Modeling and Analyzing Malware Behavior in Virtual Networks Using EVE-NG

Maria-Mădălina Andronache[1], Alexandru Vulpe[2], Corneliu Burileanu[3]
Research Institute "CAMPUS", National University of Science and Technology Politehnica Bucharest, Bucharest, Romania[1]
Telecommunication Department, National University of Science and Technology Politehnica Bucharest, Bucharest, Romania[2]
Speech and Dialogue Research Lab, National University of Science and Technology Politehnica Bucharest, Bucharest, Romania[3]

***Abstract*—Malicious attacks have become increasingly common in all organizations and systems. The continued evolution of such software aims to extract information from diverse systems. Therefore, the objective of this study is to introduce another approach to analyze some network attacks, within a virtual infrastructure, through multi-vendor network emulation software (Emulated Virtual Environment–Next Generation - EVE-NG). Basically, through emulated resources, the aim is to implement a complex network, which also includes a Security Information and Event Management System (SIEM), which can detect some attacks, both from the network area (carried out by malicious attackers) and through malicious files (from the public resources area), that are accidentally or intentionally downloaded by certain users. Within this environment, various scenarios can be implemented to simulate the real production environment, in order to test network vulnerabilities, but also to improve some methods for learning network attack and defense modes. In the experiments performed, the SIEM system detected most of the simulated attacks, but failed to distinguish between the displayed alarms so that the alerts could indicate the type of attack. Thus, the potential of EVE-NG for simulating and analyzing the behavior of malware is demonstrated.**

***Keywords*—EVE-NG; network attacks; network defense; SIEM; network architecture**

I. Introduction

In the current security landscape, the primary actors are continuously evolving, shaped by intensifying threats and by the new and more effective defense against the aforementioned threats. As a current way of working, in both camps, new methods based on artificial intelligence and automation of various attacks or, on the contrary, new, much more effective, defense methods, are emerging. In practice, advances are being made both in attack methods, as well as in the area of zero-trust architecture and security policies based on identification methods. In addition, key aspects are also becoming the use, on an increasingly large scale, of cloud resources and IoT technologies, which, paradoxically, increases security risks and zero-day vulnerabilities.

As networks evolve, network devices become more secure and more capable of advanced access control or observability policies with each update. In practice, most network device providers also integrate artificial intelligence or machine-learning techniques to capture various logs and detect anomalies, both in the device's own operation and at the level of a party directly involved with the device. Thus, through them, a better segmentation and a more efficient combination of network capabilities with security policies can be achieved. If a few years ago, the emphasis was on the hardware area, currently, the most intensively researched part is the software.

According to study [1], understanding how cyberattacks work and their key terminology is essential in order to know how different types of threats work and which are the most important methods to respond to some attacks. Therefore, from the attack name, one can infer commonly used strategies, frequently adopted by that malicious file.

From study [2], the key aspects that define effective cybersecurity measures are:

- Providing information that includes key characteristics about the threat, quickly.
- The ability to provide details about the main risks of the victim organization.
- Must contain clear information about the impact, which prevents the formation of a false sense of security at the organizational level.
- Requires clear and concrete information about the attack.
- Include recommended actions to prevent damage and the expansion of the attack.

As networks become larger and more interconnected, understanding how different types of attacks work and how defenses respond has become a critical area of study in cybersecurity. Experimental analysis is essential for this purpose, but conducting such research on production systems can be both expensive and dangerous, potentially exposing real networks to damage. As a result, the use of virtualized environments to simulate attacks and evaluate defenses has attracted significant attention from both researchers and educators.

Emulated Virtual Environment-Next Generation (EVE-NG) [3] is a working environment used to build various virtual network architectures, which can include equipment from several vendors. Its purpose is to learn numerous concepts and working methods, in an easier way. For an attack-type testing environment on a real network topology, as is the case in this paper, EVE-NG is an ideal environment because it facilitates access to virtualized resources while keeping the production environment intact. Thus, it works similar to a sandbox, only replicating the entire network, not just a specific part of it or a specific workstation.

EVE-NG is not formally intended for executing malicious software or simulating attack payloads. Despite this limitation, its isolated design and versatile networking features make it a practical option for controlled cybersecurity experimentation. In this paper, is explored how EVE-NG can be adapted to perform secure simulations of various network-level attack scenarios while maintaining system integrity and ethical boundaries.

Considering the information presented in study [4], emulators such as EVE-NG offer a virtualized solution closer to real working environments, by including, in the test environment, the real code images of the devices that are emulated. Through this way of working, a more accurate assessment of the impact on workstations or network devices can be achieved, by testing some features much closer to the real environment. Thus, EVE-NG is an extremely useful virtualized environment for learning the characteristics of real equipment, without disturbing their functioning in the production environment.

Malicious software is a piece of code or an entire program designed to exploit various network resources. Its main purpose is to steal data, obtain material benefits (by ransoming resources) or destroy information. The types of malicious files are diverse, but the most commonly used for normal users are viruses, adware or spyware. Within an organization's resources, the most used malicious files are ransomware or viruses, worms and trojans because the information that is wanted to be extracted is much more valuable, from a financial gain point of view, and, thus, the attack must be more sophisticated. Also, considering the latest trends in tactics, the newest ones used include combinations of multiple malware categories, changes in its own code, to make it easier to replicate, or the use of legitimate resources, already present on the victim system (living-off-the-land).

Considering the aspects presented in the paper [5], the impact of malicious software is different, depending on their intention. Thus, while some are directed to a station or a system, others are used to be spread to a part of the network or the entire network, to infect a larger number of devices. In addition, certain types of malicious files (worms, trojans, ransomware) have higher levels of aggression, even leading to the permanent interruption of some services, while adware or keyloggers collect information about the device for commercial purposes or for the purpose of information theft. In order to be characterized efficiently and completely, a certain type of malware must be analyzed in a hybrid way, both statically (without being executed) and dynamically (through its execution).

External network-level attacks complement malicious file-based threats. Therefore, the usual modus operandi is that external malicious actors can perform DoS (Denial of Service) attacks to overload services, phishing attacks to trick users into running malicious software, or brute force attacks to gain initial access. After this step, introducing a malicious file into the network is an easy task. If there are also malicious actors within the internal area, various privilege escalations, data exfiltration or even the intentional downloading of malicious files can be used. The impact comes from changes to the configurations on network equipment, deactivating access control policies, account blocking or lateral movements. Citing [6], external attacks are carried out by unauthorized users, who enter the network in the form of intrusion. Their specific modus operandi includes launching and intercepting malicious packets to gain access to private network resources. On the other hand, internal attacks are launched directly from the private area and include violating internal rules, with the aim of degrading network performance. It should be noted that the threat actor pool and the number of security incidents is growing, which means that, although it may seem counterintuitive, organizations with fewer devices are increasingly targeted. Thus, it is imperative to develop effective defense policies and understand, in as much detail as possible, how network resources and attacks work.

The experiments conducted in this study focus on replicating and analyzing specific attack behaviors to observe how threats affect network traffic and device performance. The proposed approach demonstrates that even within the confines of a non-dedicated simulation tool, significant insights into attack mechanisms and defense responses can be obtained. Furthermore, this framework provides a cost-effective and reproducible environment for cybersecurity research, giving students and professionals the opportunity to study real-world threats without jeopardizing live systems. Also, the results encourage further exploration of hybrid simulation methods that combine traditional network tools with security-focused experimentation, helping to bridge the gap between theoretical knowledge and practical defensive practice.

Despite significant progress in malware analysis techniques, existing approaches remain limited in their ability to capture the full complexity of real-world attack behavior. Current methods also focus primarily on host-level changes, overlooking crucial network-level behaviors such as command-and-control communication or lateral propagation. These limitations restrict the accuracy, scalability, and reproducibility of malware research. Consequently, there is a growing need for adaptable, network-oriented frameworks that can safely simulate attack behaviors and analyze their effects under controlled yet realistic conditions.

Despite the availability of malware sandboxes, there is limited research integrating full-network emulation platforms like EVE-NG with SIEM systems for behavioral malware analysis. So, the main objectives of this study are:

- to emulate a realistic multi-branch enterprise network using EVE-NG,
- to simulate network and malware-based attacks,
- to evaluate SIEM performance in detecting and analyzing intrusions.

This paper is structured as follows. Section II and III review the existing literature and background work in the cybersecurity field. Section IV outlines the methodology employed in developing and implementing the proposed system. Section V presents the results obtained from analyzing malware samples and network attacks. Section VI discusses the implications of these findings, and Section VII concludes by summarizing the key insights and suggesting directions for future research.

II. Related Work

To achieve a comprehensive analysis of how network-level attacks or malware work, it is essential to examine relevant research efforts and commonly used tools in this field. Therefore, this section aims to provide a deeper insight into the current state of research, thereby positioning the present study in the broader cybersecurity landscape.

In study [5], a method for analyzing malicious files was implemented using a virtual test environment and a combination of methods such as dynamic analysis and Windows event logging as data input. To make this possible, a database containing both non-malicious samples (including well-known applications such as YouTube or those from sites such as Softonic) and malicious samples (from the open-source Malware Bazaar database [7]) was used. Following experiments comprising 60% legitimate and 40% malicious files, it was found that an MLP-type machine-learning algorithm achieved a remarkable accuracy of 91.2%. In this paper, malware samples from the same database were used, but the testing methods of these samples are different, using a classic sandbox and processes monitor approach to observe malicious behavior.

