FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Architektura a technologie bezpečnostního dohledového centra (SOC)
Student:	Jakub Šimůnek
Supervisor:	Ing. Alexandru Moucha, Ph.D.
Study program:	Informatics
Branch / specialization:	Computer Security and Information technology
Department:	Department of Computer Systems
Validity:	until the end of summer semester 2023/2024

Instructions

A Security Operations Centre (SOC) is an essential part of securing corporate networks and dealing with cybersecurity incidents. The thesis will analyse the behaviour of security surveillance from the perspective of the operation of the center at the staff level, as well as from a technical perspective. It is also expected to deploy some of its technologies in a test or production environment and evaluate the behaviour of these systems.

Requirements:
1) Define the SOC activities. Describe of the roles and evaluation of the functioning of the security teams (staff part).
2) Analyse the technologies used for monitoring (SIEM, SOAR, Logging, Infrastructure monitoring).
3) Analyse the monitored technologies (IDS, IPS, Firewalls, Honeypots, Mail Filters, End Stations and Servers, Auditing Systems and issues of collecting logs from such devices).
4) Deploy, set up and evaluate the operation of selected security surveillance technologies.

Electronically approved by prof. Ing. Pavel Tvrdík, CSc. on 22 December 2022 in Prague.

Bachelor's thesis

ARCHITECTURE AND TECHNOLOGIES OF A SECURITY OPERATIONS CENTRE (SOC)

Jakub Šimůnek

Faculty of Information Technology
Katedra informační bezpečnosti
Supervisor: Ing. Alexandru Moucha, Ph.D.
January 4, 2024

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Šimůnek, Jakub. *Architecture and Technologies of a Security Operations Centre (SOC)*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Contents

List of Figures

List of Tables

List of code listings

I would first like to thank my supervisor, Ing. Alexandru Moucha, Ph.D., for providing guidance during the process of writing this thesis. His expertise was truly helpful to me. I would like to acknowledge my colleagues from work for bringing up and refining a topic this interesting and for encouraging me to work on it in my thesis. I would also like to thank my family that has supported me throughout the whole time of my studies and provided me with a comfortable environment for work.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In Prague on January 4, 2024 . . . .

Abstract

This bachelor thesis focuses on describing the principles of a security operations center (SOC), the technologies it uses, and the technologies it typically protects. Within the theoretical part, it analyses the reasons for the creation of such monitoring centers and their technological and personal structure. This bachelor thesis focuses on describing the principles of a security operations center (SOC), the technologies it uses, and the technologies it typically protects. Within the theoretical part, it analyses the reasons for the creation of such surveillance centers and their technological and personnel structure. In the practical part, the thesis focuses on the deployment of surveillance technologies in the form of their freely distributed variants, as well as introducing paid versions of these systems. Furthermore, the thesis contains a demonstration of the functioning of the previously built security infrastructure in the case of a simulated attack on the protected systems. The thesis succeeded in building a functional monitoring of systems in a laboratory network environment, based on techniques that can be applied in large-scale networks. The results of this work provide a clear procedure by which it is possible to build functional small-scale security monitoring.

Keywords Security operation center, defensive security, Security Information and Event Management, security logs, reactive security, cybersecurity incident response

Abstrakt

Tato bakalářská práce se zaměřuje na popis principů fungování bezpečnostního dohledového centra (SOC), technologií, které používá a technologie, které obvykle chrání. V rámci teoretické části rozebírá důvody vzniku takových dohledových center a jejich technologickou a personální strukturu. V praktické části se práce zaměřuje na nasazení dohledových technologií ve formě jejich volně šířených variant, ale i na představení placených verzí těchto systémů. Dále pak práce obsahuje ukázku fungování vybudované bezpečnostní infrastruktury v případě simulovaného útoku na chráněné systémy. V rámci práce se podařilo vybudovat funkční monitoring systémů v prostředí laboratorní sítě, a to na základě postupů, které lze aplikovat i v rozsáhlých sítích. Výsledky této práce tedy poskytují přehledný postup, kterým lze vybudovat funkční bezpečnostní dohled malého rozsahu.

Klíčová slova Bezpečnostní dohledové centrum, defenzivní bezpečnost, Security Information and Event Management, bezpečnostní logy, reaktivní bezpečnost, kyberbezpečnostní incident response

List of Abbreviations

CIA	Confidentiality, Integrity, and Availability
CISO	Chief Information Security Officer
EDR	Endpoint detection and response
IoC	Indicator of compromise
NIC	Network interface card
NOC	Network operations center
SIEM	Security information and events management
SLA	Service level agreement
SOAR	Security orchestration, automation and response
SOC	Security operations center
SWG	Secure web gateway

Introduction

In today's world great part of people's lives depend on the Internet, but the confidentiality, integrity, and availability of the services that constitute the Internet as we know it is always threatened. Various parts of the cybersecurity field pursue the highest protection and loss and breach resistance possible.

We can sort the field of cybersecurity into two major groups: reactive and proactive. Amongst the reactive types of jobs, we can find mainly incident response, forensic analysis, and reverse engineering. Amongst the proactive ones, we can find secure programming, threat hunting, vulnerability scans, patch management, and secure design (of networks or systems). The security operations center (SOC) teams' operations are mainly based in the reactive area (incident response), even though some proactive tasks are usually also carried out by the team (especially vulnerability scans and coordination of patches). For such a wide range of tasks, a common SOC team consists of different specialists with dissimilar skill sets and knowledge of various technologies.

The theoretical part of this work is centered around the description of roles inside a SOC team, duties and tasks of its members, and also the description of technologies deployed in the typical company network topography, that are either being monitored or are used for monitoring of activity in the network. The practical part of this work focuses on the creation of a laboratory environment with several different systems (router combined with firewall, endpoints, and DNS) and on the deployment of SOC technologies for monitoring. In the end, this laboratory environment will be subjected to a simulated attack, that will be mitigated using some of the procedures of a typical SOC.

In the first chapter, we will focus on the staff part of the SOC, where we can identify several levels, which all cooperate on incident handling and response. We will explain, how the tasks of a Level 1 analyst diametrically differ from the work of other levels, why they are labeled as the eyes and ears of the team, and also what are the typical issues of such a job. We will elaborate on the job of the Level 2 analyst and Level 3 analyst, why they need to have good communication skills, and what they use them for. Last but not least, the work of security engineers, managers, and other roles will be put in context with other roles in the team.

In the second chapter, we will focus on the network and infrastructure technologies, which are usually monitored by the SOC team. Therefore, this chapter will introduce network security devices and services such as firewalls, intrusion prevention or detection systems, honeypots, VPN systems, or mail agents and the security risks of such devices. These technologies also include the ones used by the employees or customers such as servers or endpoint stations.

The third chapter will focus on the security infrastructure of SOC operations. Therefore, this chapter will introduce logs, log formats, and specialized systems. These technologies are servers and programs for log management, correlations, and log searches, but also systems such as SOAR, which can provide automatic incident detection and response. Also, this chapter will describe how this infrastructure works and what needs to happen before any information can get to a human operator and can be properly dealt with.

In the practical part of this thesis, a small laboratory network environment will be designed and built. This network will consist of several types of devices, that would be part of the network topology in the case of a real company's network. These devices will later on become monitored by freeware solutions that will provide functions of the SIEM, log management, and correlations. After this network will be created and connected by this security infrastructure, this topology will be subject to a simulated attack, that will demonstrate how this type of incident would be handled in a real-life scenario. Therefore, the event will be logged, this log will be sent to the correlation engine and a human operator will decide how to mitigate this attack.

The goals of this thesis are to present the structure and internal functioning of a typical SOC team of both the technical and personal parts but also to provide a procedure that leads to the successful creation of a SOC infrastructure and the deployment of specialized security technologies that are used for the protection of organization's systems. The last goal of this thesis is to demonstrate a simplified process of Incident response with the tools that will be deployed in the practical part of this work.

The results of this work could be beneficial to relevant specialists who are looking for information about the architecture and functioning of a typical Security operations center. I have chosen this topic because it isn't well covered in an organized manner in any of the publicly available sources.

Chapter 1

SOC Activities and roles

1.1 Why do we need SOC teams?

When any company or organization provides services to the public or its employees and needs to ensure proper functions of such systems, they need a team of specialists, that monitor the systems for any anomalies. Therefore, we build or hire the services of such a team. First of all, we need to ensure, that all systems work properly (they are connected and responding correctly and without inappropriate delay), so we employ a NOC (Network Operations Center) team. However, the NOC team might lack the proper cybersecurity knowledge. Therefore, at this point, we should employ cybersecurity specialists to monitor the systems for cybersecurity anomalies and so we create the SOC team. Therefore, if we want to provide around-the-clock functionality and security, we create both NOC and SOC teams to ensure the proper operation of all the systems.

1.2 What are the primary objectives of the SOC team?

The SOC team is set to monitor the organization's systems they are entrusted with in order to protect them from any malicious or unwanted activity. Moreover, reactions to some types of unwanted activities can be regulated by contracts (even the longest acceptable time of reaction needs to be monitored because some limiting time window can be set). All this is usually specified in the SLA (Service Level Agreement), which needs to be followed or the SOC team risks fines. These activities can be divided into two groups: Security events and Security incidents. A security event is an activity or a state of a system, that threatens the confidence, integrity, or availability of data or the system, but no confirmed violation was found. Security events happen a lot and need to be properly investigated. However, if any of the above-mentioned CIA (Confidentiality, Integrity, and Availability) triad properties were confirmed as compromised, the security event needs to be immediately escalated to a security incident level. This naming of different types of security problems is important for a proper reaction to such activities, and especially for legal compliance (usually only the security incidents need to be announced to public authorities, however, such procedures differ across states).

1.3 Activities

As mentioned in the introduction, there are various tasks and activities carried out by the SOC team. We usually divide them into 3 separate categories[1]:

- Preparation, planning, and prevention
- Monitoring, detection, and response
- Recovery, refinement, and compliance

1.3.1 Preparation, planning, and prevention

The most important thing a SOC team must have is deep knowledge about all the deployed systems in the protected networks, their versions, and also configurations. Without deeper knowledge, many of the incidents across the network couldn't be solved in a short enough time to prevent further losses on the infrastructure. Also, the SOC team must be fully aware of the security technologies that are used for the monitoring. Overall, the deeper the knowledge, the better the reaction to any serious incident.

The next important part of the activities of a common SOC team must be the maintenance of the security technologies used for log management, detection, correlations, and response. Sometimes the SOC team can be tasked with patching and updating certain parts of the network and systems, but this can vary according to the division of such tasks between the IT management teams.

One of the most important parts of the SOC team's job is preparing plans for the event of a security incident happening somewhere in the network. Such plans need to be prepared well because every second counts in such cases of emergency. Every team member must be assigned specific roles and responsibilities in the case. Such responsibilities need to be distributed across all the members of the team because incident response needs to be quick and efficient. For example in the event of a DDoS attack, a reaction must be made in a couple of minutes to prevent the denial of service (if possible).

Another activity that might be provided by the SOC team is vulnerability management and testing. Therefore, the SOC team provides security information to the relevant (for example IT administration) staff, but can also test if the mitigation technique or system patch has successfully avoided any future abuse of the vulnerability. Consequently, the SOC team should provide continuous testing of a large spectrum of vulnerabilities that could be present in the monitored systems. This testing can be done through tools that perform such checks more or less automatically, or the SOC team itself can have a penetration tester that continuously tries to attack the network and the systems.

1.3.2 Monitoring, incident detection, and response

The most important part and the sole purpose of the SOC team is monitoring and incident handling. For this, the ideal SOC team delivers 24/7/365 monitoring of all the systems in the network. All of the SOC teams need to use specialized services like SIEM (Security Information and Event Management) and others (services that don't necessarily do correlations, which is the main purpose of the SIEM, but rather present the events in the form of dashboards) for such tasks.

”For many SOCs, the core monitoring, detection, and response technology has been security information and event management, or SIEM. SIEM monitors and aggregates alerts and telemetry from software and hardware on the network in real-time, and then analyzes the data to identify potential threats. More recently, some SOCs have also adopted extended detection and response (XDR) technology, which provides more detailed telemetry and monitoring, and the ability to automate incident detection and response.”[1]

1.3.3 Recovery, refinement, and compliance

This category mostly differs across the various SOC teams. In smaller organizations, the recovery might be done by the SOC team itself because they might have direct access to the monitored infrastructure. However, in the case of larger organizations, direct access usually is not possible (there would be many such systems and the administration itself would pose a huge challenge to the team). Therefore, in the case of larger systems, the recovery procedure is done by other teams of either the customer (in the case the SOC is provided as a service to a customer) or the organization. Such recovery can consist of restarting systems, switching to backup systems, restoring accounts with stolen or cracked passwords, etc. [1]

There might be further steps for upgrading security, like reflecting the last attack into the standardized procedures and incident detection, staff training, and the usage of security tools.[1] Consequently, new malware samples and indicators of compromise can be obtained through forensic analysis of the previously attacked device.

However, the last part of the SOC team's job after a security breach is the incident reporting to national authorities (for example in case of GDPR violation or other damages) and the legal compliance in the IT security area.

1.3.4 Activities conclusion

In conclusion, the job descriptions and activities of any SOC team can greatly vary depending on the legal environment of the country of operations, the type of service provided (in-house SOC vs SOC as a service), and the internal policies of an organization that operates the SOC team. However, the most common ones are:

- Prevention and preparation - incident response plans, maintenance, vulnerability management
- Monitoring, incident detection - incident detection and mitigation
- Recovery, compliance - law compliance, recovery procedures and cooperation with other teams, reporting to authorities

1.4 Roles in the SOC team

Traditional SOC team is divided into several levels, whose personnel greatly differ in the depth of skills and sometimes also technologies that are provided for the solution of an incident. [2]

1.4.1 Level 1 Analyst

Level 1 (L1) analyst or a so-called Incident responder is responsible for immediate reaction to a security incident or a suspicious event. He must be the first one to know when something has gone wrong because he is tasked with monitoring the technologies and with the triage of events that have happened across the network and systems. If such an event is recognized as a serious security incident, the Level 1 Analyst can escalate it to the L2 Analyst who is expected to have better skills to determine the reason for such event and to make a proper response. Such security incidents or events can be discovered through the information provided by the monitoring systems or in the worst case by someone's call. Therefore, in summary, an L1 Analyst is tasked with triaging the events, monitoring systems, and answering calls as a security helpdesk, then doing as much investigation as it is possible within his level of knowledge and skills and then escalating given problems with discovered information to the L2 Analyst or to consult the mitigation approach with him.

1.4.2 Level 2 Analyst

A level 2 (L2) Analyst or a Security investigator is tasked with a deep and thorough investigation of the incident that was handed over by the L1 Analyst. His task is to determine the level of risk and possible damages to the systems, contact the chief information security officer, and alert the administrators of the affected system to mitigate ongoing or past attacks.

1.4.3 Level 3 Analyst

Level 3 (L3) analyst or a cyber security expert should be the most skilled person involved in the Incident response and prevention. In many smaller SOC teams, this position might be merged with the job of an L2 Analyst. But, if such a worker exists, his job is mainly to advise the people on the lower levels, if they are past their level of expertise. Another important role of this position is the prevention of incidents from happening and the preparation for the case the incidents happen. Therefore he must do threat hunting across the network to find all the weaknesses and also the preparation of all the systems and procedures for cases when someone would abuse such vulnerability. This role can also advise the organization to change deployed software and hardware to improve the overall security of the organization.

