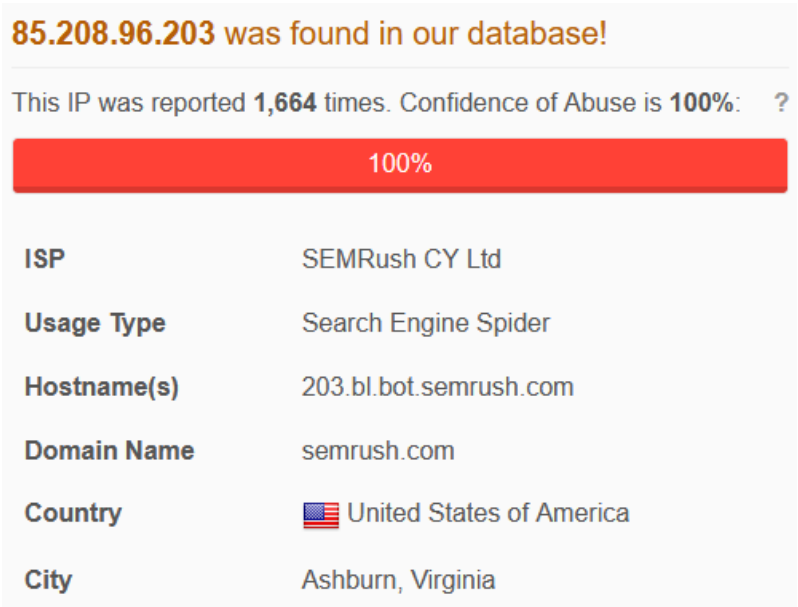

Figure 7.1: AbuseIPDB: Data found for 85.208.96.203.

Source IP	Alert Count
2001:690:21c0:f606::163	84578
68.183.188.120	44314
146.190.106.111	27498
10.121.21.17	25341
10.101.12.11	19992
157.230.38.174	17499
85.208.96.199	17351
85.208.96.200	17312
85.208.96.198	17202
85.208.96.195	16980
85.208.96.197	16619
185.191.171.18	16415
85.208.96.193	16409
85.208.96.202	16318
185.191.171.19	16223
185.191.171.17	16154
85.208.96.196	16154
85.208.96.208	16076
85.208.96.212	16052
85.208.96.209	16042
185.191.171.1	16032
85.208.96.211	16010
85.208.96.201	16007
85.208.96.207	15985
185.191.171.5	15967

Table 7.2: Top 25 Source IPs that generated the most alerts in July 2024

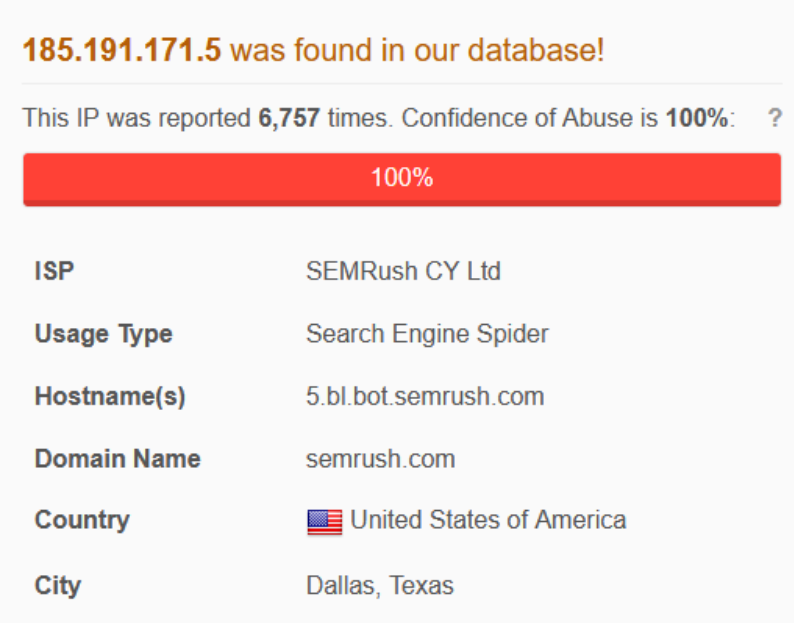

Figure 7.2: AbuseIPDB: Data found for 185.191.171.5.