Another important work in the literature is [8], where an application of malware detection in systems that include Active Directory, is presented. For this detection, the authors used a new intrusion detection system based on provenance, called HADES. Its advantage, compared to conventional ones, is given by the tracking of the anomaly identified at the level of the entire network, through Active Directory and connection sessions. Basically, HADES works as a system that integrates the specific aspects of a SIEM system, bringing the observability context to the network level, and of an efficient intrusion detector of the Active Directory concept. In the current work, a SIEM system is also used, which leads to the capability of being able to observe intrusions within the network architecture presented in the next section. The difference between the two papers consists in the way of capitalizing on the information brought by the logs from the SIEM system and the attacks used in the experiments.

In research [9], a specific network aspect is presented, which includes the concept of DNS (Domain Name System) spoofing. This aspect includes installing malware on the victim computer and introducing a rogue DHCP (Dynamic Host Configuration Protocol) server into the network. The purpose of this paper includes performing a DNS spoofing and phishing attack. To perform these tests, a DNS spoofing engine from the Python programming language is also used, through which the web interface of a site is modified to replicate a legitimate one, and the phishing attack is performed. The test environment used for these experiments includes the Kali Linux operating system and the multi-vendor network emulation software (EVE-NG). This study uses the same test environment; however, the network architecture and experiments are more complex than in the cited paper. Table I shows comparison of related work.

TABLE I. Comparison of Related Work

Study	Environment	Methodology	Attack Type	Detection Mechanism	Accuracy / Outcome	Key Contribution
[5]	Virtual test environment	Dynamic analysis + Windows event logs	Malicious/benign file samples	MLP-based machine learning	91.2% accuracy	File-level malware classification using hybrid analysis
[8]	Enterprise network with Active Directory	Provenance-based IDS (HADES)	System/network anomalies	Rule tracing via AD sessions	Qualitative results, improved network visibility	Network-level anomaly tracking using provenance data
[9]	EVE-NG full-network emulation	Malware installation + rogue DHCP + DNS spoofing	DNS spoofing and phishing (web interface tampering)	Network and host-level logging	Qualitative demonstration of successful DNS spoofing/phishing	Demonstrated attack chain combining rogue DHCP and DNS spoofing to enable phishing
[10]	EVE-NG full-network emulation	Malware installation + rogue DHCP +	DHCP starvation	Network and host-level logging	Qualitative demonstration of DHCP attacks	Demonstrated DHCP attack
[11]	EVE-NG environment simulating real firewalls	Full-network emulation of production systems	DoS and intrusion attacks	Log and alert analysis from virtualized network devices	Similar setup to present work but focused on device configurations	Modeling of real firewall behavior and testing network attack responses
[12]	EVE-NG vs. Packet Tracer and GNS3	Comparative evaluation of network emulators	General network simulation	Feature and resource analysis	Identified firmware-dependency limitation	Benchmarking of EVE-NG's strengths and constraints
[13]	Broad cybersecurity framework (Wireshark, VirusTotal, Google Rapid Response)	Literature-based analysis of cybersecurity evolution	Incident response, malware detection	Theoretical and empirical synthesis	Comprehensive understanding of cybersecurity methods	Overview of tools and practices for malware discovery and response
This work	**EVE-NG full-network emulation**	**Network and host-level logging, SIEM (OSSIM)**	**Nmap, hping3, real malware samples**	**Rule-based SIEM correlation**	**Binary detection (logged / not logged)**	**Evaluation of SIEM visibility for real attacks in emulated networks**

The aspects presented in the paper [10] include similar concepts to the paper [9] because network-level attacks are also used. In this case, the focus is on the DHCP protocol, and the attack brought to the fore is DHCP starvation. The objective of the paper is to have an efficient way of authenticating clients through the DHCP protocol and to prevent DHCP replay attacks. Thus, through the Python programming language, a token exchange is generated that includes the pre-shared key (PSK), MAC address, PIN and system time of the clients. The testing environment was similar to the one present in this paper, EVE-NG, but the reason for this paper did not include security concerns, but only efficient operation, with minimal consumption of hardware resources.

Considering [11], it is concluded that the common operating systems, Linux and Windows, within a network, are protected quite inefficiently if the protection of network devices is missing. Thus, in this work, through the EVE-NG simulation environment, real network devices (firewalls) are modeled, which constitute a complete attack testing environment. As is also specified in this work, the environment created simulates a real production environment. It is mentioned that the experiments carried out in the cited paper are similar to those present in this one, the key differences between the two being given by the fact that, in this paper, the emphasis is not placed on the equipment configurations, but on the results obtained following attacks. Another essential difference is that, in this work, a SIEM type system is implemented, while in the cited work the network security is done through a firewall type device (Web Application Firewall).

Taking into account paper [12], a comparison of the EVE-NG emulation software with other similar tools such as Cisco Packet Tracer and GNS3 is considered. The authors specify that EVE-NG is a robust environment for complex network scenarios, an aspect also certified within this work and has the potential to be an extremely good test environment for the accumulation of knowledge. However, to the authors' claims, an unfavorable aspect of this environment must also be considered: network resources are unavailable unless the firmware images are pre-obtained. This aspect implies having different user accounts or different licenses, otherwise downloading resources is sometimes impossible.

From study [13], a comprehensive understanding of the field of cybersecurity is provided. Thus, useful information can be extracted from the work regarding how incident management is effective at the moment, its development throughout history or the most effective tools needed for malware detection. Throughout the study, a comparison is also made on the detection of a malicious file, which is approached from the perspective of several traditional methods, in relation to those used in antivirus systems. The public resources presented include tools such as Wireshark [14] (which is also used in the experiments in this work) VirusTotal [15] and Google Rapid Response (GRR) [16], used in malware discovery and analysis. However, an essential idea from this work remains the fact that no hardware or software approach can completely detect all forms of malicious software (neither from the area of traditional methods nor from the area of Machine-learning or cloud).

The references cited throughout this section highlight various approaches similar to the one in this paper. Some of them include key concepts from the cybersecurity framework, some include aspects about malicious files, and some include the same testing environment, such as EVE-NG. However, the present paper presents a much more complex network architecture and a way to test the EVE-NG environment different from the aspects in the literature, bringing added value to the field of cybersecurity and to the methods of learning the detection of attacks or software with malicious potential. In contrast to previous studies that examined individual components or specific protocols, this work combines full-network emulation with the execution of actual malware samples and centralized monitoring through a SIEM system.

III. Background Work

Implementing a complex network architecture in a simulated environment emphasizes reproducing production characteristics. Thus, aspects such as border or core routing, network segmentation, access control policies, firewall rules and datacenter area are imminently integrated. In this work, in addition to all of these, elements of malicious attack and network attacks were also included, to model a more unusual behavior of the network and test it under unfavorable conditions. However, EVE-NG is not an environment designed for such operations, but rather for didactic and research purposes.

Another important aspect to mention is the SIEM system. AlienVault OSSIM [17] performs centralized log collection, both from the workstations and from the network devices, their correlation in real time (or in the shortest time available), their display and the creation of alerts, depending on various events or predefined thresholds.

For these things to be possible, an agent is installed at the level of each device in the network, through which the connection with the log manager and its dashboard is ensured. The goal is to identify, analyze and respond, in the most efficient way, to various intrusions or attacks on the network. Platforms such as Wazuh [22], Splunk [23] and QRadar [24] have been studied and analyzed and even compared in terms of their log collection, correlation and alerting capabilities and what makes them different are the characteristics of scalability, detection accuracy and integration flexibility.

Being one of the key subjects of the experiments carried out in this work, the cyberattacks and malicious files require additional explanations. Thus, if we consider network-level attacks, such as network scanning and denial of service, which are more easily implemented in EVE-NG, it must be borne in mind that the purpose of performing them was to test the capabilities of the working environment and create customizable alerts. For the malicious files, the test environment created was the demanding one.

Thus, having the attacks at the network level already performed, some files with malicious software were needed. Although there are a lot of pre-labeled databases, the purpose of these work scenarios was not simply to adapt them, but to perform experiments with real samples. To obtain these samples, the MalwareBazaar Database [7] was used.

This open-source resource includes various types of malicious files specific to Windows and Linux operating systems. The method of choosing the samples did not include a predefined algorithm, but rather the use of a ransomware file and a malware was desired, in order to observe the impact on the work network.

Due to their unpredictable nature, these files were chosen with the aim of carrying out experiments, which can contribute to improving the literature in the field of cybersecurity.

All these aspects, along with other, more advanced details, will be found in the following sections.

IV. METHODOLOGY

Monitoring and analyzing network traffic are essential aspects in protecting infrastructures against cyberattacks. Network anomaly detection aims to identify unusual behaviors such as malware attacks, data theft or unauthorized access attempts. This aspect is imminent within infrastructures that include various types of cybersecurity measures and are essential for any network infrastructure.

The process of detecting them includes certain stages that are based on advanced data analysis or machine learning techniques.