1.4.4 Security engineer

Security engineer could be described as the backbone of the whole team because his main job is to constantly evaluate the systems and monitoring tools so that they are ready for as many types of incidents as possible. He might be sometimes tasked with the administration of such systems, but this can greatly vary from one team to another. Sometimes his role can also include the tasks of security compliance and security architecture.

1.4.5 SOC manager

The SOC manager should be the one, who keeps in touch with the Chief information security officer (CISO) and other teams (software development, network administration, infrastructure administration) and makes the critical decisions that might arise from the daily SOC operations.

1.4.6 Other possible roles

There are other possible roles of the SOC team that can be provided for the organization or the customers. Such roles could be penetration testing or forensic analysis. Such roles can be combined with various previously mentioned positions, can be outsourced to other specialists inside or outside the organization, or can be separated into special positions belonging to the SOC team. However, these services should be placed inside the SOC team, because the information extracted from systems during penetration tests or forensic analysis could be invaluable for further upgrade of the security measures in the company.

1.4.7 SOC team roles summary

To summarize, the roles inside the SOC team usually include:

- L1 up to L3 Analysts
- Security engineer
- Manager

- Forensic analysis specialist
- Penetration tester

1.5 Evaluation of the function of a SOC team

It is difficult to analyze the SOC team structure globally because every SOC team's structure is different. That is because the structure is deeply dependent on the needs of diverse customers, the capacities of the team, or the overall needs of the parent organization. However, we can evaluate the most common structure, roles, and problems, that do exist in many SOC teams around the world.

According to a study performed in 2022, the majority of the SOC team members globally identify the following issues in their work: "... Too much information, more work than they can handle, difficulty finding and keeping SOC experts, insufficient downtime, too many tools (and lack of tool integration), and too many alerts... " [3] The common denominator of such issues is the lack of resources and staff and this leads many of the specialists to consider changing position. Moreover, the cyber security field is constantly changing and evolving (there were 8 051 vulnerabilities listed just in the first quarter of 2022 according to the NVD database [4]). Therefore the personnel, but also the tools need to evolve with it. However, for many SOC teams around the world, this is not the case.

Another thing that can increase the workplace problem is, that the majority of the work done by the SOC team, is (or at least mostly should be) reactive. Therefore, sometimes you can feel a lot overstaffed, but a moment later you are missing quite a few people to help you with everything that you need to do to avoid further problems. This puts a huge stress on the members, especially when internal or customer SLA times are set (this is sadly usually the case).

1.5.1 L1 analysis

Every team needs an entry-level role, where the inexperienced team members have the opportunity to learn the procedures and knowledge needed for everyday operations or for the position on higher levels. L1 analyst's job doesn't necessarily require deep cyber security knowledge.

However, this low-level position can pose a huge threat to the stability of the L1 team. The quality of the L1 team overall may suffer in such cases. The better L1 team members are often promoted to other levels and the average or below average remain on L1. Therefore, the quality of an average L1 Analyst can deteriorate quickly in favor of higher positions. To have a well-functioning team, you need both stabilized higher levels and a continuous flow of newcomers to either expand the team or to replace the bad or leaving employees. Moreover, the job of an analyst can be even more exhausting in the case of the work in shifts that might include working overnight. Therefore, this is not an easy job, particularly under today's circumstances where the industry is short 3.4 million cyber security workers [as of October 2022][5]. This position's instability poses a huge threat to a functioning SOC team.

1.5.2 L2 and L3 analysis

The L2 analyst is a senior position in the team that requires a higher level of expertise (compared to the L1 analyst) and skills from the employee. L2 is supposed to work much more independently (this is the key difference between L1 and L2 analysts). They must be truly proficient in the area of investigating skills (they are usually the ones who investigate all the details of an incident), Also, they must be able to effectively communicate (they are way more likely to communicate with other teams, administrators, managers or the customer). However, this communication tends to be very time-consuming, ineffective, challenging, and exhausting. The quality of the L2 analyst is much more crucial to the success of the SOC team than the L1 Analyst. Therefore, they must possess a range of skills to effectively carry out their responsibilities and must be trained regularly to keep up with the development of the cybersecurity field.

1.5.3 Security engineer

This job can be exhausting as well, especially when the parent organization of the SOC team does not prioritize cybersecurity. In such cases, the engineer may be required to maintain obsolete technologies, that might not be designed to withstand today's workloads. As the systems age, it might become increasingly difficult for the engineer to maintain them correctly, which can later result in problems for other team roles within the SOC. Additionally, it is the job of the Security engineer to regulate the number of alerts coming to the L1 Analyst, because otherwise they could get overloaded with work and this could lead to a security incident going unnoticed by the team. However, regulation of the number of alerts is a challenging task, especially if the SOC team has weak vulnerability management and asset management. Every old and new alert rule needs to be regularly and carefully considered because every functional alert rule in the system can clog the system and personnel with false positives.

1.5.4 Conclusion of evaluation

The SOC team's job can be exhausting, and the work hours required might be long (especially in the case of night work). Moreover, the job can be complicated by the organization itself when it doesn't prioritize cyber security. Additionally, some of the positions require deep knowledge of the systems and IT and are therefore harder to fill. In conclusion, thanks to the added global shortage of cyber security workers, it might be hard to establish or keep a functioning, professional, and skilled SOC team.

Chapter 2
Monitored technologies

The systems of modern organizations are very complex and contain many distinct types of services, therefore SOC teams are put against the vast amount of varying types of monitored technologies. Every device produces its own type of activity that needs to be properly logged and analyzed to guarantee the security of the device and the network around it. It greatly depends on the organization's needs, which specific technologies and systems are deployed, which are logged, and where these logs go (this activity might not be centralized). This chapter however is focused on a brief general introduction of these technologies, which might be and usually are monitored by the SOC teams, rather than on the implementation side specifics of each solution.

2.1 Endpoint detection and response

We can suppose that there is no organization globally that wouldn't have to deal with end-user or endpoint security. It is not difficult for malicious actors to intrude into the system through the end user, because he is usually the one with the least abilities.

Even though intrusion through emails, malicious files, phishing, malicious links or portable media should be low (usually the employees of the company go through regular training about such basics of security), in reality, it is not. Especially in large organizations the possibility of such an incident is not insignificant and therefore we need another layer of protection to avoid further damage. For this type of protection, we use special pieces of software called Endpoint detection and response (EDR - not to be confused with Extended detection and response (XDR), which has a slightly different approach to dealing with system and network security on the organization's level).

EDR extends the possibilities of a normal antivirus (which might or might not be completely replaced by EDR) to provide complex security solutions for the user's endpoint. "*The Endpoint Detection and Response Solutions (EDR) market is defined as solutions that record and store endpoint-system-level behaviors, use various data analytics techniques to detect suspicious system behavior, provide contextual information, block malicious activity, and provide remediation suggestions to restore affected systems.*"[6] If the incident isn't prevented, the EDR provides logs that can help the SOC or the endpoint administrator investigate the activity for better prevention of such incidents in the future or on other machines. The logs that are collected by the EDR might include the history of files, web browser history, IP connections of the system, authorization activities, process execution, or external drive usage).

However, with more intensive log collection on a larger network topology, every SOC needs to consider investments on their part (such detailed logs might require a large volume of available storage).[7]

2.2 Mail technologies

Mail technologies are another critical part of the IT infrastructure of any organization. Organizations have two simple solutions for such infrastructure - on-premise or cloud. If any organization chooses the cloud version of such services, it is also outsourcing its security (that's all done by the cloud provider). However, if an organization chooses an on-premise solution, it needs to protect its systems. According to Check Point Research: "*81 % Of malicious files were distributed by email, 1 in 239 email attachments are malicious, and 1 in 415 Links in emails are malicious.*"[8]. Therefore, if we want to efficiently protect the organization's network, we need to constantly monitor emails flowing both ways.

For such protection, specialized services such as Cisco Email Security Appliance or Barracuda Email Security Gateway are developed and deployed in the networks. With this kind of tool, we can check emails for suspicious or malicious links (based on domain blacklists) and we can examine any email attachments with antivirus solutions. Also, these solutions check the legitimacy of a given email sender, so they can block forged email messages. Unfortunately, such solutions aren't flawless and therefore we need another layer of protection. This is where the SOC team with its tools might step in.

However, email security is one of the areas, where the security is not perfect because all the measures are just partial and the rest of the protection remains on the user. For example, these technologies are unlikely to protect a user from spear-phishing campaigns on a small scale, because if the email is sent from a legitimate address to a minority of users, such Email security devices have a low chance of detection, and SOC team might not have such detailed visibility for such small scale attack. However, if a user is enough watchful, this email can be reported to the SOC team which can take measures in order to mitigate this attack attempt.

2.3 VPN concentrators

With the rise of Home office work, companies needed to provide their employees with remote access to internal systems. In order to do this securely, the VPN technologies were deployed. However, this type of remote access to otherwise inaccessible inner resources of the organization creates new vectors for attack. Therefore, access through VPN concentrators to the inner network needs to be constantly monitored to detect any anomalies in the access of users (brute-force attacks, remote access from other parts of the world, especially to administrator accounts).

However, such monitoring greatly depends on the set procedures of the organization (such suspicious access to the resources needs to be checked with the VPN administrator or with the superior of the employee). Therefore, VPN monitoring is one of the crucial jobs that is almost exclusively done by the SOC team.

2.4 Network Security Devices

Another crucial part of securing an organization's network is the securing of the network devices themselves. There are many such devices and their combinations, so this chapter is focused on an explanation of basics of the network security devices.

2.4.1 Intrusion prevention systems and Intrusion detection systems

The intrusion prevention systems (IPS) or intrusion detection systems (IDS), which exist in both hardware and software versions, are a type of devices or programs that are tasked with continuous monitoring of the network traffic. Such network traffic can be almost any traffic that carries data or metadata, that are being processed somewhere (therefore not only HTTP, but also many other standards of internet communication). The main idea of using solutions for intrusion detection or prevention is to detect and/or stop malicious traffic or data leaks from the internal network and systems.

The basic difference between the IPS and IDS is the severity of the action that is taken after any malicious or uncommon activity is detected. IDS is set to just report any detected malicious activity to a human operator (this is where the SOC team operations might come into place), or it can actively mitigate the malicious activity (in the case of an IPS).

2.4.1.1 Location-based classification of IPS/IDS

While there is not only one way to the classification of the IDS and IPS systems, we can divide them based on their position in relation to protected devices:

- Network-based systems
- Host-based systems

The main type is the Network-based prevention or detection systems (usually called NIPS or NIDS), which reside in the network topology at strategic points and provide monitoring of network traffic of the devices that are connected to such monitored network. [9] A network-based ”*...intrusion prevention system is placed inline, in the flow of network traffic between the source and destination, and usually sits just behind the firewall.*”[9]

There are also Host-based prevention or detection systems (usually called HIPS or HIDS), that reside on the device they are tasked to protect.[9] However, the HIPS or HIDS device ”*...works best in combination with a NIPS, as it serves as a last line of defense for threats that have made it past the NIPS.*”[9]

2.4.1.2 Threat identification approach-based classification of IPS/IDS

However, we can subdivide the IPS and IDS devices based on the threat identification approach [9]:

- Signature-based threat identification
- Anomaly-based threat identification
- Policy-based threat identification

The signature-based platform tries to match traffic with signatures that have been acquired after previous attacks. Therefore, it has a low chance of intercepting an attack through a new attack vector or one with changed parameters.

The second type is anomaly-based threat identification, which analyses the current traffic against a standard baseline. Therefore, it should be able to intercept even new yet unknown types of attacks, but only in case that they produce a high enough volume of traffic that differs from a standard baseline. For this type of IPS/IDS, the deployment of machine learning is sometimes used for the identification of anomalies.[10] However, this strategy is more prone to false positives. [9] The third type is policy-based, therefore it uses policies defined by the administrators for the decisions about traffic. [9]

However, the structure of solutions can be mixed to include the best of both worlds. Therefore it is usually a combination of Signature-based detection with Anomaly-based detection, which can however create a significant increase in the number of detections.

These detections however usually need a human operator (most likely a SOC team member), who will decide on the next procedure and contact, for example, the administrator, who is responsible for the targeted system.

2.4.2 Firewalls and Next generation firewalls

Firewalls are another essential devices that are doing their best to provide security for the inner network of an organization. It is used for filtering out traffic that is defined by the administrator‘s policies as malicious and for letting legitimate traffic enter the network or exit it.

2.4.2.1 Stateless and stateful firewall

This group is divided into two types, that do differ in the approach they take to analyze key indicators of the traffic that is coming in or leaving the network. The first and today obsolete type is a stateless firewall, that checks traffic on a packet-by-packet basis. The check of any packet focuses on the parameters like source, destination, and several others.[11] This approach however has proven to be less efficient in eliminating threats, because some types of attacks are based on legitimately looking packets, that are without context let through by a stateless firewall. Examples of such types of attacks are TCP Scans. "*Some scans will send a TCP packet out of sequence and observe the response. Examples include ACK and FIN scans.*"[12] Another type of attack, that might not be eliminated by a stateless firewall is a DDoS, where the context contains the number of packets sent by the source.[12]. Therefore, for further network protection, we need a more modern approach. For this purpose, a stateful firewall type was designed. This type tracks the communication throughout its entire duration and checks the legitimacy of a given packet in the context of other packets that were sent before and the ones that are expected to be sent in this type of communication. "*Now thought of as a “traditional” firewall, a stateful inspection firewall allows or blocks traffic based on state, port, and protocol. It monitors all activity from the opening of a connection until it is closed. Filtering decisions are made based on both administrator-defined rules as well as context, which refers to using information from previous connections and packets belonging to the same connection.*"[13]

2.4.2.2 Packet-filter firewall

However, we can also divide the firewalls by the layer of the OSI network model they operate on. We can have the packet-filter type of firewall, that operates on layer 3 (network layer). Therefore, this type of firewall device decides the actions based on the criteria that take into account mainly the following characteristics of the traffic: destination IP address, source IP addresses, packet type, port number, and the network protocols the packet is part of. Most importantly, the packet filter does not examine the payload of the packet. Moreover, this type of firewall processes the packets on an individual basis (it does not take into account the context of the communication). Therefore, the security measures imposed by this firewall type can be bypassed by techniques like IP spoofing.[14]

2.4.2.3 Circuit-level gateway

Another type of firewall, that is based on another layer of the network OSI model is a Circuit-Level Gateway. This one operates on layer 5 of the OSI model (the session layer). However, it is not a standalone firewall solution, because of its focus on only one type of communication. This type of device monitors solely that the TCP handshake process between local and remote hosts is correct. However, even this type of firewall does not inspect the payload of any of the passing packets and therefore does not protect the systems from malware data being sent.[14]

2.4.2.4 Application layer gateway

Another very important type of firewall is a proxy firewall, or a so-called Web Application firewall or an Application layer gateway, is a type of firewall, that can protect the servers inside the company against malicious payloads sent to them. It operates on layer 7 of the OSI model, therefore the application layer. They provide deep packet inspection (they consider not only the packet header information but also the packet payload and the context of the whole communication with the remote host) and thus they can provide much better protection than the above-mentioned types. Moreover, they provide additional security to the protected system, because it hides the real IP address of such system. Thanks to the complexity of the checks, additional rules for the protection can be set.[14] One such example is a geoblocking (for example in the case of a severe Denial of service attack that comes from a small number of countries).