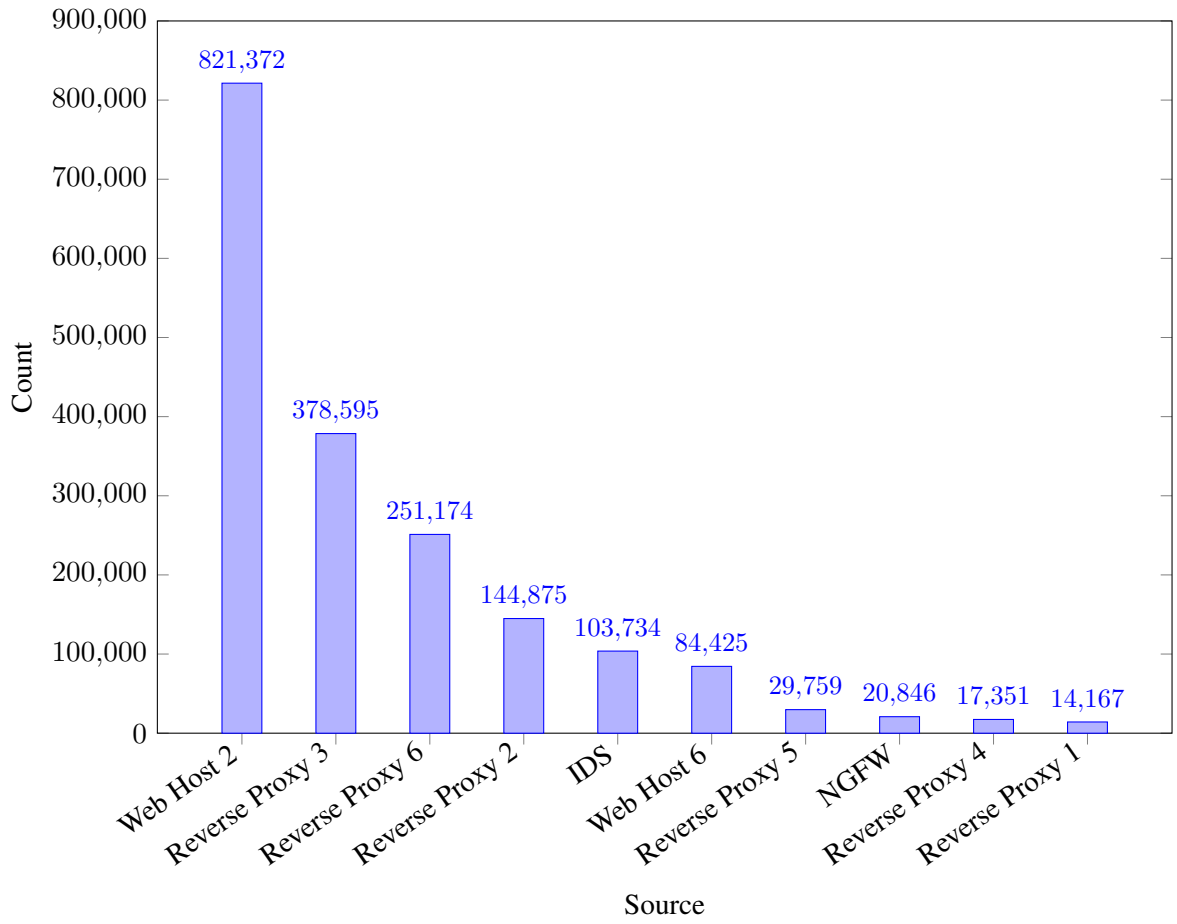

Figure 7.3: Top 10 Agents/Information sources in July 2024

malicious activities. Useful information from IDS and NGFW primarily includes IP addresses and attack data. However, many benefits come from alert enrichment. By applying additional filtering mechanisms CIÊNCIAS ULisboa can focus on key aspects. One of these, includes determining how many of the incoming alerts still originate from reverse proxy IPs directed toward internal servers as this is an issue referred in Sec. 6.2.1.

The implementation of Wazuh also enables CIÊNCIAS ULisboa to better monitor and understand internal vulnerabilities, addressing the limitations in data observation and filtering noted in Sec. 4.3. This improvement allowed CIÊNCIAS ULisboa to re-activate user notifications, providing insight into whether notifying users was sufficient to reduce the alerts originating from their devices. This will be detailed in the next section.

The blocklist generator has proven to be an invaluable tool for determining whether IP addresses are considered malicious. An unexpected but useful outcome was the identification of multiple IPs (listed in Fig. 7.4) that were verified but not detected as malicious. This prompted CIÊNCIAS ULisboa to investigate the reason behind so many alerts incoming from IPs considered non-malicious. Consequently, it was discovered that many Snort alerts were being wrongly triggered, calling for a refinement in its configuration.

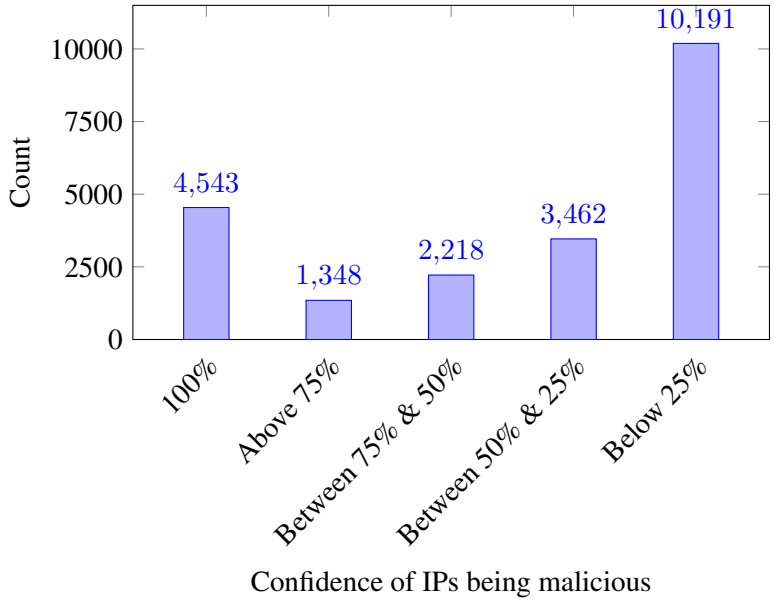

Figure 7.4: Analyzed IP addresses and their confidence levels of being malicious.

The majority of the IPs originate from data centers or web hosting services as illustrated in Fig. 7.5, where individuals or groups can rent systems to launch attacks while masking their real IP addresses. Identifying the true IP address often requires cooperation from the systems' owner and can be challenging without their involvement. This process typically occurs when attacks are successful and authorities intervene.

As shown in Fig. 7.6, most of the triggered alerts coming from IPs that are considered non-malicious by AbuseIPDB originated from Snort and the Web Application Firewall (WAF). This data could indicate that these IP addresses were in the early stages of an attack, with CIÊNCIAS ULisboa being among their initial targets. However, the significant discrepancy in alert frequencies may indicate that Snort and the WAF may require further tuning to minimize false positives and enhance the accuracy of threat detection.

From the analysis of Fig. 7.6 and Fig. 7.7, a key observation regarding the alerts triggered by WAF and Snort can be made.

WAF and Snort alerts are the most frequently triggered by IPs considered non-malicious, with Snort IDS showing 5,250 counts and the WAF showing 3,078 counts. Compared to the amount of alerts incoming from IPs considered malicious, (2,019 for Snort and 2,796 for WAF) this suggests that these systems detect a large number of potential threats, but only a subset of these are coming from known malicious sources.

This is just an example that demonstrates the type of insights we can derive from this data. By delving deeper into the analysis, we can uncover various other conclusions. More importantly, this data has inspired new future work aimed at enhancing existing security measures, which will be detailed in the conclusions.