The first one is network data collection: data packet analysis (which includes information about its source and destination, the protocols used and the packet sizes), data flow analysis (NetFlow/sFlow – which can lead to understanding the general behavior of network traffic and can allow the identification of anomalies at the session level) and security log analysis (generated by security equipment, such as firewalls or Intrusion Detection System/Intrusion Prevention System – IDS/IPS).

The second one is creating a "normal" traffic model through identifying traffic typologies (by classifying it according to the nature of various characteristics such as applications and users Web traffic, File Transfer Protocol connections, etc.) and user behavior (by identifying "normal" user behavior).

The third one is continuous monitoring and anomaly detection through machine learning algorithms (identifying anomalies through behavioral models). An example, in this case, is volumetric analysis, which determines Distributed Denial of Service (DDoS) attacks and anomalies in user behavior (deviations from the usual activity of a user or a group of users).

The last one is creating alerts and automatic responses by isolating affected resources and further investigating the attack.

Within the network architecture created for the test scenarios that will be developed throughout the experiments in this work, several different parts of the network were included. Some of these are the data center area (where several types of servers were implemented), the data acquisition and detection part (given by the SIEM Manager), the monitoring area (given by monitoring servers and various end-devices) and some branch areas (that have specific functions). These details can be found in Fig. 1, where some abbreviations appear, whose explanation is necessary.

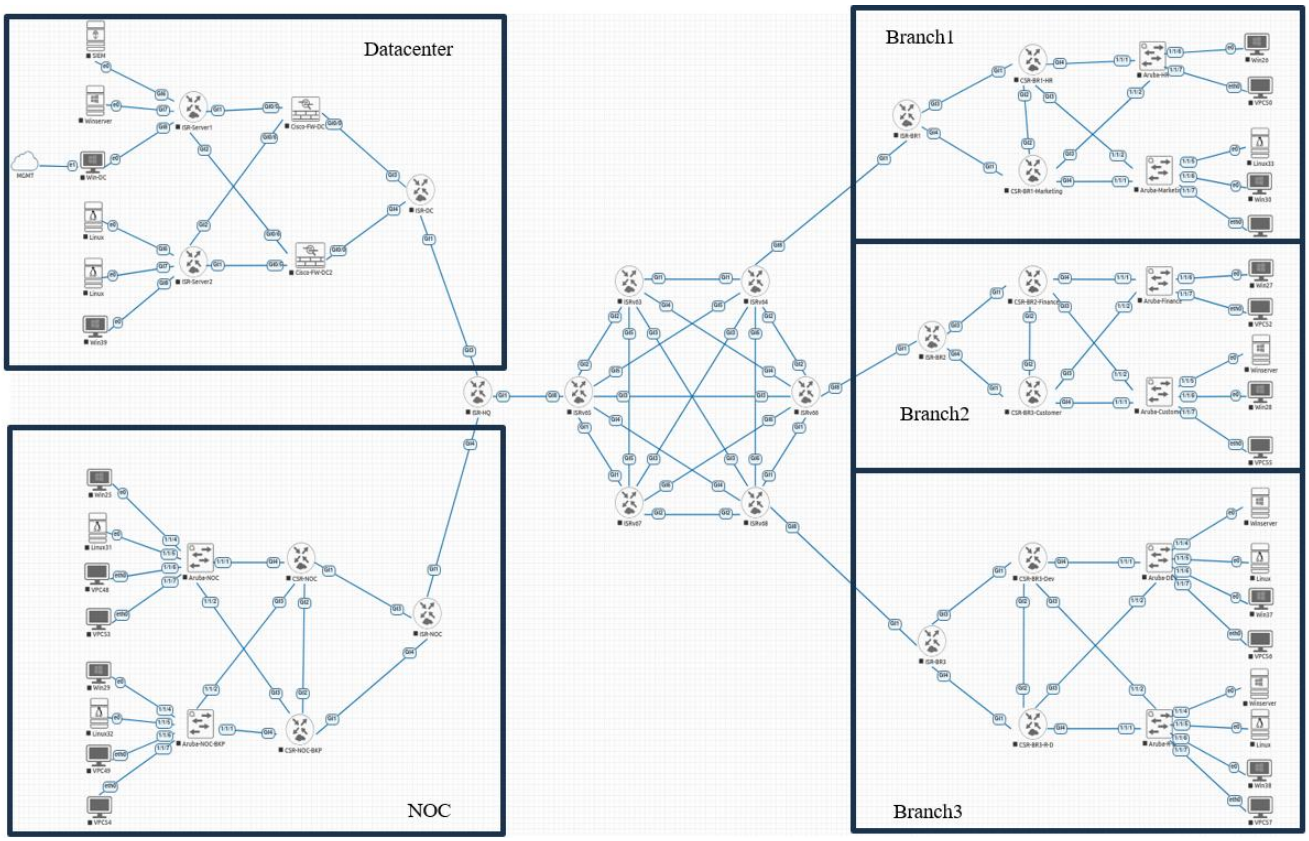

Fig. 1. The analyzed network architecture.

In the datacenter block, located in the upper left part of the figure, there is a SIEM system implemented, the external network management area, which ensures file transfer and other types of communications with the external environment, servers with the Windows Server, Linux or Kali Linux operating system, client stations with Windows and Linux operating systems, ISR routers (to ensure communication from the datacenter area with the rest of the network) in a configuration that also ensures a backup, in the event of an unforeseen event, firewalls (to protect access resources in and from the datacenter) and a router through which data is transmitted to the branch area and to the NOC (network operations center) area.

The NOC block and the branches area (located in the lower left and throughout the right side of the architecture) include Windows and Linux client stations, Aruba access switches, routers in a backup configuration to ensure communication with the external environment and the efficient transmission of CSR monitoring data and ISR gateway routers.

The central block features a full mesh network infrastructure, in which all ISR routers are connected to other routers of the same type. In this way, the internet resources within this infrastructure are simulated. Practically, the test scenarios will include both attacks from the datacenter area, although these are quite rare, and from the NOC (Network Operations Centre) area, but also from the branch area. Therefore, in this way, it will be possible to test the acquisition method and the efficiency of the SIEM system, as well as the propagation method of various types of attacks at the network level.

The entire test infrastructure was implemented using the EVE-NG solution, along with various network components, downloaded from the public resource area. The proposed network architecture included an available hardware environment of approximately 2 TB storage and 512 GB RAM, so that all equipment can function properly, at the same time. Several types of equipment were chosen, from several suppliers, so that the scenario was as realistic as possible and closer to a real work environment. All monitoring infrastructure and agents were deployed as guest virtual machines within the EVE-NG topology.

The SIEM server was provisioned on a virtual machine, also within the EVE-NG platform and configured according to the vendor's documentation.

The branches were created to perform specific functions, such as:

- Human Resources Department – the area that includes the human resources management part.
- Marketing Department – the area that deals with promoting the organization's resources.
- Financial Department – the area that ensures the financial side of the organization.
- Customer Support Department – the area that ensures the interface with the organization's customers.
- Software Development Department – the area that implements IT resources.
- Research Department – the area that innovates IT environments or other types of environments in the organization.

These are key areas of any medium-sized organization and require permanent monitoring and constant communications with the datacenter area.

The NOC part ensures the monitoring of resources, both in the datacenter area and in the branch area, in order to have a centralized view at the level of the entire network. In fact, although the SIEM system has its physical connection in the datacenter area, it is accessed, through its graphical interface, also from the monitoring area. Through this way of working, important resources and attacks at the network level are constantly visualized, with minimal possibilities for the NOC team not to notice an attack.

Agent installation was performed inside each endpoint virtual machine using the native installer for the respective operating system (using, in most cases, the AlienVault documentation).

Agent registration used the SIEM server's EVE-NG internal IP address, TLS keys, and authentication tokens.

Local log forwarding was configured to point to the local agent, which in turn forwarded the logs through an encrypted channel to the SIEM server virtual machine.

Basically, the entire infrastructure worked on the agent-server-dashboard principle, where the agent provided logs to the server, which in turn displayed them and created alerts based on them. Therefore, the logging flow can be summarized as: (emulated host VM) → (local syslog/agent inside the VM) → (encrypted agent channel) → (SIEM server VM inside EVE-NG) → (indexer/correlation engine) → (dashboard).

This aspect leads to the idea that, for a medium business environment, such as the one represented in the figure below, the most expensive equipment will not always be used to standardize the network, but there will also be equipment that includes a lower cost, but with functionalities similar to the "premium" ones.

Attacks at the network level, as well as its actual architecture, provided an ideal framework for observing the behavior of a real infrastructure in the event of an intrusion. Therefore, although it is a simulated environment, through the EVE-NG platform, this framework can constitute a secure basis for performing several types of work scenarios, in a controlled environment, without affecting the production infrastructure.

All malware traffic came from attacker virtual machines created in the EVE-NG topology, so that all malicious activity remained confined to the emulated environment. Widely adopted open-source tools was used, in order to ensure reproducibility. Representative tools and sample commands are:

- Hping3 (v0.9.0) — used inside the attacker virtual machines to generate volumetric and specially crafted packet floods sudo hping3 --flood -S -p 80 --rand-source 10.10.10.20.

- Nmap (v7.x) — reconnaissance from the attacker virtual machines (example): nmap -sS -A -p1-65535 10.10.10.0/24.