2.4.2.5 Secure web gateway

SWG (Secure web gateway) is a type of device that needs to be added to the network in order to protect the systems from the known or unknown unsafe activity of users inside the network (usually the employees). Therefore, this device provides several services that are focused on protecting users from different online threats. It for example prevents access to known malicious websites (either phishing ones, illegal ones, or the ones distributing malicious code) or unwanted application servers. Moreover, it adds antivirus capabilities, that are used to analyze incoming web traffic for any known threats.[15]

Therefore, the protection provided by SWG is comprised of several features [15]:

- URL Filtering
- Application Control
- Data Loss Prevention
- Antivirus
- HTTPS Inspection

URL Filtering is deployed because of the need to monitor websites, that are accessed by the users of the inside network (because those could be phishing sites [for example fake sites with email login], sites distributing malware, or mining Bitcoin for example). This traffic can be either discarded immediately or sent for further analysis to the SOC team analysts.

The second part of protection is Application control. This enables monitoring of unwanted application traffic or complete blocking of it. Examples of unwanted applications might be Remote desktop applications, that might be used by an attacker to compromise the network or the data of the user.

The third part is data loss prevention. "*Data Loss Prevention (DLP) ensures that critical and sensitive information is not sent outside of an organization's network. Data loss prevention pre-emptively protects your business from unintentional loss of valuable and sensitive information by monitoring data movement and adhering to industry compliance regulations and standards.*"[15]

Antivirus solution provides protection against malware, that could be part of the network traffic. This traffic is therefore constantly monitored and compared with virus signatures in order to prevent malware delivery to the endpoint.

The last important part is the HTTPS inspection, when the encrypted traffic is decrypted, checked with standard HTTP protection procedures and the re-encrypted and sent to the destination.

2.4.2.6 Next-generation firewall

A next-generation firewall is a relatively new term, that describes an all-in-one device that combines the abilities of a firewall with IPS/IDS services and sometimes services of secure web gateway and other security measures like Application awareness and control or Threat intelligence [16]. Therefore, it provides deep packet inspection, TCP handshake checks, VPN services, malware scans, and NAT or SSH inspection. [14] Examples of such devices are Cisco NGFW and ASA.

As described, firewalls are very independent devices, that protect the network from incidents. However, in the case of DDoS, human operator activity is needed for mitigation. In other cases, especially after any network-based incident, we need the logs to check the activity of the attackers before and during the attack. This is why the Firewalls are another, but very important piece of software or hardware that is monitored and logged by the SOC teams.

2.4.3 Honeypots

The most common definition of a Honeypot (in the sense of a network device) is, that it is a trap created for any kind of unwanted activity, that could affect the network and systems of an organization.

Such traps come in many forms. There are email traps, that place special email addresses in places, where only bots or malicious actors could find them. Emails coming to such inboxes are then analyzed. If any such email later comes to the legitimate mail inbox of an employee, it is automatically treated as a malicious or unwanted message and therefore removed immediately. There are also network traps, that create an illusion of a system vulnerability in the network communication or internet services. These include decoy databases that are focused on SQL vulnerabilities or privilege abuse on a database. Other ones are malware honeypots that mimic apps and APIs to catch malware samples that can be later analyzed by a reverse engineering specialist or a forensics analyst to mitigate such attacks on the real infrastructure or to develop tools to detect such intrusion attempts in the future. [17]

We can divide almost all of the above-mentioned honeypots into two main types: low interaction and high interaction. These differ in the depth of possible interaction of the system with the attacker or malicious actor. Low interaction honeypots only simulate a certain small number of internet and network protocols, but the system does not offer any advanced communication with the decoy system. Therefore, the usual attacker will soon lose interest in this honeypot and the information obtained from the interaction probably won't be too detailed. On the other hand, high-interaction honeypots are created in such a way that the attacker will be kept interested for a much longer period. Therefore, the information obtained from such interaction can bring much more information and indicators of how the attack would take place on a real system of the company or organization. [18]

There are two important roles of the honeypots. They for one serve as the decoys, that are more motivating for attackers, bots, or spammers for the intrusion than the better-protected standard systems. The second crucial role is the support for threat hunting. Attackers send their malware or malicious payloads to the allegedly vulnerable server, which sends them to another system and the SOC team for the analysis of intrusion procedures, that could have been used against a real target in the network. Moreover, even the fact that some address was accessing the honeypot is very valuable on its own, because any later communication from such attacking address to real infrastructure can be later reviewed because it can be automatically tagged as suspicious.

2.4.4 Other systems

Depending on the size of the organization the SOC team works for, other technologies can be monitored. One example of such a system is servers.

2.4.4.1 Auditing systems

Another service that should be monitored by a common SOC team is the Auditing services. Those work inside the common operating systems and do collect logs about the activity happening in the system. Examples of such auditing systems are auditd (Linux Audit Daemon) or Windows auditing software. Various events inside these systems should be (and usually are) logged [19]:

- Logon and logoff events of a user
- Account management
- Server access and logins.
- Object access (objects as files or I/O devices)
- Registry access
- System events such as power-on or off

2.4.4.2 Servers

There are many types of activity on the servers that need to be monitored. These include: authentication (brute force attacks to administrator accounts for example), file system changes (like replacing legitimate files with their malicious counterparts), or path traversals to protected files [20] (like a CVE-2021-41773 Apache HTTP Server Path Traversal and Remote Code Execution that tries to access files with logins and password hashes [21]). For this, a typical SOC team can log the activity of the servers (e.g. network requests, file auditing logs, settings changes), however, sometimes the access of the SOC team might be limited and such logs are accessible only through cooperation with other teams that provide the technical support for such servers. On the other hand, the monitoring of servers might be partially replaced by the Application layer gateways (as described in the section Firewalls and Next Generation firewalls), that can monitor the network traffic of the server.

2.4.4.3 Cloud systems

Especially for smaller-sized organizations cloud systems are an easier way to provide some services either to their customers or their employees. Therefore, a company cloud can be a vital part of the organization's infrastructure and therefore should be monitored as well as any other system for any deviation from the normal operation. Even the threats for such cloud operations are mostly similar to the ones that are posed by traditional on-premise solutions, however, some additional challenges arise. Those are for example misconfiguration of specialized cloud technologies like cloud data buckets (like the BlueBleed leak in 2022, when sensitive data of 65,000 entities became public after a wrong configuration [22]). Therefore, an organization needs to collect the logs from these systems and analyze them in the normal way.

One of the examples of systems that are usually located inside the cloud infrastructure is CASB (Cloud Access Security Broker). *"The CASB serves as a policy enforcement center, consolidating multiple types of security policy enforcement and applying them to everything your business utilizes in the cloud—regardless of what sort of device is attempting to access it, including unmanaged smartphones, IoT devices, or personal laptops."*[23]

2.5 How are the logs collected?

All the previously mentioned technologies need to be monitored constantly, therefore any SOC team needs to collect logs and process them as quickly as possible. There are two main different ways, in which the logs are collected: Agent-based and Agentless. These deeply differ, as the latter provides a higher level of security to the security infrastructure itself. This is because, in the case of agent-based monitoring, we can encrypt the log data and secure the communication. Therefore any possible attack vector against the security infrastructure is less likely. [24] However, in the case of agentless monitoring (which still needs a lower-level agent to exchange the data throughout the network), administrators need to rely on some standards, which include those, that are unsafe for today's deployment. However, some of the devices do not support agent-based communication and therefore many of today's security infrastructures include a mix of the above-mentioned approaches.

2.6 Conclusion of monitored technologies

Many different types of devices might be monitored by the SOC team. Their tasks are different and so are the threats such devices are trying to prevent. A typical organization‘s network uses many such devices (like firewalls, IPS, VPN, and servers) and therefore we need to properly work with logs from such devices, understand them, know their limitations and capabilities, and can respond to any issues correctly and swiftly. This poses many challenges to the functioning of the SOC team, that need to be overcome.

Chapter 3

Monitoring technologies

As we have already established the technologies that are being monitored by the SOC, we need cutting-edge technologies that provide real-time visibility into the organization's network and systems. SOC constantly monitors network traffic, logs, and activities of the users and accounts in the network, however, such information needs to be presented in a well-arranged and systematized way so that the personnel can react swiftly and in the best way possible to any threat to the infrastructure. For this reason, we need a specialized group of software including a Security Information and Event Management (SIEM), a log management solution, parsers (devices that take in the incoming log and process it to a parsed information), connectors, and sometimes Security orchestration, automation, and response (SOAR). This chapter aims to describe all these systems in a well-arranged way.

3.1 The common SOC structure

When we are building a new SOC, we usually only have the systems that are to be monitored. In order to build a security infrastructure on top of these existing structures, we need a good understanding of the systems that we are going to monitor. With this information in mind, we can start designing the SOC infrastructure.

The most crucial part of the SOC infrastructure is without any doubt Log Management because it encompasses most of the SOC team infrastructure work. Therefore, when designing, we need to start with the procedure of log collection and processing. First of all, we need to be able to collect the logs from all parts of the network. This might be particularly challenging if the organization's network is vast and segmented (therefore special rules on the switch and router infrastructure need to be set). When we have the routes the logs will travel through, we need to set up the end devices to send logs and the opposite side to collect the incoming logs. The device that needs to collect the logs is usually called a connector. Its job is to provide a connection point to where the logs flow from any remote devices (either another connector that serves as a relay or the monitored technology itself). However, the implementation of the log collection can vary greatly. Sometimes special software is used for the collection (sometimes called the log collectors). It depends on the specific implementation of the connector, but it will usually transform the incoming messages to the CEF (Common Event Format). "*The CEF is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications. CEF enables customers to use a common event log format so that data can easily be collected and aggregated for analysis by an enterprise management system.*"[25]

When we have collected the logs, we need to store them and (sometimes) process them into a more accessible (parsed) format. Storage of logs, especially for bigger networks, can pose a huge problem. Security logs might need to be kept for a comparatively long time (especially if the log storage is required by law). Depending on the law and regulations, the log retention period for the most critical logs may range between six months and seven years [26]. Such storage therefore needs to have enough capacity to store such volumes of logs, but also should be reasonably fast for the work of the SOC team (for example for the case of an analysis of any older security incident).

If we have the logs stored and parsed to fields, we need to apply the set of detection rules to identify any possible threats or any deviation from the standard operation of the systems that are under the monitoring of the SOC team. This is the work of the SIEM or the SOAR.

3.2 Log management

Log management is where the most work of SOC operations happens because it needs to cover many processes that are essential for incident detection and response. There are many tasks that a log management solution needs to fulfill in order to function properly. These include [27]:

- Log collection, which needs to obtain and aggregate logs from various sources across the network
- Monitoring, which handles tracking events and activity of the monitored systems
- Analysis, which oversees the log collection and detects bugs
- Retention, which handles the data storage for longer periods of time
- Indexing and searching, which server for the purposes of the analysis and the filtering of the logs
- Reporting, that servers mainly for the purposes of audits and operational compliance

3.2.1 Key parameters of log management design

Therefore, when we need to build a log management infrastructure from scratch or upgrade an existing one, we need to establish the following:

- Which systems will be monitored? Moreover, are there any other systems that we know they might need to get connected in the lifetime of this logging solution?
- How detailed logs will be provided (some systems like firewalls offer to log only certain events)?
- For how long do we need to store the logs? This can vary greatly and can also depend on the legislature of the country of operation.
- How often will the logs be downloaded? It might not be a continuous logs flow, but the logs can be loaded in batches. This is especially important for the network environment in the vicinity of the log management solution.

Second, we need to identify which log formats will be sent to our systems (it depends on the connected systems, however, some of them support more than one format of logs), so that we can process and analyze them properly. With the log formats identified, we need to process the logs (which means extracting the information fields from a plain text log format or a log standardization to the chosen format). That is usually done automatically by the log management solution by default, however, if an organization uses lots of different technologies (and therefore probably more log standards), it might be a very challenging task to keep all the logs from the technologies flowing and processed. For some standardized formats of logs (like JSON, CEF, or Extended Log Format) the log management solutions usually have prebuilt solutions for log processing inside it, [28] however sometimes the organization uses such obsolete or obscure technologies, that it is necessary to write device-specific rules. ”*Typically this is done using regular expressions or the logging solution’s proprietary language.*”[28] Moreover, the format of a log from a certain device can change (usually after a system update), and then it is needed to provide good maintenance of such systems because otherwise, the SOC team could lose visibility to some systems.

3.3 Security information and event management

For the correct work of the SOC, we need to provide information to the SOC team members. In order not to overload the members, only events detected by the preset rules about malicious or suspicious activity are sent to the human operator for analysis. This sorting and rule matching is the work of the Security information and event management software (or SIEM).

3.3.1 How does the SIEM operate?

”*At the most basic level, all SIEM solutions perform some level of data aggregation, consolidation, and sorting functions in order to identify threats and adhere to data compliance requirements.*”[29] Therefore, in order to operate correctly, it needs to provide some basic features, which are [30]:

- Correlation rules for threat detection
- The ability to search in the limited volumes of logs
- Selective ingest of security-related logs from other sources
- Security-related reporting features (for example monthly reports about certain vulnerabilities that the attackers are trying to abuse)

Only after such features are provided, the solution we are talking about can be considered a SIEM solution.

When we have a SIEM, we also need to either use detection rules that are provided with the solution, we need to obtain third-party rules, or create custom ones. These rules can detect many types of activity (HTTP-based attacks, account attacks, suspicious VPN logins, or others), however, it is deeply dependent on the devices that are monitored. If we need to monitor attacks against web servers, we need to create custom rules for the logs coming either from the IPS/IDS or the Web Application Firewall. If we need to monitor the VPN brute force type of attack, we need logs from the VPN concentrator. Therefore, many incidents can be detected, but we are required to have enough logs from the right devices and custom rules that are specially crafted for the right type of device and SIEM solution.

Moreover, we might want to add additional information to SIEM in order to detect even more suspicious events. Additional information usually consists of Indicators of Compromise (IoCs), which can for example include:

- Unusual Network Traffic (especially to and from IP addresses that are linked to malicious activities or those that tried to communicate to honeypots)
- Unusual IP destination ports (both from and into the network)
- Login attempts to accounts with previously leaked login or password

- Suspicious websites
- Abnormal outgoing communication from a small number of computers

3.4 Security orchestration, automation, and response

Security orchestration, automation, and response is a technology that wants to simplify the work of analysts through response automation. It is very similar in capabilities to SIEM, however, it adds the crucial capability to respond automatically according to the set rules without a human operator.[31]

The operations of SOAR are divided, as the name suggests, into three areas:

- Orchestration - It serves for the combination of information from the internal network with the tools of open source intelligence and publicly available information. ”*Connected systems may include vulnerability scanners, endpoint protection products, user and entity behavior analytics, firewalls, intrusion detection, and intrusion prevention systems (IDSes/IPSes), security information and event management (SIEM) platforms, endpoint security software, external threat intelligence feeds, and other third-party sources.*”[32]
- Automation - Many tasks of the SOC team analyst are tedious and therefore can be done by automation to save human operator’s time. ”*Security automation, fed by the data and alerts collected from security orchestration, ingests and analyzes data and creates repeated, automated processes to replace manual processes.*”[32]
- Response - based on the parameters set, it can decide about a security event without a human operator and take appropriate action. Therefore, orchestration and automation procedures are employed.