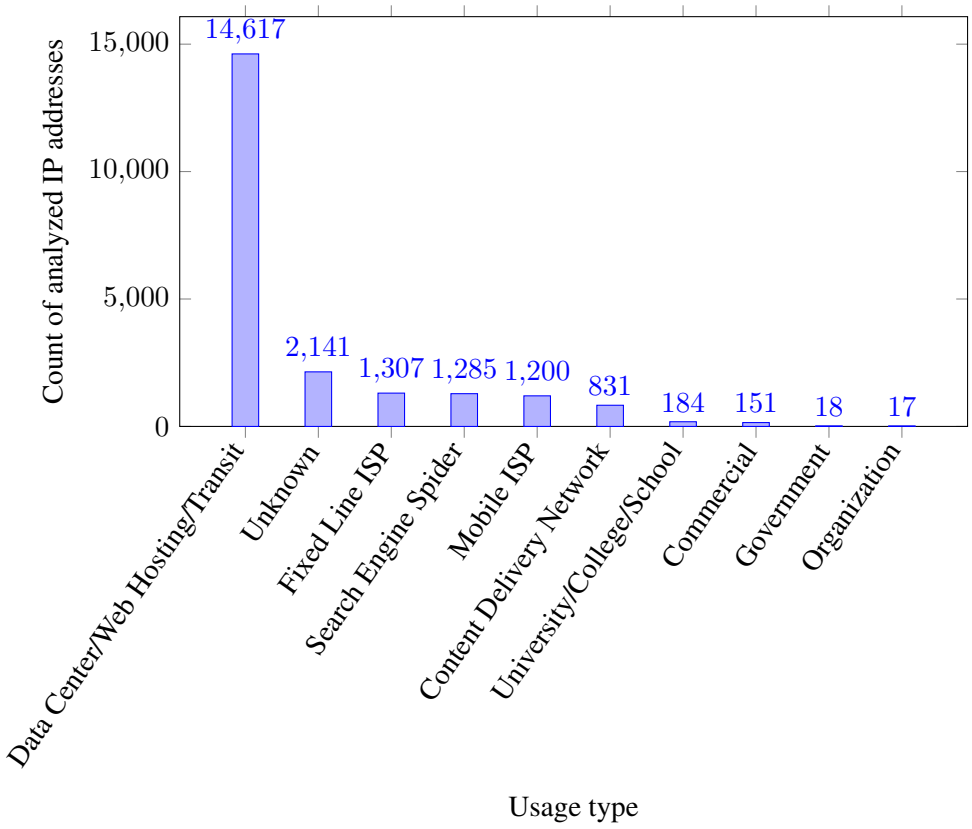

Figure 7.5: Top 10 usage types of analyzed IP addresses

7.3 Effectiveness at mitigating security threats

7.3.1 User Alerts

As stated in Sec. 6.1.7, emails began to be sent out to notify users about alerts related to their devices. The data gathered from this process proved useful in determining whether user notification was sufficient to deter future alerts. This analysis is based on a limited subset of distinct alert types sent to users. This subset was chosen by the following criteria:

Alert Frequency Alerts were chosen to strike a balance. Excluding those that would overwhelm the support team with excessive notifications from users, while ensuring that alerts with too few occurrences were also omitted to maintain relevance.

Verification Effort Alerts were selected based on the ease of verifying whether they were true or false positives, prioritizing those that could be confirmed without excessive difficulty.

Resolution Complexity Alerts for which solutions are straightforward and quick to implement.

Fig. 7.8 illustrates[1] the evolution of user notifications for a single type of alert, specifically the usage of torrents. The graph suggests that repeat offenders exist, indicated by their respective data points increasing over time.

[1]Only a small sample of 5 users is provided, as including more would reduce the readability of the graph.

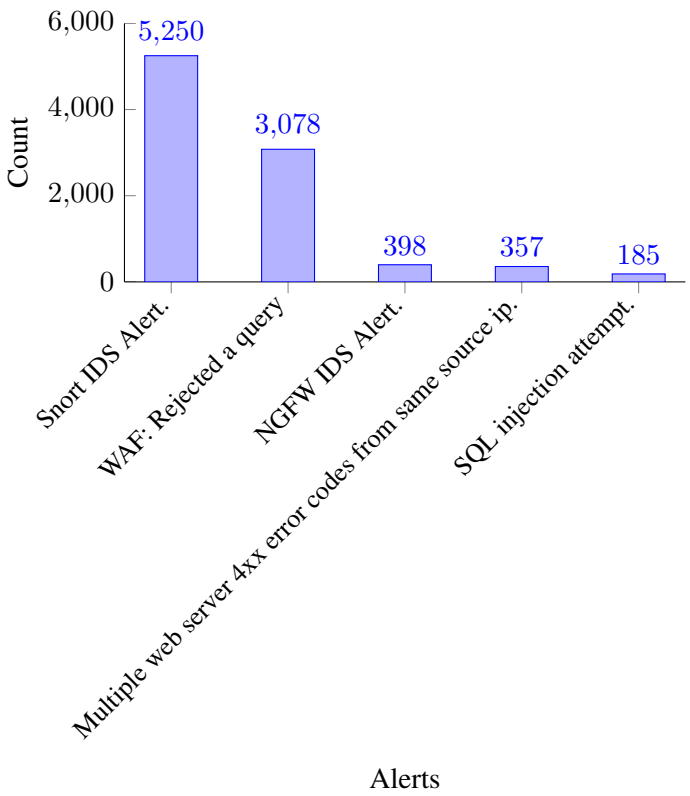

Figure 7.6: Top 5 alerts triggered by IPs considered malicious by AbuseIPDB with $\leqq 25\%$ confidence

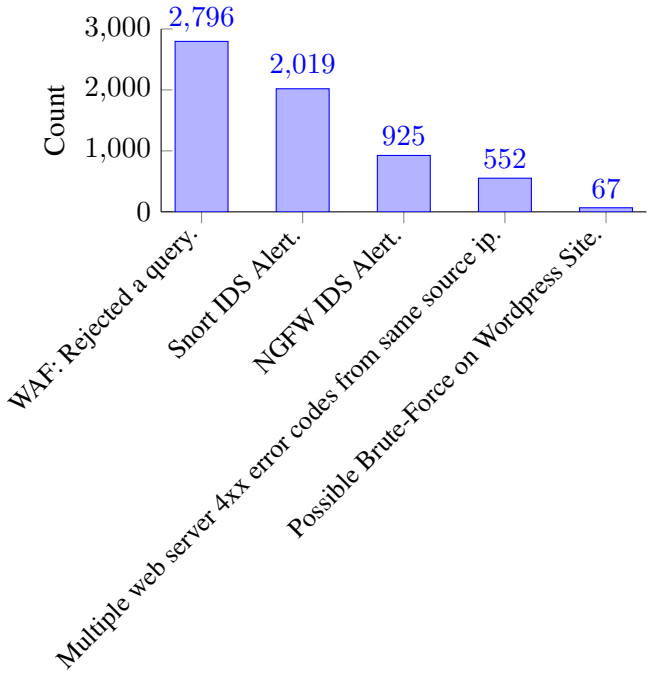

Figure 7.7: Top 5 alerts triggered by IPs considered malicious by AbuseIPDB with $\geqq 75\%$ confidence.