For each attack scenario, the attacker and victim virtual machine identifiers in EVE-NG and the precise start/stop timestamps were recorded in the SIEM system.

The purpose of carrying out these experiments is to evaluate the infrastructure's response in the event of an attack, to test various types of solutions to choose the best one and to test several types of countermeasures, which can improve the resilience of the network. To achieve this goal, several types of attacks were tested, both at the network level and through malicious files, various types of security solutions and different types of sources that can cause security incidents, both from the external and internal areas, accidental or intentional.

The experiments were designed to be repeatable by leveraging the EVE-NG snapshot functionality and a fixed running protocol. For each scenario, multiple independent runs were performed. Each run followed these steps:

- Bringing all virtual machines to a clean state.
- Starting SIEM services and agents and confirming agent connectivity for at least 60 seconds.
- Collecting normal traffic from monitored virtual machines.
- Initiating the attack sequence from the attacking virtual machines for a minimum duration of 60 seconds.
- Continuing monitoring for a period of 120 seconds post-attack.
- Archiving logs and exporting SIEM alerts for analysis.

After each run, the initial state of the virtual machines was restored to ensure isolation and eliminate side effects.

Practically, this section will include all the resources mentioned above, with the main purpose of having a unified exemplification of the concepts mentioned above and which are indispensable for a secure network.

The infrastructure presented in Fig. 1 was considered in order to be a simulated environment of a real network, with the purpose of testing various functionalities, in order to be able to observe the real implications. However, what should be mentioned is that the hardware and software resources owned by the real network are infinitely larger, compared to its simulated version. Nevertheless, this minor inconvenience is much less important than the purpose of the experiments carried out through this infrastructure.

Although during the experiments, several attack scenarios will be used, in a real case, analyzing the infrastructure in Fig. 1, the place of origin with the highest probability of achieving an intrusion is through a branch office. This unfavorable aspect is caused by the fact that, within them, there are more "relaxed" in terms of security measures and, sometimes, even the protections of certain resources are implemented in a more permissive and trust-based way.

This aspect is extremely unfavorable for the general security measures of the network because, in the event of an attack, the affected resources will increase exponentially. As a result, the attack can also extend to the Data Center or critical infrastructure area and can affect both the image of the organization and resources that should be operating in a constant manner.

Also, if we put aside the technical area, the huge impact of an attack will have financial implications, both for measures to prevent the spread, and for the remediation of the affected aspects (ransomware) or for the subsequent use of more effective protection measures.

The experiments were designed to be repeatable by leveraging the EVE-NG snapshot functionality and a fixed running protocol. For each scenario, multiple independent runs were performed. Each run followed these steps:

- Bringing all virtual machines to a clean state.
- Starting SIEM services and agents and confirming agent connectivity for at least 60 seconds.
- Collecting normal traffic from monitored virtual machines.
- Initiating the attack sequence from the attacking virtual machines for a minimum duration of 60 seconds.
- Continuing monitoring for a period of 120 seconds post-attack.
- Archiving logs and exporting SIEM alerts for analysis.

After each run, the initial state of the virtual machines was restored to ensure isolation and eliminate side effects.

Practically, this section will include all the resources mentioned above, with the main purpose of having a unified exemplification of the concepts mentioned above and which are indispensable for a secure network.

The infrastructure presented in Fig. 1 was considered in order to be a simulated environment of a real network, with the purpose of testing various functionalities, in order to be able to observe the real implications. However, what should be mentioned is that the hardware and software resources owned by the real network are infinitely larger, compared to its simulated version. Nevertheless, this minor inconvenience is much less important than the purpose of the experiments carried out through this infrastructure.

Although, during the experiments, several attack scenarios will be used, in a real case, analyzing the infrastructure in Fig. 1, the place of origin with the highest probability of achieving an intrusion is through a branch office. This unfavorable aspect is caused by the fact that, within them, there are more "relaxed" in terms of security measures and, sometimes, even the protections of certain resources are implemented in a more permissive and trust-based way.

This aspect is extremely unfavorable for the general security measures of the network because, in the event of an attack, the affected resources will increase exponentially. As a result, the attack can also extend to the Data Center or critical infrastructure

area and can affect both the image of the organization and resources that should be operating in a constant manner.

Also, if we put aside the technical area, the huge impact of an attack will have financial implications, both for measures to prevent the spread, and for the remediation of the affected aspects (ransomware) or for the subsequent use of more effective protection measures.

However, the most important implications are given by attacks in the NOC area or in the Data Center area because the people who have access to this data have much greater visibility over the entire infrastructure. Thus, attacks can be tested before the actual attack so as not to be easily detected or can be hidden within legitimate files, until they are detonated. Although it is less likely that users of these areas of the network infrastructure will initiate an attack (due to the level of training, strict policies or secure access to resources), if they do, the impact can be devastating.

The most common risks are data from compromised devices, intentionally or not, users unfamiliar with the cybersecurity area and the implications of security measures, phishing attacks that led to the attacker finding out key credentials, improperly secured Wi-Fi networks, configuration errors that can be exploited or simple credentials (admin/admin type).

The most likely attacks at the network level are infection with malicious files, data exfiltration, erroneous or apparently erroneous configurations that add additional, unauthorized routes, exploitation of weak security measures, use of firmware backdoors, DoS, unauthorized access to various types of resources, etc.

Some of these attacks will be presented in the following section in order to achieve a more realistic example of several types of attack scenarios and to be able to identify network vulnerabilities and the most effective methods of protecting them. The focus will be on network scans, identification of network flooding attacks or Denial of Service (DoS) attacks.

V. Results

In this section, we will consider several types of attacks, both network-based and file-based. The main goal is to visualize the attack response time and effectiveness within a fully virtualized system.

A. Network Scanning

Network scanning attacks are used to allow an attacker to view the essential characteristics of certain network devices or end-devices in order to create a network topology and discover various vulnerabilities. These vulnerabilities can be found both at the operating system level and at the exposed services level, and their exploitation is done in a unique way by each attacker.

In the experiments carried out at the network level, through the EVE-NG platform, the scans at its level were successful in detecting IPs and open ports in most of the devices tested (both for network devices: routers, switches, and for devices used by a normal user).

The main advantage was given by the SIEM system, which managed to capture most of these attacks, even though the network logs were extremely difficult to read and process due to the large volume of data. This aspect highlights the importance of monitoring and log capture measures both in complex and simple networks.

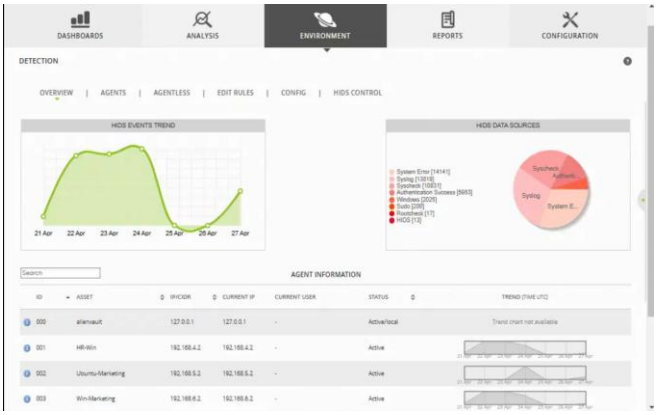

Fig. 2. Network architecture visualization mode at the SIEM system level.

Fig. 2 shows how the SIEM system analyzes the available agents and manages them efficiently, capturing their logs. As can be seen, the agents come in several categories, including various types of operating systems (Linux or Windows).

In the first test scenario, both attacks from the network area that includes the Marketing architecture and the network area that includes the HR part will be analyzed, both being contained in the same branch, as can be seen in Fig. 1.

Consequently, in this work scenario, both operating systems (Linux and Windows) were used as victim resources, and the attacks were generated using a Kali Linux virtual machine. This is not common in a normal network architecture, but the fact that the attacker machine is not a regular Ubuntu or CentOS does not impact the result at all because the attacks were generated using regular Linux commands and not with specific Kali Linux tools.

The first attack was carried out on the company's own network infrastructure, from the marketing area to the same network, to observe the available devices and open ports. This aspect led to the identification of certain attacks, which can be focused on those ports to destabilize the infrastructure. The way of carrying out the attack included running, through Kali Linux, the nmap 192.168.5.0/24 (local network) command.

Although the initial expectations were that the SIEM system would identify this attack and alert security administrators to this fact, it was concluded, based on experiments, that the Security Event and Information Management system did not even register the event, it was "lost" in the multitude of network logs.

The next work scenario targeted an attack from the Marketing area to the HR area, maintaining the same characteristics as in the previous case. Therefore, another, more aggressive attack was generated using the nmap -A 192.168.6.0/24 (distant network) command, which led to the identification of several features of the HR network area.

This time, the attack was recorded at the Security Information and Event Management system level, and the logs were displayed in a relatively short time at the dashboard level. Therefore, in this case, the SIEM system functioned effectively, recording the attack so that security administrators could

intervene to identify and isolate it. Following this type of event, a rule could be created to trigger an automatic action, which would isolate the traffic generated by an attacker workstation. Therefore, the resulting behavior consisted in the fact that the attacker station's IP is declared suspicious and the scan stops, the attack not being directed to the initial destination.