3.5 Infrastructure monitoring

As the infrastructure of a Security operation center can be huge, it is needed to monitor the functioning of all the operational devices and the network in between. This monitoring job can be done by infrastructure monitoring teams, but usually, it needs to be done by a SOC team. However, this greatly depends on whether the issues are related to the administration provided by the SOC Engineers. However, the problem might be based on the underlying infrastructure, to which the SOC team might not have any access (for example in the case of virtual servers on physical hardware, where the machine is administered by the SOC team, but the Hardware system is administered by some infrastructure team).

3.6 SOC infrastructure conclusion

The SOC infrastructure might get quite complex for large networks, but the whole operation still has to be based on log management and sophisticated analysis. Therefore, all the technologies either transfer, store, analyze, or provide access to the logs, or they are monitoring the functioning of such log devices.

Chapter 4

SOC technologies deployment and configuration in the lab conditions

SOC deployment into an existing environment can be very challenging due to varying systems of different types, unsupported technologies, and the organization of the computer network. This chapter describes how a complex of security systems and services can be deployed and used for detailed monitoring of activity on a simple lab network with several devices of different types.

4.1 Steps of the deployment of the network and security monitoring

For the purpose of this thesis, first, a lab network environment needed to be established. The primary parameter was given for the operation of this environment: It needed to be separated from the normal operations of the surrounding network and its devices (in order not to leak any private data during the work on this thesis or to accidentally compromise any other outer systems during the simulated attack). The second parameter was centered around the minimal cost of the whole lab and monitoring solution and the network itself.

So the steps for the deployment were the following:

1. Choice of technologies and systems that would be present inside a typical network of any organization. Therefore, a routing solution with firewall and IPS capabilities has been deployed. A DNS solution was also needed for the purposes of local domain resolution. Additionally, a web server solution has been configured, because that is an important piece of technology that could be present in a typical network. To represent endpoints, that would be present inside a typical network of any organization, several different operating systems have been deployed.

   Moreover, an interconnecting network for the aforementioned devices needed to be established. Because the network is a mix of virtual and physical machines, a simple unmanaged switch and another virtual unmanaged one have been used to provide connection.

2. The second step was the choice of the security monitoring solution. There is a large variety of different solutions available on the market that differ by the features provided but also license cost.

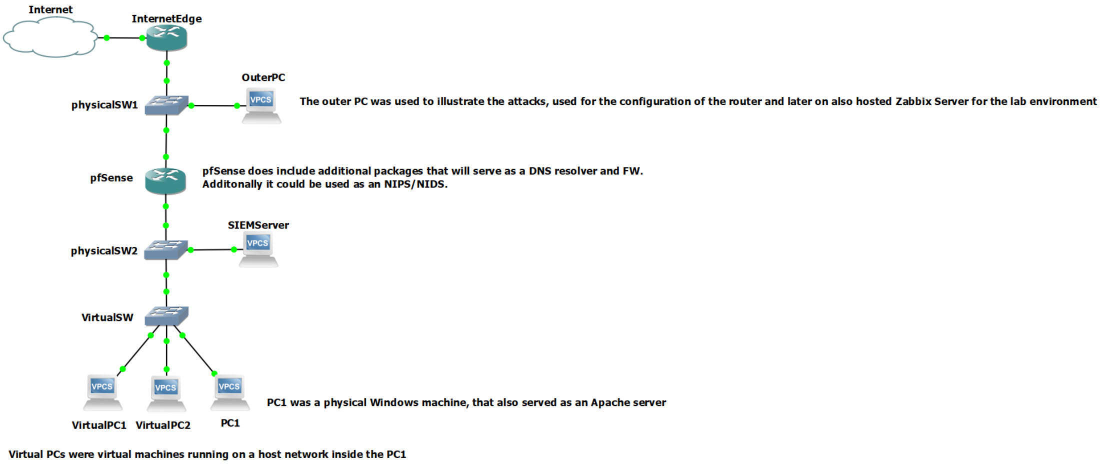
3. Deployment of the chosen SIEM and log management central components on a dedicated server.

4. Deployment of monitoring agents to devices selected for monitoring. However, thanks to the choice of router operating system, we also had to connect the device for which there is not any official agent provided, therefore we had to connect this device through syslog (therefore agentless) log-sending service.
5. Tuning of the central monitoring solution (rules adding and editing) for detection of a few selected types of attack, that will be shown in the attack stage.
6. Deployment of a network operation monitoring solution for the control of operations on the router. This solution later provides graphs and monitoring with alarm triggers for key operational parameters (like usage of disk, RAM, and network connections) of monitored systems. This solution provides us for example with an option to detect high-volume internet traffic thanks to custom trigger settings.
7. Several simulated attacks or suspicious activities of different types were carried out in this part of this section in order to demonstrate the detection by the deployed solutions.
8. In the next part, other capabilities of the deployed solutions that were not needed for the topic of this thesis were described. However, several challenges for the deployment and use of these systems were also presented.
9. Finally, options for further development of the network and monitoring solution were proposed and described. An evaluation of the operation of all the deployed technologies is provided.

4.2 Lab environment establishment

The lab environment was designed to illustrate the usage of the most common systems inside big companies' or organizations' networks. Therefore, this environment consists of a router with a firewall, a simple DNS resolving service (that simply forwards any request except local ones to the predefined public DNS servers), and IPS/IDS capabilities. Other lab network devices are a simple physical unmanaged switch and another virtual switch that serves for the connection of several virtual machines deployed on a laptop.

Figure 4.1 The lab environment network diagram that has been created using GNS3 software



The outer PC was used to illustrate the attacks, used for the configuration of the router and later on also hosted Zabbix Server for the lab environment

pfSense does include additional packages that will serve as a DNS resolver and FW.
Additonally it could be used as an NIPS/NIDS.

PC1 was a physical Windows machine, that also served as an Apache server

Virtual PCs were virtual machines running on a host network inside the PC1

This network has only one wide area network connection to the internet through the router, on which no routing protocols with neighboring routers have been set up. It has only been given one default gateway (another router that we could call an Internet Edge router), that is accessible through another switched network. Therefore, the network has been mostly independent and self-sufficient, especially due to the nature of activities that have been done inside the network (simulated attack and firewall settings that could collide with normal operations of a network. So, to conclude, the lab network is connected to the internet but mostly separated from the local network traffic.

For the server, log management, and endpoint devices, a log management server with all the central components was deployed on an old PC, several Linux or Windows machines simulating typical endpoints, and finally, a web server that has been set up on the Windows machine.

4.2.1 Router and network devices

Many different router solutions are available in today's market. Such solutions differ greatly in features, types of deployment (physical or virtualized), the maximum speed of ports, available add-on packages, the maximum number of users for certain services, license prices, or the cost of purchase. Many business-grade technologies have been considered for this lab environment router deployment (like Cisco NGFW [that provides combined services of firewall, IPS, router, VPN concentrator, etc.] or Ubiquiti Unifi next-generation gateway). Such solutions are very complex, powerful, and reliable, however, as such they usually have one thing in common: high price. Therefore, at least a little bit more accessible and cost-effective solution had to be chosen. Therefore, the focus shifted towards routing operating systems that provide free licenses. Therefore, pfSense, OPNSense, and OpenWRT solutions have all been considered and compared. pfSense and OPNSense are very similar in terms of capabilities and support[33], but the OpenWRT was eventually ruled out due to a lower level of support in the topics this thesis needed to consider. After a careful comparison of the documentation, functions, and available support, the pfSense operating system solution was finally chosen. pfSense (a FreeBSD distribution-based operating system) provides all the business-grade services like IPS (Suricata or Snort add-on packages), firewall (out of the box), and basic DNS blackholing and resolving services (pfBlockerNG add-on package)[34] and can therefore provide all the services that would likely be deployed in a typical business-oriented network.

4.2.1.1 Hardware for pfSense router

The pfSense solution is available either as an OS for normal PC hardware or can be bought with hardware in the case of Netgate devices. For the purpose of this thesis, the first option has been chosen, and therefore a compatible custom PC with compatible hardware was needed to be built.

Our pfSense router runs on a custom-built computer with two 2.5Gbps ethernet ports. The choice of the NIC (Network interface card) for this solution is crucial because pfSense tends to be a little bit selective about NICs (it is given by the support of cards by the FreeBSD OS) and usually supports only some (Intel NICs are usually a good choice because their drivers are supported out of the box) [35]. The rest of the build is made up of common customer-grade components (Intel i5 CPU, 16 GB DDR4 RAM, 500GB NVMe SSD drive), that provide more than enough capacity and power for the operations of the router and other needed modules.

4.2.1.2 Lab network topology

The network topology of the lab environment is simple, as it mostly, but not entirely uses a core collapsed network topology. pfSense router creates the Core layer and an unmanaged switch serves as an access layer. To this switch, all the endpoint devices have been connected. Moreover, the Wazuh central server is connected to the same switch. For any real-life deployment, a better network infrastructure model would be highly recommended (for example any endpoint device has direct access to the Wazuh dashboard, which would be highly undesirable because of security issues), but for the purpose of this thesis, this simple network design is more than sufficient.

4.2.1.3 Setting up network devices

The only network device that needed to be configured (thanks to the use of an unmanaged switch) was the router. The pfSense operating system is not very hard to deploy. Installation of the operating system is very similar to any Linux OS type of system. After the installation, basic things need to be configured from the command line. At least we need to configure WAN and LAN ports in order to be able to access the web user interface for further settings. This might be complicated only because of the misidentification of the hardware ports located on the device.

After basic command line configuration, we are able to get to the web UI to access settings for the firewall, DNS, and other packages. For the purpose of this thesis, we set the DNS resolver to resolve addresses inside the local domain in order to simplify access to such devices. For example, we later set host override for SIEM at address wazuh.prague.local.

Moreover, we set an IPS. There are two different packages available for Wazuh providing such capabilities: Suricata and Snort. Suricata was finally chosen, however in the end it didn't play any key role for this thesis because all the security testing was done locally. However, the logs from Suricata have been set to be added to the Firewall security logs that were being continually sent to the Wazuh solution for the detection of suspicious activities.

The firewall inside pfSense came with several already configured rules, however, some need to be added later in order to provide access to some technologies (access to non-standard ports later needed for network operations monitoring). In addition, some rules were added in order to block access to a specific public IP address (this was done in order to demonstrate the detection of potentially unwanted access from the inside of the network).

4.2.2 Endpoints

For the purpose of this thesis, two common personal computer operating systems are deployed: Windows 11 on physical hardware and Linux (both CentOS and OpenSUSE) as virtual machines. Other Linux distributions (Debian and Solaris) have been installed, but later there were major issues with Wazuh agent installation (more on that in section *Limitations of the deployed SIEM solution*).

Choosing Windows as one of the operating systems however poses a threat, because log collection from Windows devices (mostly on larger scales and only for some log management solutions) tends to be a little more challenging than from other considered systems. However, as it was found out later, the chosen log management solution Wazuh works very well with Windows logs without any issues.

No special settings or policies have been deployed on these endpoint machines.

4.2.3 Web server

For the lab environment of this thesis, several web server solutions have been considered for the task. This group consists of the ones, that are most common in today's typical deployments: IIS, Apache, Apache Tomcat, and Nginx. All of those previously mentioned would have been a very good choice, however, Apache has been chosen because it is one of the most used and supported ones across the internet today. Moreover, specific proof of concept of an attack (and a specific corresponding vulnerability) has been chosen shortly before the web server solution, and this attack targets solely Apache servers. Therefore, the Apache Haus distribution binary has been deployed on the Windows machine that also serves as an endpoint device (only to provide basic service, in the real network a dedicated machine would have been used).

4.3 Log infrastructure and SIEM choice

The choice of security infrastructure was very complicated because there are many complex solutions available on the market. This section will explain how the solution for the purpose of this topic was chosen and what parameters it was based on.

Many SIEM and log management solutions were considered for the purpose of this thesis as they differ in provided features and capabilities. There are many business-grade solutions available on the market, but they usually require costly licenses or they offer very limited trial versions with data volume limits. Moreover, some of these solutions are cloud-based, which would collide with the goal of the network being solely laboratory and separated from normal operations of the outer networks. The following paid solutions were shortly considered: IBM Security QRadar, AlienVault USM, Micro Focus ArcSight, and Splunk Enterprise SIEM. The pricing differs, but for example, IBM Security QRadar's license for a network of this size would cost around 500$ per month of license[36]. This cost would be too high for the character of this project.

Therefore, the focus shifted towards the free products on the market. These included: AlienVault OSSIM, OSSEC, and Wazuh. Many properties of these solutions were compared, but after careful consideration (mostly thanks to native log agents and support for many types of devices), the Wazuh solution was chosen for deployment. This solution covers all the parts the theoretical SIEM and logger solution needs to provide: it receives the logs, analyzes them, and reports issues to the human analyst. Therefore, this solution consists of the Wazuh indexer, Wazuh server, and Wazuh dashboard, whose roles in the entire solution are described below.

4.3.1 Wazuh components

The Wazuh indexer is the main component, as it serves for full-text searching and analysis of the logs. It indexes and stores alerts generated by any monitored technologies in the network. [37]

The Wazuh server is used to analyze the data coming from registered Wazuh agents and it raises alerts in case of identified anomalies or potentially unwanted activity. [38]

The Wazuh dashboard is used for web access to the alerts, data visualization, monitoring of events, generating reports, and the configuration of the entire solution.[39]

Therefore, the aforementioned structure of separated log management and SIEM solution that co-create the most important and complex system any working SOC team can have is well illustrated in this case.

4.4 Central log manager deployment

For the purpose of deployment of the central Wazuh server and old PC was reinstalled to the Ubuntu operating system. Wazuh supports installation to Linux 64-bit OS, but Ubuntu was amongst the ones recommended for deployment. For the purpose of this thesis, a single-node physical machine configuration has been chosen. However, this would be highly problematic for any real environment network, because of the requirements for high service availability. Also, the lab environment network is a small one (with a small number of agents), and therefore an old PC with a 4-core CPU is sufficient for the operations. However, a more powerful machine would be required for larger networks in order to withstand a significantly higher load on the central components.

After fulfilling all the requirements for the proper installation, we have downloaded the installation package available from the official Wazuh website. There are also other options available like the Docker container or the OVA virtual machine [40]. Installation package versions with more detailed control over installation procedures are available for each central component, however, for the purpose of this thesis, the basic installation is sufficient. Therefore, after a simple installation procedure that was done by the Wazuh installation assistant, we prepared the central Wazuh server with all the aforementioned components for operations.

After the installation, we can log into the web user interface (which is provided by the dashboard component on port 443), where we immediately see that no agent is currently reporting (we have not yet set any to do so) to the central server and therefore no alerts except the ones from the server itself have been generated. Therefore, we need to connect all the devices that have been chosen to get monitored using either agents or syslog.