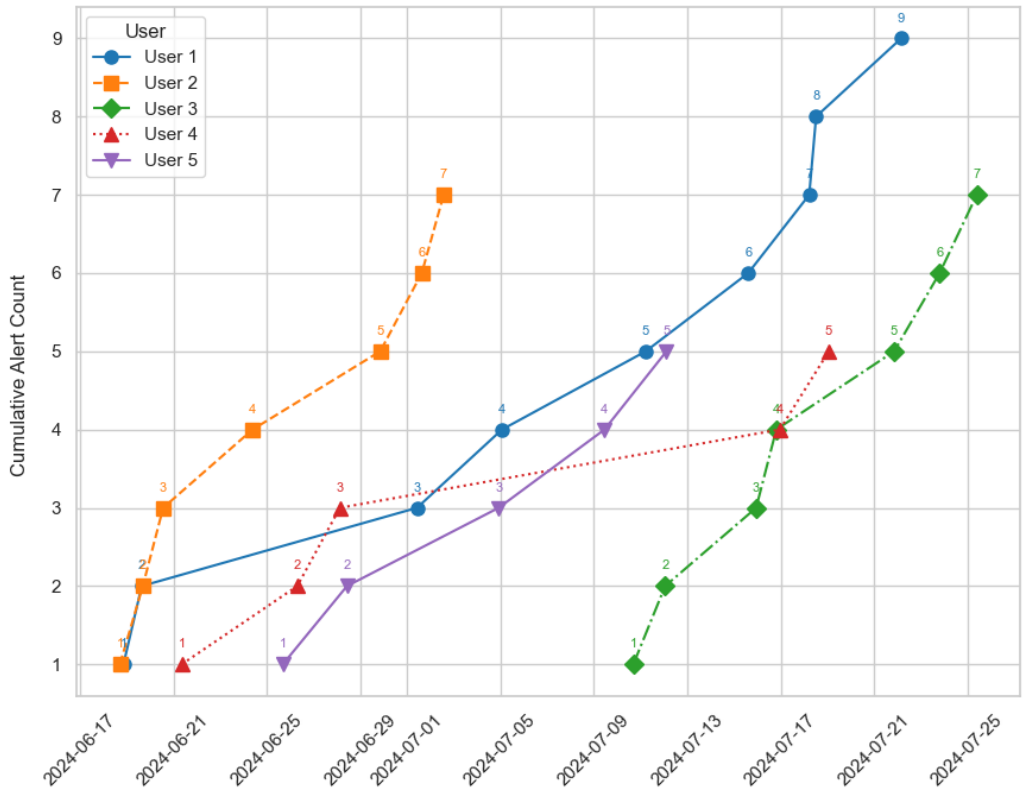

Figure 7.8: Cumulative evolution of user notification count over time for an alert related to torrents.

To conduct a more in-depth analysis, Fig. 7.9 illustrates a noticeable decline in the number of notifications sent over time. This reduction could be attributed to the conclusion of the school term, during which network usage by faculty members typically diminishes. There are also recurring dips in the notification count that align with weekends. This trend confirms that network traffic decreases significantly during weekends, resulting in fewer notifications being generated.

Further analysis suggests that user notifications may be effective, as demonstrated in Fig. 7.10. By examining the amount of users that received more than one notification (Repeat offenders) against the users that only received one (Non-repeat offenders), we can assess the impact of the notifications mechanism.

The findings suggest that CIÊNCIAS ULisboa should consider expanding its notification system to include additional types of alerts and look for better ways to deter repeat offenders. Continued monitoring will be essential to measure the effectiveness of new solutions and refinement of the existing one.

7.3.2 External Threats - Simulation

In Sec. 6.2.2 the blocklist generation script was explained. The data gathered from the analysis of the IPs that triggered alerts served to evaluate if the effects of blocking IPs are beneficial enough to deploy the solution. The observations reveal that many alerts were being triggered by IPs that were already on the blocklist.

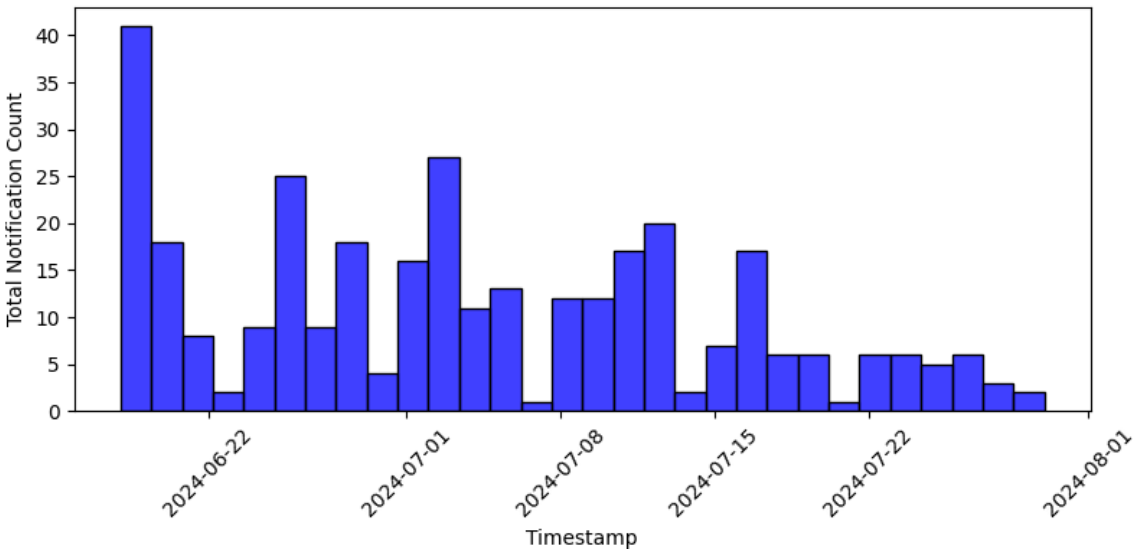

Figure 7.9: User Notification Count over time

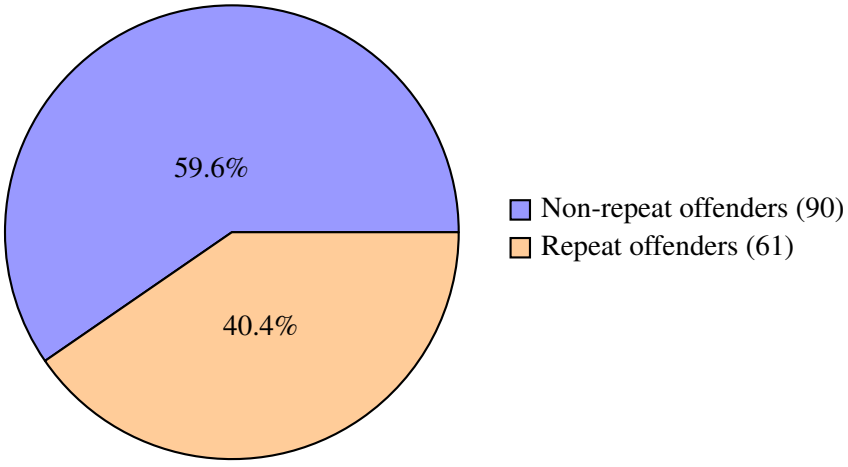

Figure 7.10: Distribution of repeat versus non-repeat offenders from 17th June to 31st July.