The way this type of attack is detected at the OSSIM level is through sensors that identify Nmap, Masscan or network scanning actions. These sensors usually generate an event in the dashboard that has a "Scan Detection" or "Potential Network Scan" alert. Following this event, alerts can be created to automatically block this type of traffic. In this case, the attack is not automatically blocked due to the fact that the test mode includes successful attacks, in order to test the efficiency of the Security Event and Information Management system in capturing logs from devices and displaying detected attacks or intrusions.

In the last scenario in the network scanning subchapter, the attack was carried out directly to the SIEM, in order to observe if and how it will prevent it. The attack was also generated through Kali Linux, as can be seen in Fig. 3 and was recorded in the SIEM system dashboard (Fig. 4). The important aspect is given by the fact that the attack led to temporary interruptions of the SIEM system, even if it consisted only of a trivial scan. Thus, even a small change, at the resource or configuration level, can lead to the total or partial interruption of log capture, leaving, at the same time, various types of attacks unnoticed and unverified. Practically, the latency in the response can indicate high CPU consumption for the analysis of each packet whose destination it is. In this subchapter, two different attack scenarios were presented in which the attacker was a Kali Linux device. The attacks were directed to a device within the network and to a device from another network, but in the same branch.

Although the initial attack was not recorded by the SIEM, the other two were recorded effectively and even alerts were displayed for security administrators.

```
root@kali:~# nmap -A 192.168.0.150
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-27 08:53 EDT
Nmap scan report for 192.168.0.150
Host is up (0.0031s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.4p1 Debian 10+deb9u7 (protocol 2.0)
| ssh-hostkey:
|_  2048 3c:15:73:09:9f:2b:8e:ad:fb:f5:15:c6:99:01:4f:59 (RSA)
80/tcp  open  http     Apache httpd
|_http-server-header: Apache
|_http-title: Did not follow redirect to https://192.168.0.150/
443/tcp open  ssl/http Apache httpd
|_http-server-header: Apache
| http-title:  AlienVault OSSIM
|_Requested resource was session/login.php
| ssl-cert: Subject: commonName=alienvault
| Subject Alternative Name: email:system@alienvault.com
| Not valid before: 2025-04-19T18:20:11
|_Not valid after:  2035-04-17T18:20:11
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
514/tcp open  shell?
Warning: OSScan results may be unreliable because we could not find at least 1
 open and 1 closed port
Aggressive OS guesses: Linux 3.12 - 4.10 (94%), Crestron XPanel control system
 (90%), OpenWrt Kamikaze 7.09 (Linux 2.6.22) (90%), Linux 3.2 (89%), OpenWrt 0
.9 - 7.09 (Linux 2.4.30 - 2.4.34) (88%), OpenWrt White Russian 0.9 (Linux 2.4.
30) (88%), ASUS RT-N56U WAP (Linux 3.4) (87%), Linux 3.1 (87%), Linux 3.16 (87
%), HP P2000 G3 NAS device (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
```

Fig. 3. Network attack including aggressive scanning of the SIEM system.

AlienVault HIDS: Windows error event.	2025-04-27 08:54:26	alienvault	N/A	Win-Marketing	Win-Marketing	2->2	LOW (0)	
AlienVault HIDS: Windows error event.	2025-04-27 08:54:26	alienvault	N/A	Win-Marketing	Win-Marketing	2->2	LOW (0)	
AlienVault HIDS: SSH insecure connection attempt (sca n).	2025-04-27 08:53:58	alienvault	N/A	Ubuntu-HR:56232	0.0.0.0	2->2	LOW (0)	
SSHd: Did not receive identification string	2025-04-27 08:53:57	alienvault	N/A	Ubuntu-HR	0.0.0.0:22	2->2	LOW (0)	
AlienVault HIDS: Windows error event.	2025-04-27 08:53:30	alienvault	N/A	Windows-HR	Windows-HR	2->2	LOW (0)	

Fig. 4. Displaying the attack in the SIEM system dashboard.

B. Denial of Service Attack Simulation

DoS (Denial of Service) or DDoS (Distributed Denial of Service) are some common network attacks that involve generating TCP connections with a large volume of data, which aim to temporarily block access to resources, affecting certain services. These types of attacks are usually focused on a specific device, and the impact consists of intentionally reducing its performance. In the favorable case where this system also has a backup connection, services will only be affected to a small extent, but in a real infrastructure, there are extremely few network areas that operate in this logic.

The essential countermeasures for this attack not to be successful consist of limiting multiple connections to a specific device, blocking various queries by firewall devices, equipment that operates in HA, devices that have integrated the concept of load-balancing, etc.

Moreover, if the attack comes from "inside", a good part of these measures becomes ineffective anyway because they have already been bypassed by the attacker. Therefore, it becomes imperative to be aware of the impact that an attack carried out from the internal area can have on the infrastructure.

To prevent these types of attacks, more access control measures must be implemented, SIEM systems that also integrate user behavior analysis (based on machine-learning) have to be considered and resource segmentation measures should be used.

The impact of this type of attack, from an experimental point of view, consisted in affecting network communications, temporarily interrupting certain essential services and degrading network performance. Moreover, the loss of certain packets at the level of a network of the size of the one in Fig. 1. is normal. This is true even if VPN connections are made between the Headquarters area and the branch area.

However, in the case of a DoS attack, the network's "defense" mode is to drop certain packets to cope with the overload. If the attack is long-lasting, at some point, the affected resource may even become unavailable, affecting even communication between certain parts of the network.

In the first working scenario, similar to the way the previous experiments were carried out, an attack will first be executed within the branch network, using the command:

hping3 -S –flood -V -p 80 192.168.5.2

The attack is recorded at the SIEM system level, but the alert is not clear enough to indicate the type of attack. However, multiple Windows errors indicate, for security administrators, abnormal behavior at the network level within the respective branch.

In the work scenario where the attack is centered on the Security Event and Information Management system (Fig. 5), the attack works in a similar way, but the SIEM system's reaction mode is extremely serious for the entire infrastructure because it becomes unavailable in the entire period of the attack.

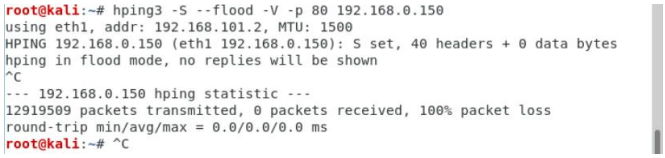

Fig. 5. DoS attack carried out on SIEM.

The immediate impact is given by the fact that, during this period, the Security Event and Information Management system does not record the attacks completely, and this aspect can completely affect the analysis network and the entire infrastructure of the organization. In practice, the SIEM system still receives logs from the devices in the network, but it can partially lose various events, and these are unrecoverable.

In this working scenario too, the EVE-NG test environment has proven important to simulate certain network behaviors in case of attack, allowing various types of important conclusions to be made, through the experiments carried out.

C. Malicious File Attacks

During the experiments, network intrusions through malicious files were also considered. The way in which these files came to penetrate the network was not included in the experimental data because it is not the scope of this work.

Consequently, also through the EVE-NG virtualized environment, a realistic scenario was created in which attacks originating from the internal area of the network were simulated and which are masked in legitimate files, which are executed due to lack of knowledge of their real nature. This type of attack is used to compromise the security of systems and to steal certain data about users or about the organization. The goal is to obtain various financial facilities or to steal various databases or ideas, for the purpose of reselling them or for the purpose of blackmail.

The test network included all the elements in Fig. 1, and the attack scenarios were triggered manually or automatically, through malicious files, transferred via FTP or downloaded from the public resource area, by a legitimate user. The attacks in this test scenario, however, came from several different parts of the network, in order to test both the intervention from the NOC area and the intervention from other branches, compared to the one tested in the previous scenario.

Basically, in this test scenario, the entire network was intended to be involved, but due to the multitude of logs, the SIEM system worked inefficiently, having moments of crashing or difficult event recording. Thus, for efficient detection, in certain experiments (based on more critical files such as ransomware), unused resources were closed. This way of working implies a more efficient use of resources in the test scenario through EVE-NG, but in a real test scenario, this is almost impossible, the purpose of the SIEM system being, mainly, to capture logs from all devices used in the network. Therefore, in this test scenario, the proportion of effective tests in EVE-NG, whose conclusions can be directly translated into the real working environment, was not respected. However, the way the SIEM system works, the way alerts appear and are customized at their level, and the way scripts that use these alerts are identified, can be implemented based on these tests.

The biggest disadvantage of these experiments in EVE-NG and the real environment is that, although the network in Fig. 1 is a large-scale network, it is nowhere near the size of a real (production) network. However, the resources used by the SIEM system to process all these logs are quite high, but the processing is not that efficient. In addition, considering the fact that the volume of data processed is insignificant compared to the real volume of data, the performance analysis on the test environment cannot fully reflect the multiplicity of the real environment.

Considering that in EVE-NG, the resources used do not have access to actual internet resources, in some experiments, Kali Linux devices were also included. The explanation for their use, compared to the use of regular Ubuntu devices, is that the former have more pre-installed libraries. An example, in this case, is the SSH (Secure Shell Protocol) service, which, in Kali Linux, has both the server and client sides installed, while in Ubuntu, it only has the client side. Therefore, Kali machines are considered regular users and agents of the network and not attacker machines.