4.5 Endpoint log agents deployment

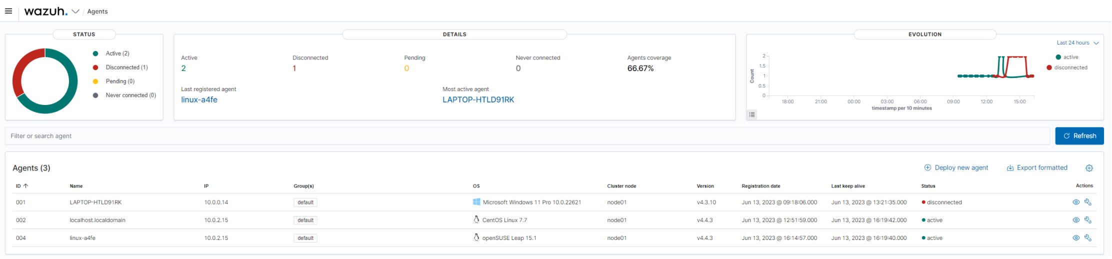
In order to monitor the endpoint devices, the Wazuh security solution is centered around agent-based monitoring. Therefore, we will need to install the official Wazuh agent on as many devices as it is possible. Other devices can be monitored through Syslog, however, this solution is not optimal (more on this issue in the following section).

The web user interface provides options for the installation of agents, where it lets you choose the operating system, the instruction set architecture the agent will run on, and deployment-specific parameters (like the address or fully qualified domain name of the Wazuh central component, device hostname or the group name of the specific agent). However, as it is mentioned later in the section *Incomplete, obsolete or flawed documentation*, the steps provided by the UI sometimes contain errors and reference only the agent version available probably at the time of build of the central components. However, newer versions of the agent are available (as they are referenced in the updated parts of the Wazuh public web documentation) and can be used even with older central components.

After choosing the correct version, you get the download of the correct package of an agent for your specified system. The installation procedure is described on the website and after setting the basic aforementioned parameters (hostname, group name, etc.), the installation should go through without any difficulties. However, during the installation of the Debian packages, insurmountable obstacles have been encountered (the Debian package manager reported the package as a broken one that cannot be installed). Another issue was encountered during the installation of another package to the Solaris operating system, where the application was unable to start the Wazuh service.

However, the installation was completed without any issues on the Microsoft Windows 11 operating system and with smaller obstacles (problems with starting the service) on the operating systems OpenSUSE and CentOS. After the installation on these devices, we might need to update the configuration of the monitored log files inside the system (typical log files are pre-configured, however any log file created by non-standard applications [like the web server] needs to be added to the XML file for log collection). Therefore, after we deployed the Apache web server on the Windows 11 machine, a new file location was added to the configuration of the corresponding agent configuration file.

Figure 4.2 The connected agents list of the SIEM after agents connection



Active 2

Disconnected 1

Pending 0

Never connected 0

Agents coverage 66.67%

Last registered agent linux-a4fe

Most active agent LAPTOP-HTLD91RK

Agents (3)

ID ↑	Name	IP	Group(s)	OS	Cluster node	Version	Registration date	Last keep alive	Status	Actions
001	LAPTOP-HTLD91RK	10.0.0.14	default	Microsoft Windows 11 Pro 10.0.22621	node01	v4.3.10	Jun 13, 2023 @ 09:18:06.000	Jun 13, 2023 @ 13:21:35.000	● disconnected	
002	localhost.localdomain	10.0.2.15	default	CentOS Linux 7.7	node01	v4.4.3	Jun 13, 2023 @ 12:51:59.000	Jun 13, 2023 @ 16:19:42.000	● active	
004	linux-a4fe	10.0.2.15	default	openSUSE Leap 15.1	node01	v4.4.3	Jun 13, 2023 @ 16:14:57.000	Jun 13, 2023 @ 16:19:40.000	● active	

4.6 Connecting devices without official log agent support

The situation of connecting devices without official Wazuh agent support is a little bit more complicated. The pfSense can work with an agent that is created for the standard FreeBSD OS [41], however "*FreeBSD package repositories are disabled by default on pfSense installations, as Netgate views unofficial software as a security and compatibility concern.*"[41] Therefore, another more secure, however less practical workaround has been chosen.

As the Wazuh documentation states, there are two ways of collecting remote logs not sent by the Wazuh agent. Option one is the collection of logs directly by Wazuh on port 514, Second option is the collection by an independent syslog daemon, which saves collected logs to a specified file, which is monitored by Wazuh [42]. The second option has been chosen in order to separate the process of log transfer from Wazuh (therefore in case of Wazuh failure, we would still be at least able to collect logs).

The logs collected on the pfSense router are gathered by the operating system itself and then sent through the syslog connection to the Wazuh central components server, where the syslog daemon has been set to save received logs to a specific file. This file is monitored by the Wazuh manager itself thanks to the change of the configuration. The configuration has been changed the same way as if it was a normal agent configuration file. However, this workaround brings a drawback: the logs inside the Wazuh system are then presented as if they were created by the Wazuh server itself.

One device in the entire monitoring system that requires an agentless solution does not pose a big problem, however, if the network contained more than one such device, the logs collection would get very confusing. In order to keep them well ordered, we would need to create a specialized connector for every incoming Syslog connection where the agent would collect logs from one Syslog device and his own. It is a question, however, if such a situation would not require a different SIEM and log management solution which would be more suitable for such a predominantly Syslog (or similar) connection-based monitoring system.

4.7 Wazuh SIEM capabilities

The Wazuh system is not only a SIEM solution, however, it claims to provide services of an XDR (Extended detection and response). Therefore, it provides services beyond the area of security logging and alerting. Such additional capabilities include vulnerability detection, file integrity monitoring, regulatory compliance reports (like GDPR compliance), or malware detection [43]. Therefore, Wazuh is a very versatile tool, however, for the purpose of this work, only the security logging and monitoring tools have been used. A vulnerability scan has been also tried, but no vulnerabilities have been detected on any of the devices (all of them are updated regularly and almost no applications run on them).

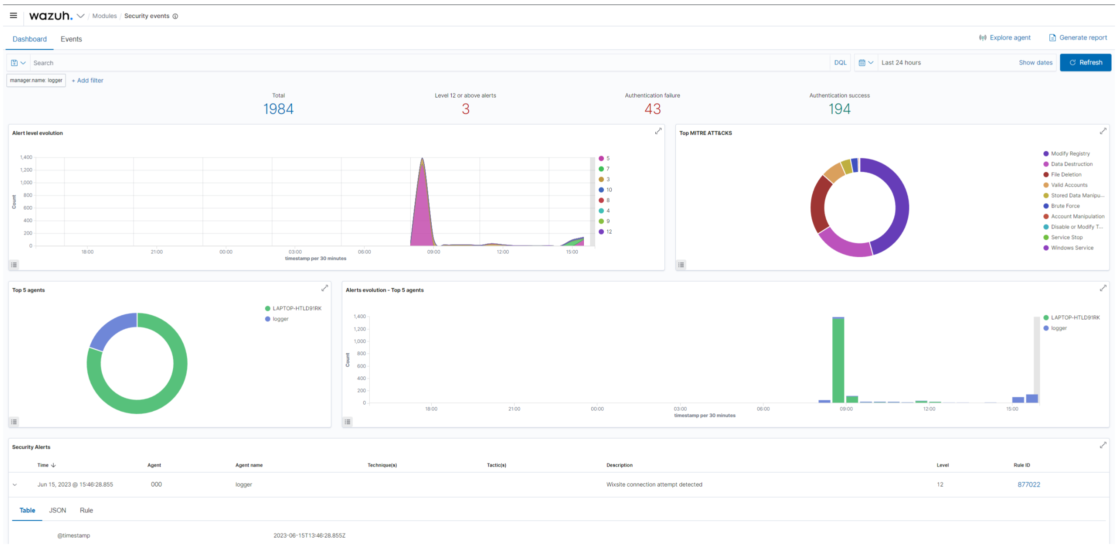
4.8 Limitations of the deployed SIEM solution

During the deployment of the Wazuh monitoring solution, several weak points, which limit the real-life deployment options, have been discovered. This section will elaborate on the most important ones and, if possible, will provide workaround tips, that have been discovered.

4.8.1 Connection of devices without official agent support

As mentioned in the section *Connecting devices without official agent support*, none of the standards for log sending between devices without an agent is supported. In most of the other

Figure 4.3 The main dashboard of the Wazuh SIEM



manager.name: logger

Total	Level 12 or above alerts	Authentication failure	Authentication success
1984	3	43	194

Security Alerts

Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description	Level	Rule ID
Jun 15, 2023 @ 15:46:28.855	000	logger			Wixsite connection attempt detected	12	877022

@timestamp 2023-06-15T13:46:28.855Z

available log management solutions, however, this agent-less log collection is possible and therefore the suitability of the Wazuh solution for specific network needs is to be carefully considered due to this functionality limitation. Workaround here is possible through the creation of virtual machines, that would serve as log relays (they would receive the logs through standard agent-less channels and send them further with their own agent). However, this solution is scalable only with difficulties, and for large networks could be completely unfeasible.

4.8.2 Missing functions

Some features of the typical SIEM solutions are missing. The chosen feature set suits well smaller SIEM deployments, however, could be limiting in larger implementations. Such features include the ability to review all logs, not only the ones that have been selected by the Wazuh algorithm based on the preset rules. Workaround here is possible: either provide selected people direct access to the server, which in fact stores all the logs that have been collected or enable SSH connection. However, either one of these solutions is only partial and provides further security challenges.

Moreover, if the cooperation of more people on the alert processing is required, good communication would be required, because the Wazuh solution doesn't provide triage capabilities (tagging the stage of the alert and naming the analyst who is responsible for the processing of this specific alert). Therefore, working on the basis of one alert stream could bring significant challenges to the operation of any real-life SOC team.

4.8.3 Incomplete, obsolete or flawed documentation

When deploying an agent from the central management component (from the central server), the installation guides usually reference old versions of the agent for installation. Some documentation uploaded in the online documentation server has small bugs that need to be eliminated in order to finish the installation successfully.

4.8.4 Agents deployment on Debian-based systems

During an installation of the agent solution to a virtual machine running on Debian OS, an error was shown by the package manager of the Debian system. This error prevents the installation of the package because of its alleged damage (it is not registered as a valid DBM package). A workaround solution to this issue has not been found by the author of this thesis up to this day.

4.9 SIEM rules and detection

This section explains how the Wazuh SIEM detects any suspicious activity and how exactly are the logs processed inside it. Moreover, it will show how the trigger rules work and how can they be edited or created in order to detect specific unwanted activities inside the monitored network.

4.9.1 How are the logs processed by Wazuh?

The logs come into the log management as a long string of data that we can call a raw log.

Code listing 4.1 Example of a ”raw” log from pfSense firewall

```
2023-06-19T18:47:37.410554+02:00 AlphaRouter.prague.alpha filterlog
    [38091] 4,,,1000000103,igc1,match,block,in,4,0x0,,255,58005,0,none
    ,17,udp,125,192.168.37.150,224.0.0.251,5353,5353,105
```

Such string needs to be split into different fields in order to be accessible and meaningful for people. Therefore. the log processing inside Wazuh SIEM and log management has three different phases: pre-decoding, decoding, and rule matching phase.

Pre-decoding phase is responsible for the identification (extraction) of timestamp, hostname, and program information from the collected log[42].

Code listing 4.2 Phase 1 actions can be illustrated using the Ruleset test function output of the Wazuh Dashboard

```
**Phase 1: Completed pre-decoding.
        full event: '2023-06-19T18:47:37.410554+02:00 AlphaRouter.prague
            .alpha filterlog[38091] 4,,,1000000103,igc1,match,block,in
            ,4,0x0,,255,58005,0,none,17,udp
            ,125,192.168.37.150,224.0.0.251,5353,5353,105'
        timestamp: '2023-06-19T18:47:37.410554+02:00'
        program_name: 'filterlog'
```

The decoding phase finds the proper decoder (if available, but there are plenty of them already prepared) based on already known data and sends the log to it for processing (parsing the rest of the data)[42]. Such log data are then available for searches based on specific parameters like source IP address or others.

Code listing 4.3 Phase 2 actions can be illustrated using the Ruleset test function output of the Wazuh Dashboard

```
**Phase 2: Completed decoding.
        name: 'pf'
        action: 'block'
        dstip: '224.0.0.251'
        dstport: '5353'
        id: '1000000103'
        length: '105'
        protocol: ’udp’
        srcip: ’192.168.37.150’
        srcport: ’5353’
```

The rule matching phase is responsible for the detection of anomalies based on the rules inside the SIEM[42].

Code listing 4.4 Phase 3 actions can be illustrated using the Ruleset test function output of the Wazuh Dashboard

```
**Phase 3: Completed filtering (rules).
        id: ’87701’
        level: ’5’
        description: ’pfSense firewall drop event.’
        groups: ’["pfSense","firewall_block"]’
        firedtimes: ’1’
        gpg13: ’["4.12"]’
        hipaa: ’["164.312.a.1"]’
        mail: ’false’
        nist_800_53: ’["SC.7"]’
        pci_dss: ’["1.4"]’
        tsc: ’["CC6.7","CC6.8"]’
**Alert to be generated.
```

There was a rule triggered and therefore an alert would appear on the dashboard and would be investigated by a human operator.

4.9.2 Wazuh rules

The Wazuh system comes with a vast amount of already prepared rules for many different operating systems that are ready to detect suspicious activities across the network. An example of such a prepared rule is the detection of logs that represent unsuccessful login on Linux or Windows machines. This rule will be useful later for the simulated brute force attack on the Debian server.

However, to detect properly the other two attacks or anomalies that have been prepared in order to illustrate the functioning of the entire system and its capabilities to detect such unwanted activities.

Later, two rule sets were edited inside the SIEM (one rule was created and one was edited in order to detect even more signatures), so that we could later detect the aforementioned attack attempts properly.

The detection rules inside the Wazuh system use an XML syntax and are therefore quite easy to understand and write:

Code listing 4.5 Custom created rule ID 877022

```
  <rule id="877022" level="12" frequency="2" timeframe="120" ignore="5">
    <if_matched_sid>87701</if_matched_sid>
    <dstip>199.15.163.145</dstip>
    <description>Outlook phishing wixsite connection</description>
  </rule>
```

Above, we can see one of the rules that have been created during this project. This one detects a connection to a specific destination IP address, that would have been detected in the past by the SOC team as a malicious one.

Now we will elaborate on some of the parameters or options that can be passed to any rule inside Wazuh. The level of rule signifies the importance of such an alert. The frequency parameter sets the number of times the rule has to be triggered by the log to be sent to the dashboard, the closely related parameter is the time frame (the time during which the alerts are counted). The ignore parameter sets the number of seconds after which this rule is not active (in case of a previous alert to the dashboard). This is done in order not to flood the dashboard with the same alert many times.

There are many parameters defined, that can be used to trigger the alert. For example, there are dstip or srcip that define the IP addresses of the destination and source. Further parameters can be regex expression, source port, URL, protocol, hostname, and many others [44]. An important internal parameter is the if_sid because it defines that this specific rule can be triggered only in case the specified rule id has been triggered before [44].

The second rule, that is originally provided by Wazuh has been slightly changed in order to detect CVE-2021-41773 PoC (therefore the detection of /bin/sh. /bin/passwd and /etc/passwd has been added).