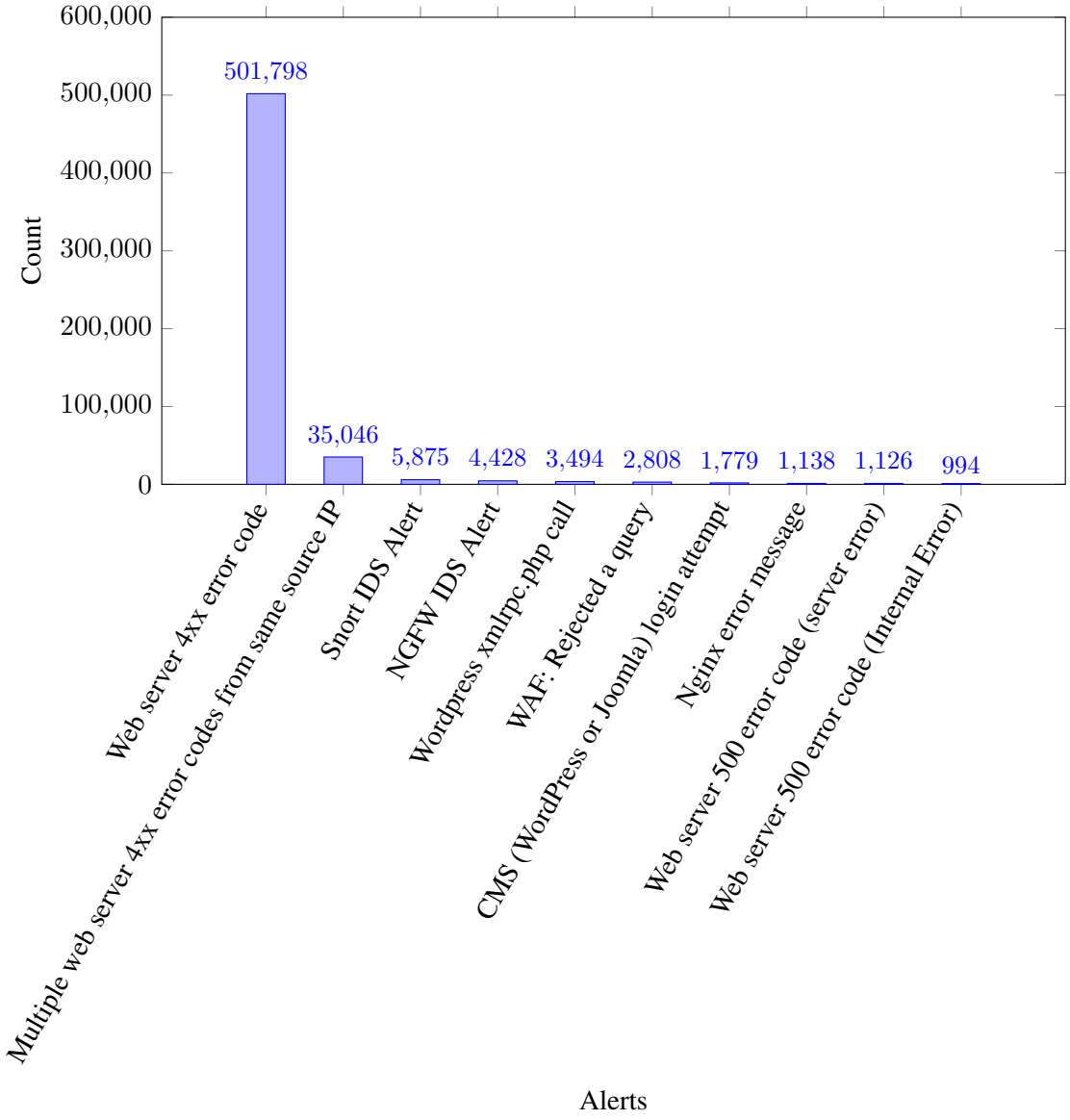

Figure 7.11: Top 10 triggered alerts from IPs in blocklist in the first half of August 2024.

The presented results demonstrate the types of alerts and quantity that would not have been triggered if the blocklists had already been deployed on the perimeter firewall. However, these results should be interpreted with caution, as the effectiveness of the blocklist can vary based on factors such as continued or discontinued attacks by specific IPs.

Figure 7.11 illustrates the distribution of alert types triggered by the IPs listed in the blocklist as of August 1st, 2024. It is evident that the most frequently triggered alert is the "Web Server 4xx error code". This high frequency is attributed to the substantial volume of web crawlers and scrapers, as detailed in Table 7.2, accessing the hosted web applications.

As shown in Fig. 7.12, over 59% of the total alerts generated in the first half of August 2024 originate from IPs on the blocklist. This excessive alerting creates additional noise, which can distract security teams from addressing genuine threats. These results reflect the impact of us-

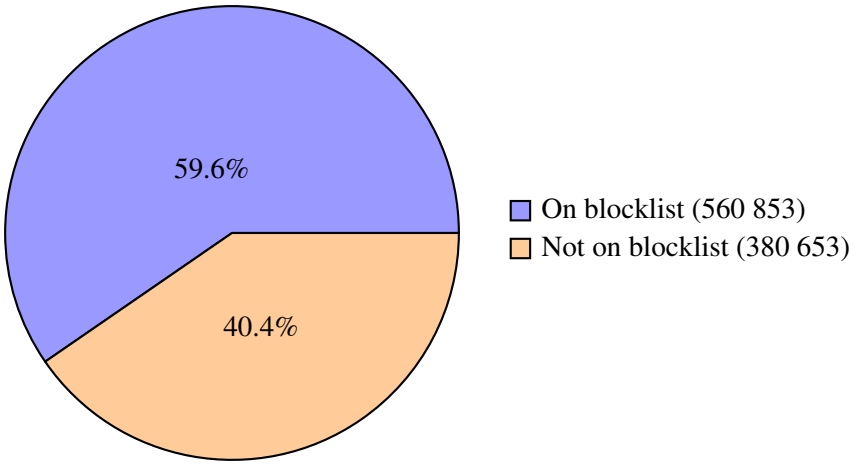

Figure 7.12: Distribution of triggered alerts between IPs on the blocklist and those not on the blocklist for the first half of August 2024.

ing a single source of cyber threat intelligence to assess IP reputation (AbuseIPDB); integrating additional sources could significantly increase this percentage.