The attacks were analyzed dynamically, including monitoring of CPU and memory resources and detection by antivirus on the victim host. Also, in some experiments, Wireshark tool was included.

One of the most relevant tests was the one regarding a ransomware attack, implemented by executing an .exe file, from the public resources area. Therefore, within this scenario, two test modes were implemented: a ransomware attack originating from the NOC area and an attack originating from one of the branches. The difference between the two types of attack was given by the affected operating system, focusing on the comparative analysis of a ransomware attack in a Linux-based operating system and in a Windows-based operating system.

The main purpose of the two types of attack was to visualize how the SIEM system registers the intrusion and to observe the impact that this malware attack has on network resources (both on the affected workstation and on its adjacent resources: routers, switches or other workstations).

The aspects observed after execution recorded typical behaviors of a ransomware file, this file scanning the disk within Windows and Linux systems and encrypting all accessible files,

in an extremely short time. Each file also received a different extension and a .txt file was created, highlighting an email and an amount through which the affected data can be recovered. It is mentioned that the behavior of the ransomware file was not detected by the antivirus system at the time of its download, remaining undetected until the time of execution. The effect on the infrastructure was irreversible, the files being inaccessible, without a backup solution.

The SIEM system did NOT manage to detect the attack in time (at the time of downloading the malicious file on the client's workstation), thus making the entire network vulnerable. And, at the time of the attack, the intervention on the resources was already far too late to be implemented effectively.

In the first experiment, a ransomware attack was carried out on the Ubuntu operating system, in branch 3, which contains the DEV department and the IT development area.

During the ransomware experiment, the execution of the .elf file [18] was attempted. However, this file required additional dependencies and dynamic libraries. Therefore, for the file to function properly, these dependencies had to be installed. Not having direct access to internet resources and having various problems with the offline installation of the necessary dependencies, this attack did not function properly, in a final version.

Within the SIEM system, the attack generated in the Ubuntu workstation is not visible, the logs not recording the execution of this file, neither when it did not have the necessary dependencies nor after they were installed in an offline mode, and the attack was run again (Fig. 6).

Fig. 6. Ransomware attack file execution from branch 3.

This aspect is quite complicated, given the nature of the malicious file executed because, even if, in the end, its execution did not affect the operating system, the fact that the file is not even registered at the Security Event and Information Management system level requires the existence of additional protection measures to compensate for this faulty way of working (by implementing IDS/IPS solutions or through application-type firewalls, which record and restrict such behaviors). The next working scenario included the same way of working, but within the Windows operating system. This time, the file chosen was .exe [19] and it worked properly, and the attack was successful, as can be seen from Fig. 7-9, which presents the entire evolution of the attack. In this case, contrary to the previous example, the Security Event and Information Management system detects malicious files and displays alerts at the graphical interface level.

Although, even in this case, the alerts are not very conclusive for the security team to know that there is a ransomware attack on the network, the multitude of errors that appeared in the dashboard require additional verification. An important aspect to note is that, although there was routing at the network resource level and the affected station was connected to the external environment (including the SIEM system, being its agent), the ransomware attack did not affect other resources on the network.

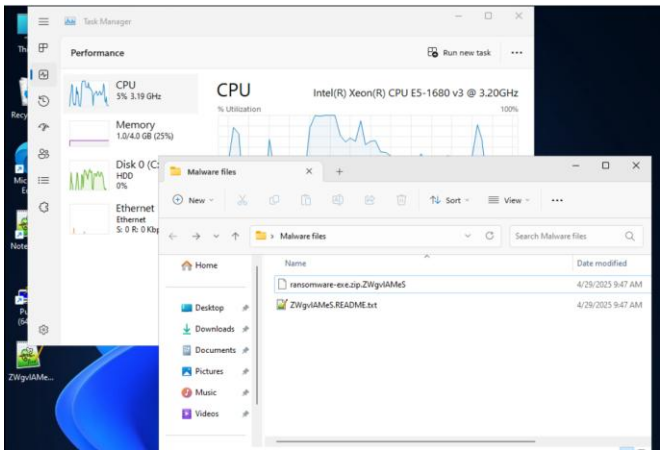

Fig. 7. Ransomware attack on windows operating system.

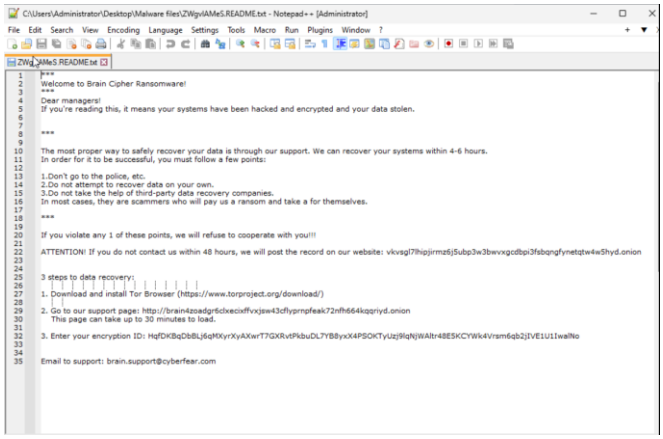

Fig. 8. Detailed README.txt of ransomware attack.

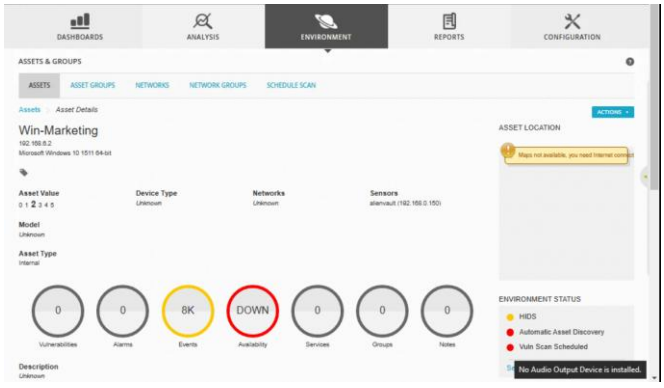

Fig. 9. The victim client station becomes inaccessible to SIEM after the ransomware attack.

Another extremely important aspect is the fact that, following the attack, the affected client station also became unavailable at the Security Information and Event Management system resource level, appearing DOWN, although it continued to function, only that its files had been encrypted. Therefore, it is considered that the ransomware affected the services, also

blocking communication with the manager, but did not specifically target the security agent.

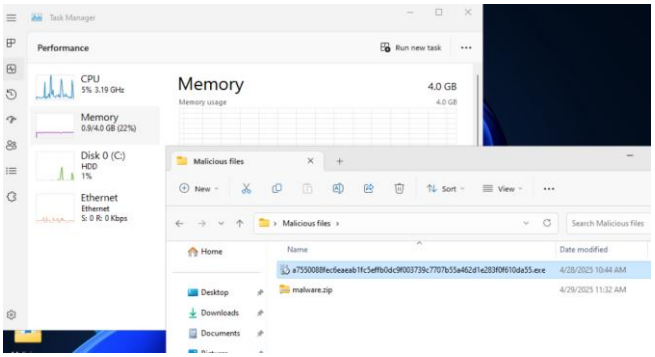

Fig. 10. Malware attack file execution from branch 3.

In a similar scenario, an attack with a malware file [20] on a Windows node was also considered, simulating another compromise, from NOC area (Fig. 10). The impact was given by the fact that activity was detected at the antivirus level, and the system resources (CPU, memory) were affected (CPU – 100%). At the level of the system that ensures Security Information and Event Management, the attack was detected, but the alerts are different, compared to those displayed previously, but still not clearly expressed enough to easily identify that it is a malware attack (Fig. 11).

In the present case, propagation in the virtual network is possible, due to the malicious nature of the file, but calls to the internet are not possible because the victim virtual machine does not have outbound access from the network. Thus, the file can see the LAN area, but, without a source of command and control, in the test environment used and during the experiments, it did not try aspects of lateral movement.

Fig. 11. Displaying the malware attack in the SIEM system dashboard.

The malware acts by running and modifying registries (from subsequent tests, it remains on that virtual machine even after reboot), but is limited to the actions performed, being valid only within the virtual workstation. To verify that the malware was propagated in the LAN area, some Wireshark and Process Monitor captures were performed on the victim station, observing both the malware scan logging and the behavior on processes.

In order to simulate a real internet environment, similar to some experiments that are carried out in real sandbox environments (not those based on EVE-NG), it would have been necessary to use tools such as FakeNet [21].

It can simulate network services and responses so that the malicious software does not notice the fact that it is a test, virtualized environment and reveals its real behavior. In this way, domains or command-and-control servers can be identified. In EVE-NG this simulation was not possible because, although a separate workstation was used, acting as a gateway, the malicious file does not adapt its behavior.

In this study, our goal was not to calculate statistical detection indicators such as false positive or false negative rates, but to verify whether the SIEM system (OSSIM) successfully recorded the occurrence of specific attacks in the environment emulated by EVE-NG. For each experiment, attacks were generated using hping3 (DoS and ICMP/TCP floods), Nmap (port reconnaissance and scanning), and two real malware samples executed in virtual machines. The outcome for each scenario was evaluated based on the detection and recording of the event by OSSIM (Recorded = Yes/No).