Code listing 4.6 Edited predefined rule ID 31104

```
<rule id="31104" level="10" overwrite="yes">
    <if_sid>31100</if_sid>

    <!-- Attempt to do directory transversal, simple sql injections,
      -  or access to the etc or bin directory (unix). -->
    <url>/bin/sh|/bin/passwd|/etc/passwd|%027|%00|%01|%7f|%2E%2E|%0A|%0D
        |../..|..\..|echo;|</url>
    <url>cmd.exe|root.exe|_mem_bin|msadc|/winnt/|/boot.ini|</url>
    <url>/x90/|default.ida|/sumthin|nsiislog.dll|chmod%|wget%|cd%20|</
       url>
    <url>exec%20|../..//|%5C../%5C|./././|2e%2e%5c%2e|\x5C\x5C</url>
    <description>Common web attack.</description>
    <mitre>
      <id>T1055</id>
      <id>T1083</id>
      <id>T1190</id>
    </mitre>
    <group>attack,pci_dss_6.5,pci_dss_11.4,pci_dss_6.5.1,gdpr_IV_35.7.d,
       nist_800_53_SA.11,nist_800_53_SI.4,tsc_CC6.6,tsc_CC7.1,tsc_CC8.1,
       tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
  </rule>
```

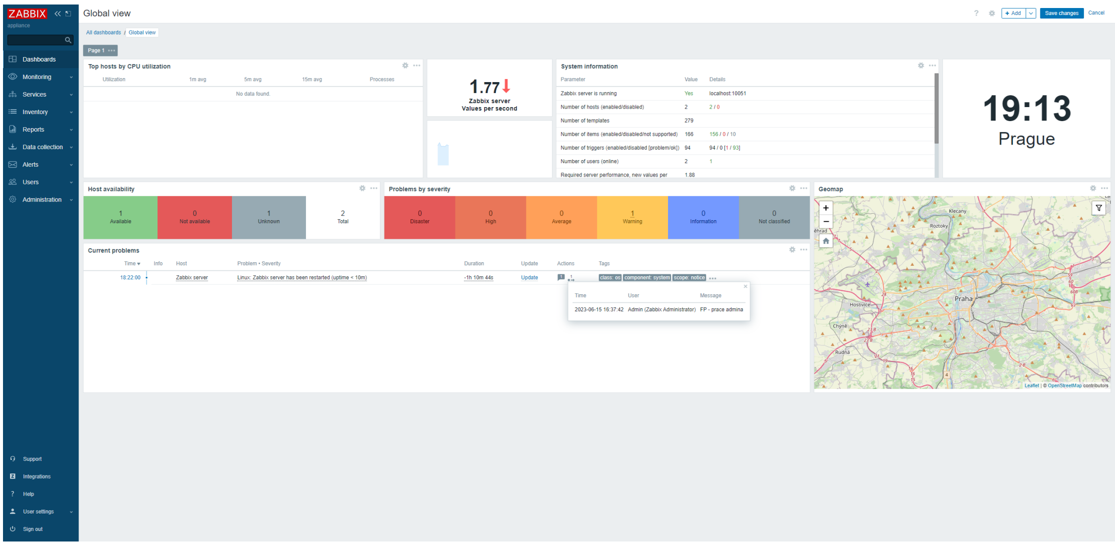
4.10 Network operations monitoring

A typical SOC should also monitor the network for any anomalies (traffic peaks, nonresponding hosts like web servers, DNS, firewalls, and the devices used by the SOC team itself). For such purposes, specialized network monitoring solutions are used. These systems monitor many key parameters of the host systems (like CPU utilization, disk I/O operations, RAM usage, available and used disk space, interface communication load, etc.). There are several solutions available on the market: Zabbix, SolarWinds Network Performance Monitor, Atera, ManageEngine OpManager, Paessler PRTG Network Monitor, or Nagios (and several others). However, only Zabbix is free to use (other aforementioned provide only free trials) so it was chosen to be deployed in our laboratory environment. Zabbix solution has been chosen to be deployed as a virtual machine on a laptop. In a real environment, it should run either as a virtual machine on a server, however, it should be a server, where the operations of Zabbix will not get disturbed by other virtual machines or it will be alone (we need to ensure the monitoring of the network is uninterrupted under all conditions).

4.10.1 Central component deployment

There are many options of server packages for installation (Hyper-V prepared Virtual machines, OVF, and iso). Several of them have been tested for deployment, however first properly working version was OVF virtual machine, which was imported into a VirtualBox solution. In order to function correctly, it needed a bridged net connection (in order to be visible from other devices in the network segment of the laptop). Therefore, the deployment itself was once again simple and now we need to configure the Zabbix agents.

Figure 4.4 Zabbix dashboard with an alert about the restart of the Zabbix server itself



Top hosts by CPU utilization

Utilization	1m avg	5m avg	15m avg	Processes
No data found.				

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	2	2 / 0
Number of templates	279	
Number of items (enabled/disabled/not supported)	166	156 / 0 / 10
Number of triggers (enabled/disabled [problem/ok])	94	94 / 0 [1 / 93]
Number of users (online)	2	1
Required server performance, new values per	1.88	

Host availability

1 Available	0 Not available	1 Unknown	2 Total

Problems by severity

0 Disaster	0 High	0 Average	1 Warning	0 Information	0 Not classified

Current problems

Time	Info	Host	Problem • Severity	Duration	Update	Actions	Tags
18:22:00		Zabbix server	Linux: Zabbix server has been restarted (uptime < 10m)	-1h 10m 44s	Update	1	class: os component: system scope: notice

Time	User	Message
2023-06-15 16:37:42	Admin (Zabbix Administrator)	FP - prace admina

4.10.2 Agents deployment

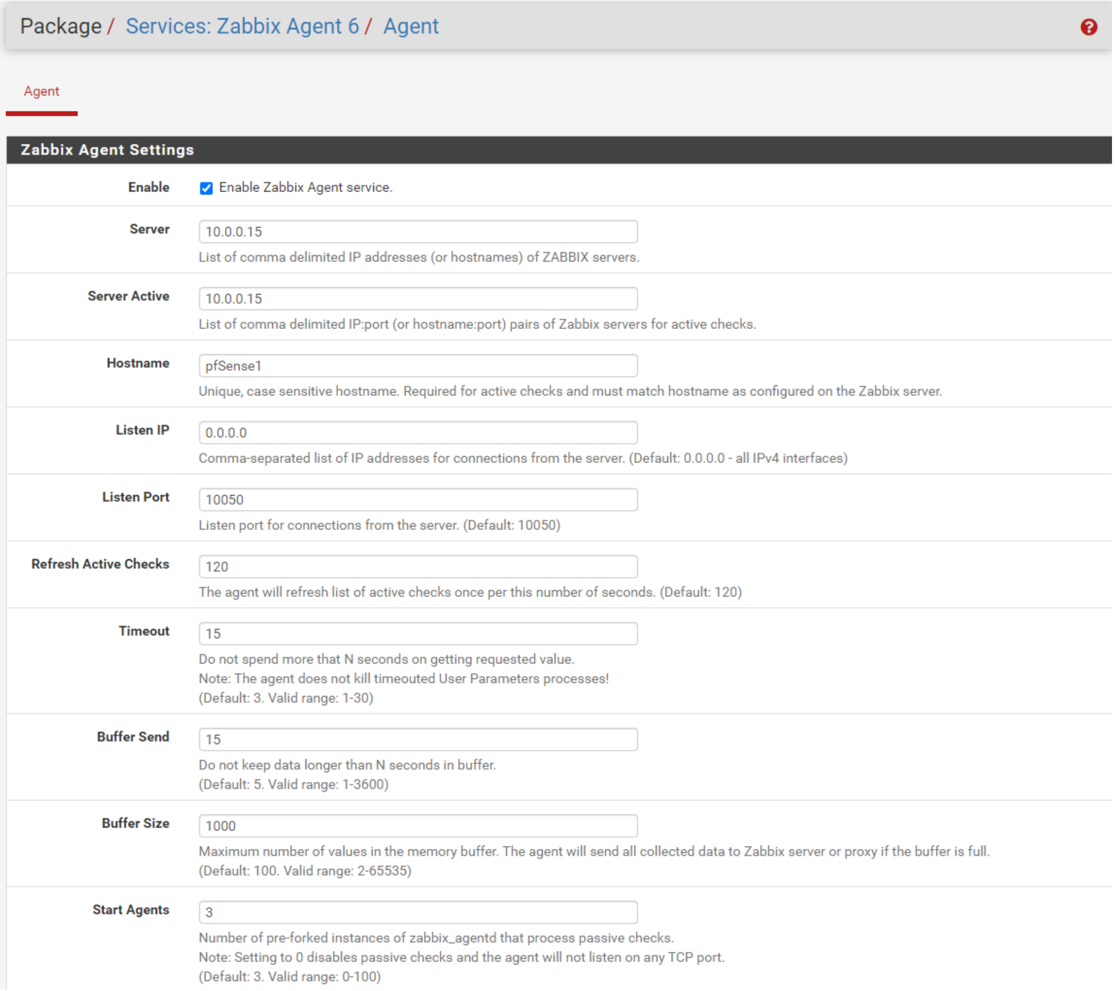
There are several ways available for monitoring with Zabbix like the SNMP (Simple Network Management Protocol), however, Zabbix provides reliable agents for a large variety of devices and operating systems. There were two devices that were chosen for the Zabbix monitoring: the pfSense router and the Windows machine.

4.10.2.1 Deployment on the pfSense

Installation to the router needs to be through the package manager function of the pfSense router. There are several versions available, we choose the version 6.4. The setting of the agent is not very complicated, because we only need to set several parameters. Namely the Zabbix server address (or an address of a Zabbix proxy, that would collect the data on behalf of the main server), the server active address that is used for checking activity, the hostname of this monitored device, listen IP address that determines, which IP addresses Zabbix agent will be listening on (0.0.0.0 in case of all interfaces), listening port, parameters of transmission and timeout. Also, we need to add this host to the list of hosts on the central server. There we need to use the function Create Host in the Inventory - Hosts section. We need to use the exact same agent's device hostname, choose the template for monitoring parameters (in this case we have used the FreeBSD one), and choose the device group and IP address of an interface the Zabbix agent can be contacted on.

After the initial configuration was done, we had to add a firewall rule for traffic allowance on port 10050 on the LAN interface of the device (in our case allow IPv4 TCP traffic to a destination called ”This Firewall” on port 10050 on the interface LAN). After a while, the Zabbix agent connects to the central server and starts to regularly send the parameters of the device operation.

Figure 4.5 Configuration of the Zabbix agent on pfSense



Package / Services: Zabbix Agent 6 / Agent

Agent

Zabbix Agent Settings

Enable ☑ Enable Zabbix Agent service.

Server 10.0.0.15
List of comma delimited IP addresses (or hostnames) of ZABBIX servers.

Server Active 10.0.0.15
List of comma delimited IP:port (or hostname:port) pairs of Zabbix servers for active checks.

Hostname pfSense1
Unique, case sensitive hostname. Required for active checks and must match hostname as configured on the Zabbix server.

Listen IP 0.0.0.0
Comma-separated list of IP addresses for connections from the server. (Default: 0.0.0.0 - all IPv4 interfaces)

Listen Port 10050
Listen port for connections from the server. (Default: 10050)

Refresh Active Checks 120
The agent will refresh list of active checks once per this number of seconds. (Default: 120)

Timeout 15
Do not spend more that N seconds on getting requested value.
Note: The agent does not kill timeouted User Parameters processes!
(Default: 3. Valid range: 1-30)

Buffer Send 15
Do not keep data longer than N seconds in buffer.
(Default: 5. Valid range: 1-3600)

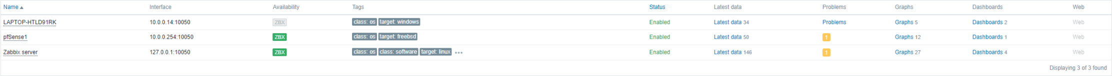
Buffer Size 1000
Maximum number of values in the memory buffer. The agent will send all collected data to Zabbix server or proxy if the buffer is full.
(Default: 100. Valid range: 2-65535)

Start Agents 3
Number of pre-forked instances of zabbix_agentd that process passive checks.
Note: Setting to 0 disables passive checks and the agent will not listen on any TCP port.
(Default: 3. Valid range: 0-100)

4.10.2.2 Deployment on the Microsoft Windows

The deployment of the Windows Zabbix agent is also straightforward. We need to download the proper agent version for our instruction set architecture and OS (in this case AMD64 Windows version) and install it. Once again, we need to set the parameters of the operation (therefore the IP address of the server, hostname of the device, listening port, and IP address for active check). As mentioned before, we need to input similar parameters on the central server (the hostname, group name, template, and IP address). However, in the case of Windows machines, we also need to allow traffic to port 10050 required for Zabbix communication. Therefore, we set the rule inside the Windows firewall (Incoming rules - New rule - Port).

Figure 4.6 Zabbix now shows all three monitored systems



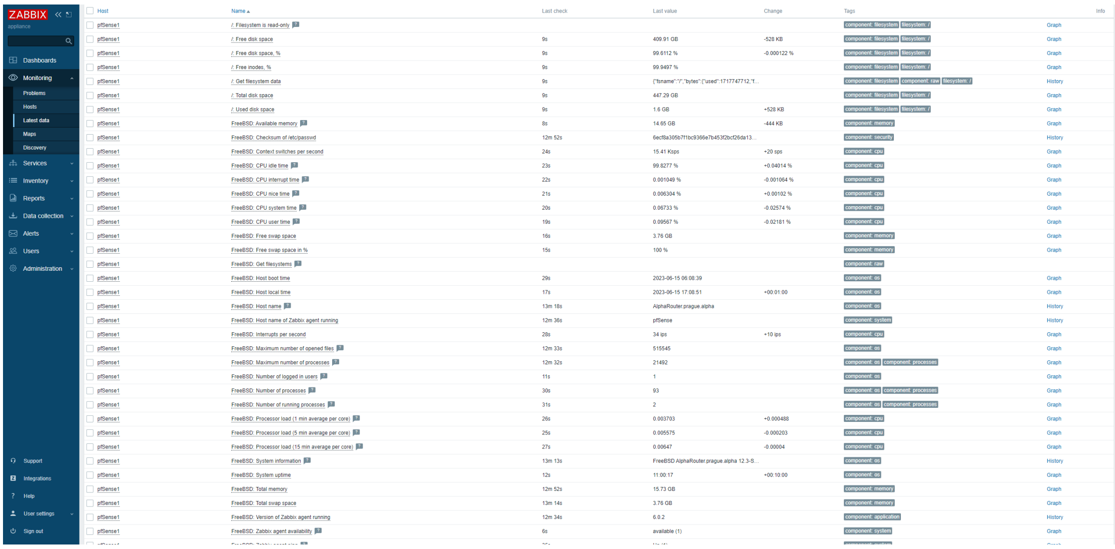
Name ▲	Interface	Availability	Tags	Status	Latest data	Problems	Graphs	Dashboards	Web
LAPTOP-HTLD91RK	10.0.0.14:10050	ZBX	class: os target: windows	Enabled	Latest data 34	Problems	Graphs 5	Dashboards 2	Web
pfSense1	10.0.0.254:10050	ZBX	class: os target: freebsd	Enabled	Latest data 50	1	Graphs 12	Dashboards 1	Web
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software target: linux •••	Enabled	Latest data 146	1	Graphs 27	Dashboards 4	Web

Displaying 3 of 3 found

4.10.3 Traffic flow monitoring

While Zabbix can monitor loads of parameters across the system (from file system to the usage of hardware), we can also monitor network traffic.