Additionally, when averaging the alert count by day, it is found that approximately 37,390 alerts per day could be eliminated from the current daily average of 62,767 alerts incoming from external threats.

7.4 Adherence to security policies and standards

Although not a primary focus of this project, its implementation uncovered abnormal communications between internal hosts that deviated from standard practices and should be disabled. Moreover, it facilitated the discovery of an unused public service that was being targeted, which was promptly disabled.

The enhanced visibility provided by the project also allowed CIÊNCIAS ULisboa to integrate it into its internal security policy requirements, effectively establishing it as the security platform of CIÊNCIAS ULisboa for cyber threat detection.

Chapter 8

Conclusion

This chapter presents the conclusions drawn from this project, summarizing the key findings and evaluating the overall value of continuing to develop and enhance this project based on the results obtained.

This project initially focused on reconfiguring and enhancing existing security measures, such as Snort, along with redesigning the associated enrichment and correlation scripts. However, Snort's reconfiguration allowed for the observation of issues relevant to the cybersecurity of CIÊNCIAS ULisboa that led to the implementation of Wazuh. Its implementation and integration into the existing infrastructure improved the visibility of the IT infrastructure as discussed in the previous chapter.

Wazuh's versatility allowed for the integration of the redesigned scripts which extended the enrichment not just to Snort alerts but also for alerts triggered by the Next-Generation Firewall's IDS.

The enriched data from the redesigned scripts, enabled CIÊNCIAS ULisboa to identify irregular communications within the network that might have otherwise been overlooked, potentially exposing vulnerabilities to attackers. It also facilitated the identification of the actual attacker source IP when alert data was coming from between the reverse proxies and internal servers.

Although the volume of generated alerts is high, ongoing efforts with Wazuh will aid in reducing the volume and prioritizing the correct alerts. As demonstrated in the previous chapter, further improvements and optimizations will be achieved over time by continuously monitoring and analyzing data. While the immediate results may not be fully tangible, as discussed in the last chapter, cybersecurity is an ongoing effort, and the most significant benefits are likely to emerge over the long term. Expectations are that, with continuous improvement, Wazuh will significantly enhance CIÊNCIAS ULisboa's cybersecurity posture for years to come.

8.1 Future Work

The nature of cybersecurity is ever-evolving, with new threats emerging daily, making vigilance and adaptability crucial. This project has significant potential for future development, including enhancements that could impact alert volumes, attacker identification, event enrichment, monitor-

ing, and protection capabilities. A few additional measures and ideas emerged but could not be fully developed during the duration of the project:

Neural Network Integration for Anomaly Detection Implement a neural network to analyze gathered data to identify patterns that could signify false/true positives. By training models on gathered data, the system can improve its accuracy in distinguishing between legitimate threats and benign activities.

Automated Incident Response Develop automated response capabilities using playbooks or machine learning to make real-time decisions on mitigating threats. For example, locking an account in Active Directory upon detection of multiple alerts associated with it.

Natural Language Processing (NLP) Utilize NLP techniques to analyze and interpret unstructured log data more effectively, to extract relevant information, as having to create new regular expressions to extract information from custom sources requires manual effort, even though it is more precise.

For continued work, the following should be considered:

- Deploy the IP blocklists generated using Wazuh and AbuseIPDB on the perimeter firewall.
- Integrate additional threat intelligence such as VirusTotal;
- Tuning of detection rules.

Despite having many obstacles, the project ultimately succeeded. The main challenges stemmed from limited real-world experience with IT infrastructures and cyber threats, as well as a lack of comprehensive documentation from CIÊNCIAS ULisboa. However, many key objectives were met with the implementation of Wazuh. The current setup, with all its internal data sources such as Snort and threat intelligence tools like AbuseIPDB, establishes a strong foundation for continuous enhancements to CIÊNCIAS ULisboa's cybersecurity posture.

Bibliography

[1] AbuseIPDB. What is abuseipdb? https://www.abuseipdb.com/, 2024. [Online - Accessed on 02-07-2024].

[2] agoffe. Ioa vs ioc. https://www.crowdstrike.com/en-us/cybersecurity-101/threat-intelligence/ioa-vs-ioc/, 2024.

[3] Log360 Cloud. What is cloud based siem? https://www.manageengine.com/cloud-siem/what-is-cloud-siem.html, 2024. [Online - Accessed on 04-07-2024].

[4] João Paulo da Costa Calado. Open source ids/ips in a production environment: Comparing, assessing and implementing. Master's thesis, Faculdade de Ciências, Universidade de Lisboa, 2018.

[5] Faculdade de Ciências da Universidade de Lisboa. Faculdade de ciências da universidade de lisboa. alojamento web. https://ciencias.ulisboa.pt/pt/alojamento-1, 2020. [Online - Accessed on 17-06-2024].

[6] Faculdade de Ciências da Universidade de Lisboa. Faculdade de ciências da universidade de lisboa. network - addressing. https://ciencias.ulisboa.pt/en/network, 2020. [Online - Accessed on 09-06-2024].

[7] Faculdade de Ciências da Universidade de Lisboa. Faculdade de ciências da universidade de lisboa. network - historical overview. https://ciencias.ulisboa.pt/en/network, 2020. [Online - Accessed on 09-06-2024].

[8] Faculdade de Ciências da Universidade de Lisboa. Faculdade de ciências da universidade de lisboa. network - infrastructure. https://ciencias.ulisboa.pt/en/network, 2020. [Online - Accessed on 09-06-2024].

[9] Faculdade de Ciências da Universidade de Lisboa. Faculdade de ciências da universidade de lisboa. salvaguarda de dados. https://ciencias.ulisboa.pt/pt/salvaguarda-de-dados, 2020. [Online - Accessed on 17-06-2024].