Table II provides a summary of all scenarios, indicating which attacks produced SIEM alerts.

TABLE II. SIEM EXPERIMENT RESULTS

Scenario	Tool/Sample	No of Runs	Recorded in SIEM	Alert Type
Port Scan	Nmap	2	Yes	SSH insecure connection attempt
DoS	hping3	2	Yes	Traffic Anomaly
Malware Sample 1	Ransomware1	1	No	-
Malware Sample 2	Ransomware2	1	Yes	Registry integrity checksum changed
Malware Sample 3	Malware	1	Yes	Registry integrity checksum changed

Another aspect that should be considered in this section is false positives. Given that in a network environment like EVE-NG there are a lot of devices generating traffic in parallel, a detailed explanation of all these values, which can appear at the SIEM level, is necessary.

The first one is about legitimate high-volume traffic because, sometimes, this traffic can generate attack-like patterns, but some VMs are doing normal operations (updates, backups, scans).

Another one can be when several VMs communicate in parallel and the SIEM may interpret aggregated traffic as suspicious.

The third one can be generated by multiple concurrent updates or temporary files can be flagged as suspicious.

Some alerts may correspond to benign parallel activity in the EVE-NG environment, representing potential false positives inherent in multi-device traffic aggregation. These were identified and differentiated from alerts generated by attacks by controlling the timing of the attacking virtual machine and even isolating a part of the network, when the attack was generated, in order to observe its impact on the SIEM system.

VI. Discussion

The experiments used in this chapter included several types of attacks generated at the level of a test infrastructure. Therefore, multiple attack scenarios were carried out, which included both network scanning attacks, DoS attacks and malware infections.

Considering that it is a test environment, EVE-NG cannot cover a real infrastructure, most of the resources being limited. Consequently, there are several cases in which the infrastructure operates slowly and, considering that a SIEM system also operates at its level, even storage will be even slower than in a real environment. The OSSIM Security Event and Information Management System requires substantial resources to function optimally. Being also an infrastructure in the EVE-NG environment, it can sometimes be overwhelmed.

Attacks carried out on this network are quite suffocating for SIEM, even if the hardware resources are sufficient because it must process an extremely large number of network logs, from all the devices. Moreover, during the attack experiments, a good part of the remaining devices was shut down, so that the logs could be tracked as easily as possible and so that the analysis would not overload the available resources. Event processing at the SIEM system level includes each network log being inserted into a database, being correlated with other events in the network and being displayed in a graphical interface. Thus, the processing time is extremely long and, sometimes, alerts appear with delays.

The SIEM system is built for the analysis of normal traffic events. The intrusion detection part is an additional one and, due to the fact that an EVE-NG system is also used, in which it is integrated, this area is sometimes inefficient. Although the Security Event and Information Management system is optimized to function as a network device in EVE-NG, it is not used to the massive volume of traffic, as in the case of previous attacks (directed to it). Therefore, when it detects an attack on it, it has two types of behavior:

- It can ignore the attack and continue recording events from legitimate devices.
- It may freeze, delay event display, or become unavailable.

In the work scenarios, it was required that the attacker know the IP of the SIEM system and that it not be protected by any firewall or other resources, but in a real work environment, this aspect is quite impossible for a branch user. It is not, unfortunately, impossible for a user working in the NOC area who could have access to such resources. However, most of the time, routing resources to the datacenter infrastructure level (where the Security Event and Information Management system is located) is not so accessible. As prevention methods, these would include separating the infrastructures at the datacenter level through hardware and software firewalls and segmenting the network in the most efficient way possible. Also, another urgently needed method is the implementation of domain policies that include limiting user access to tools such as nmap or to sensitive network resources.

A particularly important aspect is the ease with which this type of file can penetrate the network, in the absence of policies for their automatic detection. Therefore, without this type of monitoring and blocking policies, the files end up being executed, without the user being able to imagine their real intention. Therefore, inspection within the SIEM, but also the presence of an antivirus becomes essential elements for securing the environment in the endpoint area. Following the execution of the files, they had various behaviors, but most of them fell within the aspects mentioned in the chapter on static and dynamic analysis of malicious files. However, if we also take into account the impact aspects at the network level, it is found that they tried to generate various illegitimate connections, created process delays and resource consumption and various types of temporary blockages.

The disadvantage is that, in some cases, malicious files passed undetected from one computer to another, even if their activity was monitored by the SIEM system. Basically, the logs related to this type of transfer went unnoticed, the alert being created only at the time of execution. The fact that the SIEM system could not automatically block this type of file required manual analysis, thus implying the possibility of extending the attack to the entire network.

Simulating these attacks in the EVE-NG virtualization environment allowed highlighting the easy way with which malicious files can completely compromise a network, observing the behavior of the network in the event of an infection and testing the real infrastructure, without implications in the production area. However, a troublesome feature in the experimental area consisted of the direct implications that certain agents had when disconnecting from the manager's area. Therefore, considering that most of the experiments were carried out with closed network parts, when reconnecting them, there were various desynchronizations between the agent and the manager. Basically, the time in the agent device was no longer correlated with the time in the manager area, and the logs, including at the dashboard level, were not recorded. This way of working has been encountered previously in the real working environment, and, in both cases, it produces serious issues because the attacks are not recorded properly, and the SIEM system does not alert administrators in any way about this aspect. It is obvious that, without integrated security solutions and correct network segmentation, even in a controlled environment, attacks can have devastating effects. The experiments carried out have demonstrated that a simple file executed by a user with limited rights can serve as an entry point

for ransomware, backdoor or other forms of malware that escalate privileges and compromise the entire infrastructure.

In most cases, the SIEM system proved to be ineffective. Although the alerts provided raised the attention of network or security administrators regarding abnormal behaviors, they were quite vague, with no clear demarcation between the types of files that affected various operating systems and the way the alerts were displayed. Moreover, in some cases, the alerts displayed at the Security Information and Event Management system level are identical, although the types of intrusion files are different. If it is also considered the fact that, in some cases, the SIEM system did not detect a certain malicious behavior at all, we find the permanent need for additional intrusion detection measures (separate IDS/IPS systems).

The proposed method has potential applications that extend beyond the quality assessment of virtual devices, in a fully virtualized environment, such as EVE-NG. Therefore, the proposed method is considered useful for making informed decisions, in an environment that simulates a production environment, based on cybersecurity strategies that can adapt to multiple types of attacks and for multiple types of network devices or operating systems. This aspect also introduces efficient methods for learning and adapting the security policies of organizations, through various work scenarios, which can help to understand the cyber threat landscape.

Therefore, the main contributions of this work are the fact that a proprietary network architecture is proposed, divided into several different blocks, similar to a real network environment, which also involves resource segmentation. Also, the implementation of a SIEM system is considered to automatically detect network threats, operating in an agent-manager system, and including a centralized dashboard (where various alerts can be graphically viewed), like a real environment. Through this system, various logs can be collected from network equipment and workstations and automatic responses to intrusions can be created.

In addition, various work scenarios are tested, including both malicious files and network-level attacks, in an environment that is not specific to these types of actions. Also, the efficiency of these attacks is evaluated, both at the level of the network's own resources and at the level of the other parts of the network (for detecting lateral movements and for testing how an attack works at the level of the entire network).

Moreover, the EVE-NG working platform is evaluated, under the conditions of a complex network architecture and malware detection, at the SIEM level, although the platform was not created for these purposes, but only for common network tests and networking knowledge training.

To summarize, the main contributions of this paper are as follows:

- We designed and implemented a controlled cybersecurity experimental framework using EVE-NG, demonstrating its potential to model both network attacks and, more superficially, real malware attacks, in a secure environment.
- The feasibility and limits of using a non-dedicated network simulator for security research were assessed, providing insights into its adaptability and practical constraints.
- Experimental results were presented illustrating the behavior and network impact of common attack types.
- The potential role of EVE-NG as an accessible and cost-effective platform for cybersecurity training, testing, and education was highlighted.
- The response of various emulated systems to attacks or attack files was tested, in order to be able to formulate various preliminary conclusions to define their influence in production environments.

These contributions aim to reduce the gap between theoretical cybersecurity studies and practical experimentation, supporting both academic and practical advances in the field of network security.

Although the proposed work has multiple advantages, the imminent limitations of any system must be taken into account. The first one is that malware samples can identify if the sandbox environment is virtualized and can modify their behavior depending on this characteristic. Therefore, it would be much more appropriate for this simulation to be performed in a real test environment, not a virtualized one. However, most of the time, this is not possible because the malware execution can lead to partial or total degradation of the test environment.

The second one can be considered by the limited number of experiments because the specific behaviors of a certain type of malware or attack cannot be extracted except after successive runs of several experiments with similar malware samples or from the same family. Such work scenarios are considered in the following tests.

Although the experiments are performed in both Linux and Windows operating systems, the third limitation is that the mobile area is not covered in this work. Other operating systems are also considered in the following experiments.