Figure 4.7 System data of the pfSense router shown in the Zabbix server



□ Host	Name ▲	Last check	Last value	Change	Tags	Info
□ pfSense1	/: Filesystem is read-only ?				component: filesystem filesystem: /	Graph
□ pfSense1	/: Free disk space	9s	409.91 GB	-528 KB	component: filesystem filesystem: /	Graph
□ pfSense1	/: Free disk space, %	9s	99.6112 %	-0.000122 %	component: filesystem filesystem: /	Graph
□ pfSense1	/: Free inodes, %	9s	99.9497 %		component: filesystem filesystem: /	Graph
□ pfSense1	/: Get filesystem data	9s	{"fsname":"/","bytes":{"used":1717747712,"f...		component: filesystem component: raw filesystem: /	History
□ pfSense1	/: Total disk space	9s	447.29 GB		component: filesystem filesystem: /	Graph
□ pfSense1	/: Used disk space	9s	1.6 GB	+528 KB	component: filesystem filesystem: /	Graph
□ pfSense1	FreeBSD: Available memory ?	8s	14.65 GB	-444 KB	component: memory	Graph
□ pfSense1	FreeBSD: Checksum of /etc/passwd	12m 52s	6ecf8a305b7f1bc9366e7b453f2bcf26da13...		component: security	History
□ pfSense1	FreeBSD: Context switches per second	24s	15.41 Ksps	+20 sps	component: cpu	Graph
□ pfSense1	FreeBSD: CPU idle time ?	23s	99.8277 %	+0.04014 %	component: cpu	Graph
□ pfSense1	FreeBSD: CPU interrupt time ?	22s	0.001049 %	-0.001064 %	component: cpu	Graph
□ pfSense1	FreeBSD: CPU nice time ?	21s	0.006304 %	+0.00102 %	component: cpu	Graph
□ pfSense1	FreeBSD: CPU system time ?	20s	0.06733 %	-0.02574 %	component: cpu	Graph
□ pfSense1	FreeBSD: CPU user time ?	19s	0.09567 %	-0.02181 %	component: cpu	Graph
□ pfSense1	FreeBSD: Free swap space	16s	3.76 GB		component: memory	Graph
□ pfSense1	FreeBSD: Free swap space in %	15s	100 %		component: memory	Graph
□ pfSense1	FreeBSD: Get filesystems ?				component: raw	
□ pfSense1	FreeBSD: Host boot time	29s	2023-06-15 06:08:39		component: os	Graph
□ pfSense1	FreeBSD: Host local time	17s	2023-06-15 17:08:51	+00:01:00	component: os	Graph
□ pfSense1	FreeBSD: Host name ?	13m 18s	AlphaRouter.prague.alpha		component: os	History
□ pfSense1	FreeBSD: Host name of Zabbix agent running	12m 36s	pfSense		component: system	History
□ pfSense1	FreeBSD: Interrupts per second	28s	34 ips	+10 ips	component: cpu	Graph
□ pfSense1	FreeBSD: Maximum number of opened files ?	12m 33s	515545		component: os	Graph
□ pfSense1	FreeBSD: Maximum number of processes ?	12m 32s	21492		component: os component: processes	Graph
□ pfSense1	FreeBSD: Number of logged in users ?	11s	1		component: os	Graph
□ pfSense1	FreeBSD: Number of processes ?	30s	93		component: os component: processes	Graph
□ pfSense1	FreeBSD: Number of running processes ?	31s	2		component: os component: processes	Graph
□ pfSense1	FreeBSD: Processor load (1 min average per core) ?	26s	0.003703	+0.000488	component: cpu	Graph
□ pfSense1	FreeBSD: Processor load (5 min average per core) ?	25s	0.005575	-0.000203	component: cpu	Graph
□ pfSense1	FreeBSD: Processor load (15 min average per core) ?	27s	0.00647	-0.00004	component: cpu	Graph
□ pfSense1	FreeBSD: System information ?	13m 13s	FreeBSD AlphaRouter.prague.alpha 12.3-S...		component: os	History
□ pfSense1	FreeBSD: System uptime	12s	11:00:17	+00:10:00	component: os	Graph
□ pfSense1	FreeBSD: Total memory	12m 52s	15.73 GB		component: memory	Graph
□ pfSense1	FreeBSD: Total swap space	13m 14s	3.76 GB		component: memory	Graph
□ pfSense1	FreeBSD: Version of Zabbix agent running	12m 34s	6.0.2		component: application	History
□ pfSense1	FreeBSD: Zabbix agent availability ?	6s	available (1)		component: system	Graph

This would come in handy for the monitoring of the load on the router itself and therefore this section will show, how we can monitor the network interface, how we create a special dashboard for the monitoring, and how we set triggers that would alert us about higher than standard network load.

4.10.3.1 Dashboard creation

For the creation of a custom dashboard (for example for a specific network port on a router), we need to go to the section Dashboards and use the function Create a dashboard. Then we use the Add widget option, set the type Graph, select the correct item pattern (Interface igc0 traffic in our case, which is the LAN one), and click Add. With that, we have created a dashboard that could be used for monitoring outgoing traffic. With such a graph, we could see for example large file downloads or the exhaustion of the capacity of the internet connection.

4.10.3.2 Trigger setting

In order to alert analysts about anomalies in the operation, we can use triggers. Such triggers can monitor any parameters that are being sent by the agents. In order to demonstrate how a specific trigger can be set, we have set a trigger on the igc0 interface (but with a trigger threshold so low that it could not be used in real-life operations). Therefore we needed to configure a new trigger in the section Data Collection - Hosts, where we click on triggers on the row of the selected host we are creating the trigger for. There, we create a new trigger with the function Create a trigger, where we had to set at least name and expression (the evaluation expression is created with the help of an assistant). The expression has 4 key parts: logical operator like minimum, last, or average, monitored parameter of the specific device, time frame for evaluation of data (like the average in the last n minutes), and the threshold value [45]. Therefore, we configured a trigger for a Warning level notification named Outgoing traffic is too high with the expression max(pfSense1/net.if.in[igc0], 1s) > 1000.

4.11 Operation of the SOC team on the provided infrastructure

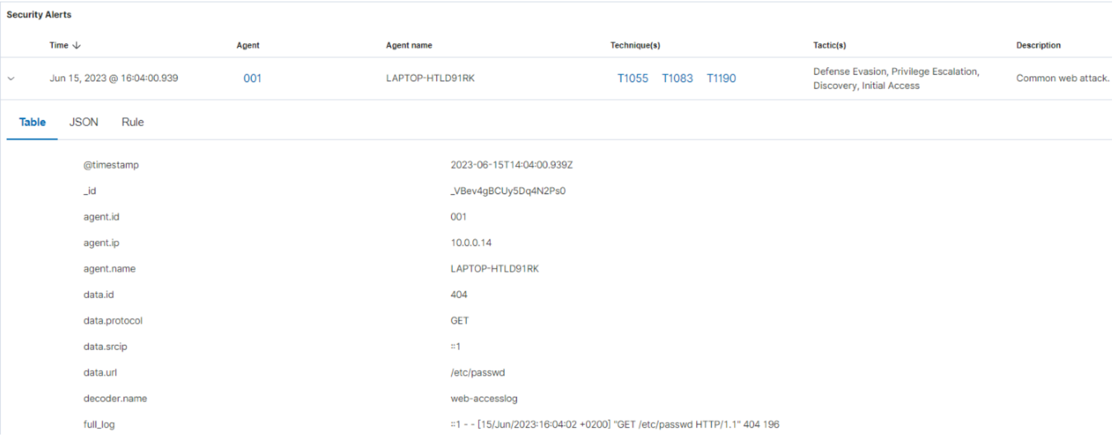
Now, we have prepared all the systems for monitoring the activity on the laboratory network. We have the SIEM and log management systems collecting logs and detecting security anomalies, but we also have network operation monitoring, that helps us monitor hosts and key parameters of the systems that are deployed on the network. This also allows us some level of security monitoring (mostly volumetric anomalies on the network). Therefore, if a real SOC team would operate in this environment, he would have to monitor Zabbix and the Wazuh system at the same time for good protection.

4.11.1 Simulated attacks on various network or endpoint devices

In order to demonstrate, how the infrastructure built in the previous parts would work in situations when some suspicious or anomalous activity has been identified, we have tried several different types of activities against the systems. The first one is an attack against a web server with the code CVE-2021-41773. The second one illustrates how an unsuccessful brute force attack on one of the machines inside the network would appear in the monitoring systems. The third activity is the detection of an attempt to connect to an IP address that has been previously marked by an unknown someone in the lab network as malicious.

4.11.1.1 Apache path traversal vulnerability

Figure 4.8 Alert showing that an attack to the Apache server has been detected



Security Alerts

	Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description
⌄	Jun 15, 2023 @ 16:04:00.939	001	LAPTOP-HTLD91RK	T1055 T1083 T1190	Defense Evasion, Privilege Escalation, Discovery, Initial Access	Common web attack.

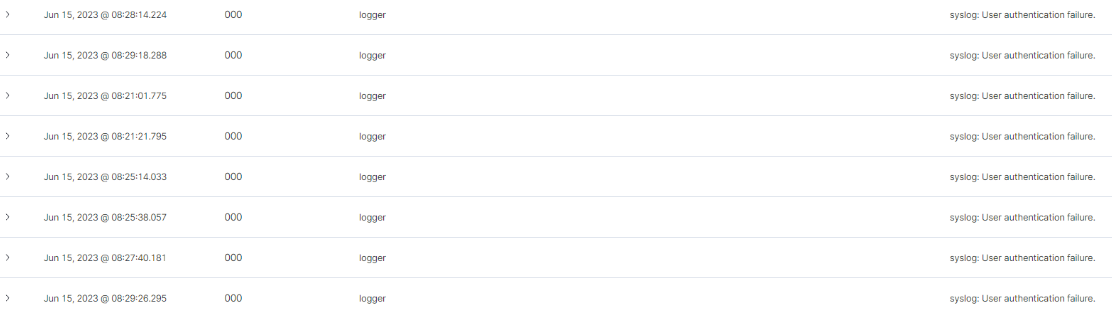
@timestamp	2023-06-15T14:04:00.939Z
_id	_VBev4gBCUy5Dq4N2Ps0
agent.id	001
agent.ip	10.0.0.14
agent.name	LAPTOP-HTLD91RK
data.id	404
data.protocol	GET
data.srcip	::1
data.url	/etc/passwd
decoder.name	web-accesslog
full_log	::1 - - [15/Jun/2023:16:04:02 +0200] "GET /etc/passwd HTTP/1.1" 404 196

This attack is based on a path traversal inside a URL string sent to the Apache server. To be successful, a specific version would have to be installed, however, for the purposes of this thesis, no successful exploitation is needed. As previously mentioned in the section about Wazuh rules, a specific rule inside the SIEM needed to be edited in order to successfully detect this kind of path traversal. The second thing that was needed for detection was the log collection of the Apache web server access log, which is described in the section *Endpoint log agents deployment*. After all this, the only thing we needed was to use the PoC url against the web server. This try was done from the machine, where the Apache server was running (the detection mechanisms work exactly the same). When the access attempt happened, the web server wrote the log to the file, the Wazuh agent sent this log to the server, and the server decoded the log and recognized, that it matched our updated rule. Therefore, an alert has been sent to our dashboard.

Under the real circumstances, we could consider blocking the attacking address on the perimeter firewall (something, that is actually described in the third simulated attack), however in this case the only thing we wanted was to successfully recognize the attack because we already know that the web server is not vulnerable.

4.11.1.2 Endpoint account password brute force

■ **Figure 4.9** Alerts that show failed logins

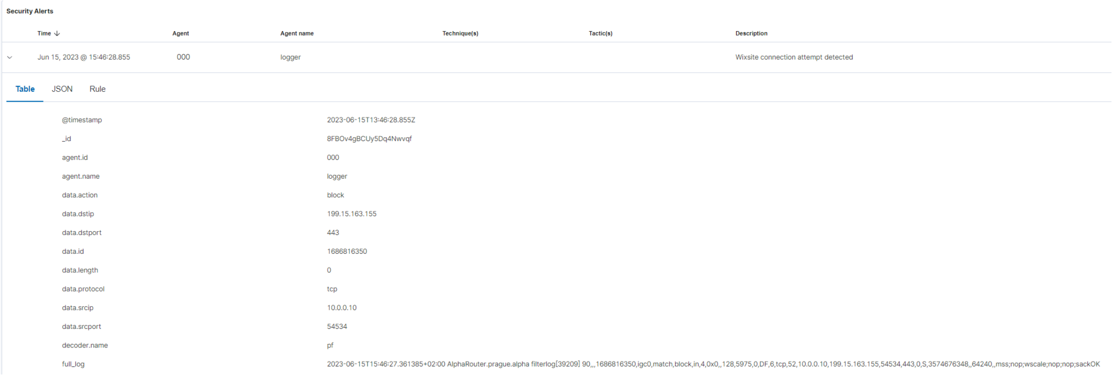


›	Jun 15, 2023 @ 08:28:14.224	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:29:18.288	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:21:01.775	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:21:21.795	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:25:14.033	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:25:38.057	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:27:40.181	000	logger	syslog: User authentication failure.
›	Jun 15, 2023 @ 08:29:26.295	000	logger	syslog: User authentication failure.

In this simulated activity, the alleged attacker tried to access the account on the Debian server by brute forcing a password. This activity was immediately recorded in the log files, sent to the Wazuh server, and recognized as a series of User authentication failures. In this case, the user password has been reset to avoid a successful brute force attack, even though no successful login has been detected by Wazuh any time after the brute force attempts.

4.11.1.3 Contact to the suspicious IP address from internal network

■ **Figure 4.10** Alert that shows that one of the devices tried to access blocked IP address



Security Alerts

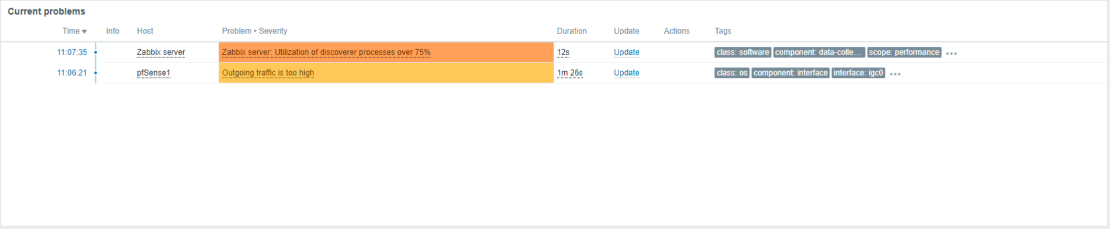
	Time ↓	Agent	Agent name	Technique(s)	Tactic(s)	Description
⌄	Jun 15, 2023 @ 15:46:28.855	000	logger			Wixsite connection attempt detected

@timestamp	2023-06-15T13:46:28.855Z
_id	8FBOv4gBCUy5Dq4Nwvqf
agent.id	000
agent.name	logger
data.action	block
data.dstip	199.15.163.155
data.dstport	443
data.id	1686816350
data.length	0
data.protocol	tcp
data.srcip	10.0.0.10
data.srcport	54534
decoder.name	pf
full_log	2023-06-15T15:46:27.361385+02:00 AlphaRouter.prague.alpha filterlog[39209] 90,,,1686816350,igc0,match,block,in,4,0x0,,128,5975,0,DF,6,tcp,52,10.0.0.10,199.15.163.155,54534,443,0,S,3574676348,,64240,,mss;nop;wscale;nop;nop;sackOK

In this hypothetical case, IP addresses 199.15.163.145 and 199.15.163.155 have been historically identified as potentially malicious (in reality these IP addresses belong to the company Wixsite, which hosts many pages and therefore this blocking would be most likely excessive). We have blocked these aforementioned addresses on the perimeter firewall and we have set the detection mechanisms in order to detect any attempts to access any of those 2 IP addresses (a detailed description of the relevant Wazuh rule is provided in the section *Wazuh rules*). Such an attempt has been made from one of the monitored computers and in this case, the hypothetical user would have been briefed about this, his PC would get deep antivirus inspection and the reason for the access attempt would have to be discovered in order to prevent any such activity in the future.