[10] Faculdade de Ciências da Universidade de Lisboa. Faculdade de ciências da universidade de lisboa. suporte a infraestruturas informáticas - serviços

e condições no modelo sgc. https://ciencias.ulisboa.pt/pt/suporte-a-infraestruturas-informaticas-de-id-em-ciencias, 2020. [Online - Accessed on 17-06-2024].

[11] Arvindn et al. cron. https://en.wikipedia.org/wiki/Cron, 2024. [Online - Accessed on 08-07-2024].

[12] Google. Google is named a visionary in its first 2024 gartner® magic quadrant™ for siem. https://cloud.google.com/blog/products/identity-security/google-is-named-a-visionary-in-the-2024-gartner-magic-quadrant-for-siem, 2024. [Online - Accessed on 11-07-2024].

[13] Patricia Hoffman. https://nap.nationalacademies.org/read/18535/chapter/4, 2013. Department of Energy, February 27, 2013, Carnegie Mellon University (CMU) Software Engineering Institute CERT®.

[14] IBM. What is a cyberattack? https://www.ibm.com/topics/cyber-attack, 2024. [Online - Accessed on 05-06-2024].

[15] IBM. What is security information and event management (siem)? https://www.ibm.com/topics/siem, 2024. [Online - Accessed on 06-06-2024].

[16] Intellipaat. What is icmp (internet control message protocol). https://intellipaat.com/blog/what-is-icmp/, 2023. [Online - Accessed on 28-06-2024].

[17] Lenovo. What is a log in computing? https://www.lenovo.com/us/en/glossary/log/, 2024. [Online - Accessed on 06-06-2024].

[18] LogRhythm. Legacy vs. cloud-native siem: Weighing the pros and cons. On-PremSIEMvs.Cloud-NativeSIEM:WhichOneIsRightforYou?, 2024. [Online - Accessed on 04-07-2024].

[19] ManageEngine. Gartner® reconhece manageengine. https://www.manageengine.com/br/log-management/2022-gartner-siem-mq.html, 2022. [Online - Accessed on 11-07-2024].

[20] Lockheed Martin. Gaining the advantage. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf, 2015. [Online - Accessed on 05-06-2024].

[21] Lockheed Martin. Cyber kill chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, 2024. [Online - Accessed on 01-07-2024].

[22] Microsoft. What is extended detection and response (xdr)? https://www.microsoft.com/en-us/security/business/security-101/what-is-xdr, 2024. [Online - Accessed on 06-06-2024].

[23] MySQL. What is mysql? https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html/, 2023. [Online - Accessed on 09-12-2023].

[24] Palo Alto Networks. What is extended detection and response (xdr)? https://www.paloaltonetworks.com/cyberpedia/what-is-extended-detection-response-XDR, 2024. [Online - Accessed on 06-06-2024].

[25] Miguel Pupo Correia Paulo Esteves Veríssimo, Nuno Ferreira Neves. Intrusion-tolerant architectures: Concepts and design. in architecting dependable systems. https://repositorio.ulisboa.pt/bitstream/10451/14253/1/03-5.pdf, 2007.

[26] Python. What is python? executive summary. https://www.python.org/doc/essays/blurb/, 2024. [Online - Accessed on 02-07-2024].

[27] Rsyslog. Rsyslog - the rocket-fast system for log processing. https://www.rsyslog.com/, 2023. [Online - Accessed on 28-11-2023].

[28] Murtaza Ahmed Siddiqi. Critical analysis on advanced persistent threats. https://www.researchgate.net/publication/303325167_Critical_Analysis_on_Advanced_Persistent_Threats, 2023.

[29] Snort. Snort - what is snort? https://ciencias.ulisboa.pt/pt/suporte-a-infraestruturas-informaticas-de-id-em-ciencias, 2023. [Online - Accessed on 17-06-2024].

[30] Splunk. Pricing. https://www.splunk.com/en_us/products/pricing.html, 2024. [Online - Accessed on 04-07-2024].

[31] Splunk. Splunk enterprise security features. https://www.splunk.com/en_us/products/splunk-enterprise-security-features.html, 2024. [Online - Accessed on 04-07-2024].

[32] Splunk. What's cloud siem? security incident & event monitoring in the cloud. https://www.splunk.com/en_us/blog/learn/cloud-siem.html, 2024. [Online - Accessed on 04-07-2024].

[33] Sabina Szymoniak. Open source intelligence opportunities and challenges: a review. https://www.researchgate.net/publication/381074245_Open_Source_Intelligence_Opportunities_and_Challenges_a_Review, 2024.

[34] Richard French Viktoriya Degeler and Kevin Jones. Self-healing intrusion detection system concept. 2016.

[35] Wazuh. Architecture. https://documentation.wazuh.com/current/getting-started/architecture.html, 2024. [Online - Accessed on 18-07-2024].

[36] Wazuh. A comprehensive siem solution. https://wazuh.com/platform/siem/, 2024. [Online - Accessed on 06-06-2024].

[37] Heidi Willbanks. Legacy vs. cloud-native siem: Weighing the pros and cons. https://www.exabeam.com/blog/siem-trends/legacy-vs-cloud-native-siem-weighing-the-pros-and-cons/, 2023. [Online - Accessed on 04-07-2024].

[38] Lawrie Brown William Stallings. Computer security: Principles and practice. https://unidel.edu.ng/focelibrary/books/Computer%20Security%20_%20Principles%20-%20WILLIAM%20STALLINGS_2089.pdf, 2011.

Appendix A

Snort setup

A.1 Install Snort3 from the git repository following instructions on the web repository and guide.

```
$ git clone https://github.com/snort3/snort3.git
$ cd snort3
$ export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:/usr/local/lib64/pkgconfig:
    $PKG_CONFIG_PATH
$ export CFLAGS="-O3"
$ export CXXFLAGS="-O3 -fno-rtti"
$ ./configure_cmake.sh --prefix=<SNORT INSTALL DIRECTORY> --enable-tcmalloc
$ cd build/
$ make -j$(nproc)
$ make -j$(nproc) install
```

A.2 Install Snort3 Extras from the git repository.

```
$ git clone https://github.com/snort3/snort3_extra.git
$ cd snort3_extra
$ ./configure_cmake.sh --prefix=<SNORT EXTRA INSTALL DIRECTORY>
$ cd build/
$ make -j$(nproc)
$ make -j$(nproc) install
$ cd ../../
```

A.3 Download and install the Snort ruleset from the Snort website.

```
$ mkdir -p <SNORT INSTALL DIRECTORY>{builtin_rules,rules,so_rules,intel}
$ mkdir rules && cd rules
$ curl -Lo snortrules-snapshot-3000.tar.gz https://www.snort.org/rules/
    snortrules-snapshot-3000.tar.gz?oinkcode=<YOUR OINKCODE HERE>
$ tar xf snortrules-snapshot-3000.tar.gz
$ cp rules/*.rules <SNORT INSTALL DIRECTORY>/rules/
$ cp builtins/builtins.rules <SNORT INSTALL DIRECTORY>/builtin_rules/
$ cp etc/snort_defaults.lua etc/snort.lua <SNORT INSTALL DIRECTORY>/etc/snort/
$ cd ../
```

A.4 Download and install the IP blocklist from Talos Intelligence.

```
$ curl -Lo ip-blocklist https://www.talosintelligence.com/documents/ip-
    blacklist
$ mv ip-blocklist <SNORT INSTALL DIRECTORY>/intel/
```

A.5 Update Snort configuration to encompass specific needs.

```
# CHANGE FROM:

RULE_PATH = '../rules'
BUILTIN_RULE_PATH = '../builtin_rules'
PLUGIN_RULE_PATH = '../so_rules'
WHITE_LIST_PATH = '../lists'
BLACK_LIST_PATH = '../lists'
HOME_NET = 'any'
ips =
{
    variables = default_variables,
    rules = [[
        include $RULE_PATH/snort3-app-detect.rules
        include $RULE_PATH/snort3-browser-chrome.rules
        .....
        include $RULE_PATH/snort3-x11.rules
    ]]
}
reputation =
{
    --blacklist = 'blacklist file name with ip lists'
    --whitelist = 'whitelist file name with ip lists'
}

--alert_syslog = { }
--log_pcap = { }

# CHANGE TO:

RULE_PATH = '../../rules'
BUILTIN_RULE_PATH = '../../builtin_rules'
PLUGIN_RULE_PATH = '../../so_rules'
ALLOW_LIST_PATH = '../../intel'
BLOCK_LIST_PATH = '../../intel'
HOME_NET = '[[<NETWORKS YOU WANT TO PROTECT>]]'
ips =
{
    mode = tap,
    variables = default_variables,
    rules = [[
        include $RULE_PATH/snort3-app-detect.rules
        include $RULE_PATH/snort3-browser-chrome.rules
        .....
        include $RULE_PATH/snort3-x11.rules
    ]]
}
reputation =
{
    blacklist = '../../intel/ip-blocklist',
}
```

```
alert_syslog = { }
alerts = {
        alert_with_interface_name = true,
}
log_pcap = { }
```

A.6 Test Snort configuration by verifying if it is producing the expected output

```
$ <SNORT INSTALL DIRECTORY>/bin/snort -u snort -g snort -c <SNORT INSTALL
    DIRECTORY>/etc/snort/snort.lua --daq-dir <DAQ LIBRARY DIRECTORY> --plugin-
    path <SNORT INSTALL DIRECTORY>/extra/ --plugin-path=<SNORT INSTALL
    DIRECTORY>/so_rules -l <SNORT LOGGING DIRECTORY> -i <INTERFACE> -D -k none
```

A.7 User email

- - - For English version, read further below - - -

Caro(a) Nikhil Tulcidas,

Informa-se que esta mensagem eletrónica foi enviada automaticamente, uma vez que, foi gerado um alerta pelo sistema de deteção de intrusões da rede de dados de Ciências ULisboa, para um equipamento que se encontra sobre sua responsabilidade, de acordo com os nossos registos.

Detalhes do alerta:
Mensagem: Torrents - Transferência de conteúdo.
Responsável: nrtulcidas@fc.ul.pt
Data e Hora: Jun 17 12:43:08
Endereço IP: 10.101.12.232
Origem: VPN
Possivel resolução: Não transfira conteúdo através de torrents.

Referências técnicas externas de ajuda:
Para informações adicionais como versões de software vulneráveis ou outro tipo de informação, consulte:
https://www.snort.org/rule-docs/1-33215

Informações técnicas:
Endereço IP fonte: 10.101.12.232
Porto fonte: 63306
Endereço IP destino: 10.121.52.18
Porto destino: 53
Protocolo: UDP
Mensagem original: PUA-P2P BitTorrent transfer
Classificação: Misc activity
Prioridade: Low

Reforçamos que este email foi gerado automaticamente pelo sistema de deteção de intrusões da FCUL.
Mais se informa que, não receberá mais mensagens eletrónicas notificando esta situação, num espaço de 24 horas, relativamente ao mesmo tipo de alerta.
A Direção de Serviços Informáticos (DSI) encontra-se disponível para esclarecer e ajudar na resolução do problema.

Saudações académicas,
Direção de Serviços Informáticos
Campus da FCUL, Ed. C1, Piso 2, Sala 1.2.8

Contactos:
Telefone: 217500067; Extensão: 521248
Email: suporte@ciencias.ulisboa.pt
Direção de Serviços Informáticos: https://ciencias.ulisboa.pt/dsi
Ciências ULisboa: https://ciencias.ulisboa.pt

Figure A.1: Example email sent to a user (Portuguese section).

Dear Nikhil Tulcidas,

This electronic message was sent automatically, as an alert was generated by the intrusion detection system of the Ciências ULisboa data network, for an equipment that, according to our records, is under your responsibility.

Alert Details:
Message: Torrents - Downloading content.
Responsible: nrtulcidas@fc.ul.pt
Date and time: Jun 17 12:43:08
IP address: 10.101.12.232
Origin: VPN
Possible resolution: Do not download content via torrents.

External technical help references:
For additional information such as vulnerable software versions or other information, see:
https://www.snort.org/rule-docs/1-33215

Technical information:
Source IP address: 10.101.12.232
Source port: 63306
Destination IP address: 10.121.52.18
Destination port: 53
Protocol: UDP
Original message: PUA-P2P BitTorrent transfer
Classification: Misc activity
Priority: Low

We emphasize that this email was automatically generated by FCUL's intrusion detection system.
Please note that you will no longer receive electronic messages notifying you of this situation, within 24 hours, regarding the same type of alert.
The IT Services Department (DSI) is available to clarify and help resolve the problem.

Academic greetings,
IT Services Unit
FCUL Campus, Building C1, Floor 2, Room 1.2.8

Contacts:
Telephone: 217500067; Extension: 521248
Email: suporte@ciencias.ulisboa.pt
IT Services Unit: https://ciencias.ulisboa.pt/dsi
Ciências ULisboa: https://ciencias.ulisboa.pt

Figure A.2: Example email sent to a user (English section).