4.11.1.4 Traffic volume detection

Figure 4.11 Zabbix showing that the Outgoing traffic on the interface igc0 of the pfSense router is too high



Current problems

Time ▾	Info	Host	Problem • Severity	Duration	Update	Actions	Tags
11:07:35		Zabbix server	Zabbix server: Utilization of discoverer processes over 75%	12s	Update		class: software component: data-colle... scope: performance ...
11:06:21		pfSense1	Outgoing traffic is too high	1m 26s	Update		class: os component: interface interface: igc0 ...

As we have also configured Zabbix monitoring and one trigger for the traffic volume on the interface igc0, we will try to surpass the threshold in order to demonstrate that the monitoring is working. We have achieved that thanks to the speed test, which generated high enough traffic to trigger the warning alert in the Zabbix current problems section. This issue has been checked by the hypothetical operator and it has been discovered that it was a short-time excessive load, that was already resolved by the time of control (the speed test ended). Therefore, he added a comment and closed the issue. However, if such overload would continue, we would need to start to explore, which specific device is responsible for the load.

4.12 Options for the further development of the lab environment

There are many options for further development of the lab. One of them is deployment on a more used environment, where we could get new detections. Another option is to add new rules (even though the Wazuh SIEM contains many by default), or to connect devices that would require adding decoders (which was not necessary for the purpose of this thesis). Another option is a comparison with other available SIEM and network operations monitoring solutions. To summarize, there are plenty of options for further development of this environment.

4.13 Evaluation of operation

Some problems with the Wazuh system have been discovered, as it is described in the section *Limitations of the deployed SIEM solution*. However, the deployment of the Wazuh system was mostly straightforward and not too complicated. The use of the SIEM is easy, the user interface is well arranged and it could be used by a SOC team on a long-term basis (considering however the previously mentioned limitations).

The configuration of the network monitoring system was only complicated because of installation packages and virtual machines that for unknown reasons would not work. In the end, however, we have been able to start a Zabbix server and configure monitoring for two devices. After the initial configuration, the server and the agent needed some time in order to start successful communication (about 30 minutes to one hour), however afterward the operations were uninterrupted and reliable.

Therefore, the process of deployment of technologies has been lengthy, but not overly complicated and we have managed to build a functional SOC infrastructure (even though for larger networks additional components for monitoring would be appropriate).

4.14 Conclusion of SOC deployment

The goals of this chapter were to deploy, set up, and evaluate the security surveillance (monitoring) technologies. We have deployed and configured the Wazuh system with its three main components, and three agents on endpoints, we have successfully set new log file locations and configured rules for the detection of some new activities. Moreover, we have configured network operations monitoring and created new monitoring tools like a dashboard and an alert trigger for a specific event. We have also pointed out some limitations of the Wazuh solution, but also its capabilities. The operations of the whole complex of the monitoring systems have finally been evaluated.

Conclusion

The main goal of this thesis was to introduce the reader to the problem of security monitoring and the principles of functioning of the Security operation centers, which are responsible for the surveillance of various systems. This goal included a description of both the personal part of such teams, but also the infrastructure part that is needed for good protection of the company or organization‘s network. The second goal of this thesis was for the practical part to design and build small-scale security monitoring on top of the laboratory environment, which includes typical systems we could find in a normal company network topology.

These goals were successfully fulfilled because the result of the theoretical part of this thesis is the description of all the types of technologies that could be monitored in any real-life company environment. Another important result of this thesis is the description of the procedure for the creation of the infrastructure for such a security operation center, which is provided in the practical part of this work.

The topic of this thesis could be further expanded to cover securing a real-life large network topology, however, the results of such would be very similar to the ones created by this work. In contrast, deploying these technologies in real networks might pose additional challenges in cooperation with administrators of the affected network. Additionally, this topic might be quite sensitive to the companies, which might not want to have their internal security principles and infrastructure details discussed in any publicly available work.

Furthermore, more focus could be added to the topic of log parsing and processing, however, this topic would become highly vendor-specific, which was intended to be mostly avoided in this work.

Bibliography

1. *What is a Security Operations Center (SOC)?* [Online]. IBM, [n.d.] [visited on 2023-02-28]. Available from: https://www.ibm.com/topics/security-operations-center.
2. KRISHNAN, Ashwin. *5 key enterprise SOC team roles and responsibilities — TechTarget* [online]. TechTarget, [n.d.] [visited on 2023-03-15]. Available from: https://www.techtarget.com/searchsecurity/tip/5-key-enterprise-SOC-roles-and-responsibilities.
3. WILLIAMS, Kayla. *SANS Cyber Security Certifications & Research* [online]. SANS, [n.d.] [visited on 2023-03-21]. Available from: https://www.sans.org/blog/it-s-time-to-break-the-soc-analyst-burnout-cycle/.
4. O'DRISCOLL, Aimee. *25+ Cyber Security Vulnerability Statistics and Facts of 2023* [online]. Comparitech, [n.d.] [visited on 2023-03-21]. Available from: https://www.comparitech.com/blog/information-security/cybersecurity-vulnerability-statistics/.
5. LAKE, Sydney. *The cybersecurity industry is short 3.4 million workers—that's good news for cyber wages — Fortune* [online]. Fortune, [n.d.] [visited on 2023-03-20]. Available from: https://fortune.com/education/articles/the-cybersecurity-industry-is-short-3-4-million-workers-thats-good-news-for-cyber-wages/.
6. *Endpoint detection and response (EDR) solutions reviews 2023: Gartner Peer insights* [online]. Gartner, [n.d.] [visited on 2023-03-23]. Available from: https://www.gartner.com/reviews/market/endpoint-detection-and-response-solutions.
7. AARNESS, Anne. *What is EDR? Endpoint Detection & Response defined* [online]. CrowdStrike, [n.d.] [visited on 2023-03-23]. Available from: https://www.crowdstrike.com/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/.
8. *What is Email Security? - Check Point Software* [online]. Check Point Software, [n.d.] [visited on 2023-03-24]. Available from: https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-email-security/.
9. *What is Intrusion Prevention System? — VMware Glossary* [online]. VMware, [n.d.] [visited on 2023-03-24]. Available from: https://www.vmware.com/topics/glossary/content/intrusion-prevention-system.
10. *Understanding the 5 Types of Intrusion Detection Systems — Helixstorm* [online]. Helixstorm, [n.d.] [visited on 2023-03-27]. Available from: https://www.helixstorm.com/blog/types-of-intrusion-detection-systems/.
11. *What is The Difference Between Stateful & Stateless Firewall?* [Online]. Fortinet, [n.d.] [visited on 2023-03-27]. Available from: https://www.fortinet.com/resources/cyberglossary/stateful-vs-stateless-firewall.

12. *Stateful vs. Stateless Firewall - Check Point Software* [online]. Check Point Software, [n.d.] [visited on 2023-03-27]. Available from: https://www.checkpoint.com/cyber-hub/network-security/what-is-firewall/what-is-a-stateful-firewall/stateful_vs_stateless_firewall/.

13. *What Is a Firewall? - Cisco* [online]. Cisco, [n.d.] [visited on 2023-03-27]. Available from: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html.

14. VELIMIROVIC, Andreja. *The 8 Types of Firewalls Explained* [online]. phoenixNAP, 2022 [visited on 2023-04-19]. Available from: https://phoenixnap.com/blog/types-of-firewalls.

15. *What is a Secure Web Gateway (SWG)? - Check Point Software* [online]. Check Point Software, [n.d.] [visited on 2023-06-14]. Available from: https://www.checkpoint.com/cyber-hub/network-security/what-is-secure-web-gateway/.

16. *What Is a Next-Generation Firewall (NGFW)? - Cisco* [online]. Cisco, [n.d.] [visited on 2023-03-27]. Available from: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-next-generation-firewall.html.

17. *What is a honeypot? How honeypots help security* [online]. Kaspersky, [n.d.] [visited on 2023-03-27]. Available from: https://www.kaspersky.com/resource-center/threats/what-is-a-honeypot.

18. LIVSHITZ, Igor. *Low, Medium and High Interaction Honeypot Security — Akamai* [online]. Akamai, [n.d.] [visited on 2023-04-13]. Available from: https://www.akamai.com/blog/security/high-interaction-honeypot-versus-low-interaction-honeypot-comparison.

19. *What is Windows Auditing? Read the Definition in our. . . — BeyondTrust* [online]. BeyondTrust, [n.d.]. Available also from: https://www.beyondtrust.com/resources/glossary/windows-auditing.

20. S, Visakh. *Server security monitoring - Why do it, and what to monitor* [online]. bobcares, 2018 [visited on 2023-04-19]. Available from: https://bobcares.com/blog/server-security-monitoring/.

21. DESHMUKH, Mayank. *Apache HTTP Server Path Traversal & Remote Code Execution (CVE-2021-41773 & CVE-2021-42013) — Qualys Security Blog* [online]. Qualys Security Blog, [n.d.] [visited on 2023-04-13]. Available from: https://blog.qualys.com/vulnerabilities-threat-research/2021/10/27/apache-http-server-path-traversal-remote-code-execution-cve-2021-41773-cve-2021-42013.

22. *Sensitive Data of 65,000+ Entities in 111 Countries Leaked due to a Single Misconfigured Data Bucket* [online]. SOCRadar, 2022 [visited on 2023-04-19]. Available from: https://socradar.io/sensitive-data-of-65000-entities-in-111-countries-leaked-due-to-a-single-misconfigured-data-bucket/.

23. *What is a Cloud Access Security Broker (CASB)? - Skyhigh Security* [online]. Skyhigh Security, [n.d.] [visited on 2023-04-19]. Available from: https://www.skyhighsecurity.com/en-us/cybersecurity-defined/what-is-a-casb.html.

24. *What is Agentless Monitoring? - IT Glossary — SolarWinds* [online]. SolarWinds, [n.d.] [visited on 2023-04-19]. Available from: https://www.solarwinds.com/resources/it-glossary/agentless-monitoring.

25. *Common Event Format (CEF) Logging Support in the Application Firewall* [online]. Citrix, 2016 [visited on 2023-04-19]. Available from: https://support.citrix.com/article/CTX136146/common-event-format-cef-logging-support-in-the-application-firewall.

26. *What Is Log Retention? — LogicMonitor* [online]. LogicMonitor, [n.d.] [visited on 2023-06-28]. Available from: https://www.logicmonitor.com/blog/what-is-log-retention.
27. SHARIF, Arfan. *What is Log Management? 4 Best Practices & More - CrowdStrike* [online]. CrowdStrike, [n.d.] [visited on 2023-04-25]. Available from: https://www.crowdstrike.com/cybersecurity-101/observability/log-management/.
28. SHARIF, Arfan. *Log Parsing: What Is It and How Does It Work? — CrowdStrike* [online]. CrowdStrike, [n.d.] [visited on 2023-04-20]. Available from: https://www.crowdstrike.com/cybersecurity-101/observability/log-parsing/.
29. *What is Security Information and Event Management (SIEM)? — IBM* [online]. IBM, [n.d.] [visited on 2023-04-21]. Available from: https://www.ibm.com/topics/siem.
30. SHARIF, Arfan. *SIEM vs Log Management: What's the Difference? - CrowdStrike* [online]. CrowdStrike, [n.d.] [visited on 2023-04-21]. Available from: https://www.crowdstrike.com/cybersecurity-101/observability/siem-vs-log-management.
31. *What Is SOAR? Security Orchestration, Automation, and Response — Fortinet* [online]. Fortinet, [n.d.] [visited on 2023-04-25]. Available from: https://www.fortinet.com/resources/cyberglossary/what-is-soar.
32. *What is SOAR (Security Orchestration, Automation and Response)? — Definition from TechTarget* [online]. TechTarget, [n.d.] [visited on 2023-04-26]. Available from: https://www.techtarget.com/searchsecurity/definition/SOAR.
33. *pfSense vs. OPNsense in 2023 - WunderTech* [online]. WunderTech, [n.d.] [visited on 2023-04-20]. Available from: https://www.wundertech.net/pfsense-vs-opnsense/.
34. *Packages — Package List — pfSense Documentation* [online]. Netgate, [n.d.] [visited on 2023-04-20]. Available from: https://docs.netgate.com/pfsense/en/latest/packages/list.html.
35. *Official pfSense Hardware, Appliances, and Security Gateways* [online]. pfsense, [n.d.] [visited on 2023-06-16]. Available from: https://www.pfsense.org/products/.
36. *Pricing - IBM Security QRadar SIEM — IBM* [online]. IBM, [n.d.] [visited on 2023-06-16]. Available from: https://www.ibm.com/products/qradar-siem/pricing.
37. *Wazuh indexer - Installation guide · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-14]. Available from: https://documentation.wazuh.com/current/installation-guide/wazuh-indexer/index.html.
38. *Wazuh server - Installation guide · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-14]. Available from: https://documentation.wazuh.com/current/installation-guide/wazuh-server/index.html.
39. *Wazuh dashboard - Components · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-16]. Available from: https://documentation.wazuh.com/current/getting-started/components/wazuh-dashboard.html.
40. *Installation alternatives · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-21]. Available from: https://documentation.wazuh.com/current/deployment-options/index.html.
41. *Monitoring pfSense with Wazuh* [online]. 0xBEN, [n.d.] [visited on 2023-06-18]. Available from: https://benheater.com/integrating-pfsense-with-wazuh/.
42. *How it works - Log data collection · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-19]. Available from: https://documentation.wazuh.com/current/user-manual/capabilities/log-data-collection/how-it-works.html.
43. *Components - Getting started with Wazuh · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-19]. Available from: https://documentation.wazuh.com/current/getting-started/components/index.html.

44. *Rules Syntax - Ruleset XML syntax · Wazuh documentation* [online]. Wazuh, [n.d.] [visited on 2023-06-19]. Available from: `https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/rules.html`.

45. *2 Trigger expression* [online]. Zabbix, [n.d.] [visited on 2023-06-21]. Available from: `https://www.zabbix.com/documentation/current/en/manual/config/triggers/expression`.