Another limitation, the fourth, is that the alerts displayed at the SIEM dashboard level are not very explicit and can be confusing if similar malicious samples (trojan and malware) are used. This can make it difficult to quickly interpret security events and prioritize incidents. In further developments, increased attention will be paid to customizing alerts, so as to provide more detailed and useful information. It will also be considered to evaluate other types of SIEM systems, such as Wazuh [22], to test their ability to generate more intuitive and informative alerts.

The last one can be considered the fact that the test scenarios included experiments were performed in short period of time (up to one hour) in order not to degrade the test environment. This aspect is considered a limitation because some malicious files or some attacks become more effective after a certain period in which they "adapt" to the victim environment.

This type of test system is effective for supporting conclusions similar to the real environment because, in a virtualized environment, there are constraints on various types

of resources. Thus, the resulting ideas can only be as a possibility of similar action of a malware in a real environment, most of the time, in the second case, the effect being much more intrusive. The most important measures for protecting the analyzed network are:

- The existence of a SIEM system, to be able to capture various network logs and to be able to effectively detect malicious executions or attacks from the internal area of the network.
- Implementation of much stricter access control policies, in order to limit users from easily creating attacks.
- Segmentation of various parts of the network, to prevent the penetration of illegitimate traffic to critical resources.
- Constant updates of antivirus systems, in order to have access to the latest versions of malicious file signatures.
- Constant updates of antivirus policies and systems at the endpoint level, to block the execution of malicious files by users.
- Implementation of secure password policies, which are constantly updated.
- Performing constant backups, in environments different from the usual ones, to reduce the effects of ransomware attacks.
- Performing experiments in virtualized environments, to track the weak characteristics of the network.
- Educating users through concrete examples, to increase awareness of the dangers.
- Verifying, through static and dynamic analysis methods, the malicious files that have penetrated the network, to fully understand their behavior and the network vulnerabilities (if they have penetrated the systems by downloading or through FTP servers and have not been detected by antiviruses or SIEM, a problem is indicated at the level of the policies implemented within these solutions).
- Using, besides SIEM system, additional IDS/IPS solutions.

This study focused on evaluating the ability of OSSIM to record and alert on network and host-based attacks in an EVE-NG environment. Quantitative evaluation of mitigation measures, such as firewall rules, traffic filtering, or SIEM correlation tuning, was not performed and is beyond the scope of this work. But, the most important ones, for all network environments, can be:

- Firewall rules or access control lists (ACLs) – that can block Nmap scans, hping3 floods, SSH brute-force attempts.
- Rate limiting or traffic shaping – that can mitigate TCP/ICMP floods (hping3).
- Segmentation – this kind of mitigation was considered even in the network topology and it's advantage can be the fact that it limits spread of malware and reduces alert noise in SIEM.
- Endpoint firewalls – that can prevent malware scanning other hosts.
- Whitelist trusted hosts – that can reduce the impact of a new malicious host.
- Logging and backup – that can retain logs to analyze attacks, false positives, and potential missed events.

In this work, it was proven that EVE-NG is an efficient test environment, but it has its limitations when the network architecture is more complex or when malicious aspects are introduced, which must be processed in parallel with the usual activities of the network devices.

But, citing again the work [13], no test environment (hardware or software) can contain, include or be impenetrable to any type of malicious software. The goal is to learn in as much detail as possible the characteristics of these files, in order to be able to offer, at any time and under any conditions, the most efficient defense and a significant reduction of their impact.

VII. Conclusion

In this work, the focus was on simulating a computer network similar to a real environment, which included both network devices comparable with those currently in use, as well as workstations with Linux or Windows operating systems. This network was created highlighting the idea of resource segmentation and included a central office and remote branch environment. Thus, five major components of the network were imposed: the datacenter area, which included server resources of various types and the network's SIEM system, the NOC area, which ensures continuous monitoring of resources, and the remote branch area, which are related to the resources used in the datacenter area and which are constantly monitored through the NOC area.

The routing method, as well as the services running at the network level, were not central to this work, the emphasis being placed on the analysis of intrusions and network traffic, in the usual way.

In the experiments carried out, two different types of network traffic analysis were considered: network traffic analysis in the presence of an attack at its level and network traffic analysis in the case of an attack by a malicious file. The malicious files used were downloaded from MalwareBazaar resource and are available online.

The conclusion of these experiments is that, although EVE-NG is a good test environment for network administrators, when various types of experiments similar to the real environment are desired, if attacks are also considered, it becomes quite inefficient. Although it also implements methods based on SIEM systems, which are usually capable of detecting these attacks, experiments show that it can only partially simulate real-world analysis.

However, from the experimental area, one can conclude both benefits in the area of learning with this test environment, as well as the identification of key parameters in the analysis of data

networks. Thus, it is found that network traffic analysis must include several methods and solutions (static analysis, dynamic analysis, monitoring and constant logging through SIEM and analysis with machine learning methods) ensuring redundancy when one method fails. In this sense, if in some cases there is a possibility of duplicating the solutions (main SIEM and backup SIEM) or of performing a more complete analysis, using solutions that, apparently, offer the same results, the composition of a network environment as secure and robust as possible is all the better defined.

This study highlights that EVE-NG provides a secure and repeatable environment for examining malware behavior, offering a practical bridge between theoretical research and hands-on cybersecurity training. By enabling controlled experiments in a virtualized network, it allows both academic researchers and industry professionals to analyze attack patterns safely, supporting reproducibility and experiential learning in cybersecurity practices.

In future work, other types of malicious files will be considered, and the analysis will be performed on other types of operating systems that can be modeled in EVE-NG. Also, research will focus on incorporating Wazuh, to enable real-time behavioral analytics. These enhancements aim to improve threat identification capabilities, allowing the system to dynamically recognize unusual activity and provide timely, actionable insights for cybersecurity monitoring and response. The ultimate goal is to create a robust database containing more recent malware samples, which can be adapted to current security systems and learn as much as possible about malware characteristics, regardless of the test environment used.

REFERENCES

[1] S. U. Shaukat, S. Khan, and S. Parkinson, "A Review on Multi-Step Attack Detection," *IEEE Access*, vol. 13, pp. 161779-161805, Sep., 2025, doi: 10.1109/ACCESS.2025.3607497.

[2] D. Omand, "Social Media Intelligence (SOCMINT)," The Palgrave Handbook of Security, *Risk and Intelligence*, pp. 355–371, Jul., 2017, doi: 10.1057/978-1-137-53675-4_20.

[3] EVE-NG, "The Emulated Virtual Environment for Network, Security and DevOps Professionals, " Accessed Aug., 2025, available: https://www.eve-ng.net/

[4] A. Marefat, A. A. M. Nishar, and A. Ashok, "Text2Net: Transforming Plain-text to a Dynamic Interactive Network Simulation Environment," SoutheastCon 2025, Concord, NC, USA, pp. 625-630, 2025, doi: 10.1109/SoutheastCon56624.2025.10971486.

[5] G. Önal and M. Güven, "Enhancing Dynamic Malware Behavior Analysis Through Novel Windows Events With Machine Learning," *IEEE Access*, vol. 13, pp. 153937-153958, 2025, doi: 10.1109/ACCESS.2025.3604979.

[6] A. K. Roy and A. Kumar Khan, "Performance Degradation in Wireless Mesh Networks via External and Internal Attacks," 2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC), Shillong, India, pp. 258-262, 2019, doi: 10.1109/IESPC.2019.8902374.

[7] Malware Bazaar, "A Repository for Malware Samples," Accessed: Dec. 2, 2024, available: https://bazaar.abuse.ch

[8] Q. Liu, K. Bao, W. U. Hassan, and V. Hagenmeyer, "HADES: Detecting and Investigating Active Directory Attacks via Whole Network Provenance Analytics," IEEE Transactions on Dependable and Secure Computing, 2025, doi: 10.1109/TDSC.2025.3611866.

[9] A. Jony, M. N. Islam and I. H. Sarker, "Unveiling DNS Spoofing Vulnerabilities: An Ethical Examination Within Local Area Networks," 2023 26th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, pp. 1-6, 2023, doi: 10.1109/ICCIT60459.2023.10441649.

[10] A. Jony, M. N. Islam and R. A. Talukder, "A Secure Token-Based Approach for DHCP Client Authentication and Replay Attack Prevention," 2024 27th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, pp. 855-860, 2024, doi: 10.1109/ICCIT64611.2024.11022024.

[11] W. Qin, "Design and Implementation of a Cyber Rang with Emulated Network Security Devices," 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China, pp. 1526-1531, 2023, doi: 10.1109/ICCECT57938.2023.10141361.

[12] Maroš HARAHUS, Matúš ˇCAVOJSKÝ, Gabriel BUGÁR, and Matúš PLEVA, "Interactive Network Learning: An Assessment of EVE-NG Platform in Educational Settings", *Acta Electrotechnica et Informatica*, Vol. 23, No. 3, pp. 3–9, 2023, doi: 10.2478/aei-2023-0011

[13] Vasani Vatsal, Amit Kumar Bairwa, Sandeep Joshi, Anton Pljonkin, Manjit Kaur, and Mohammed Amoon, "Comprehensive Analysis of Advanced Techniques and Vital Tools for Detecting Malware Intrusion," *Electronics* 12 (20), 4299, 2023, https://doi.org/10.3390/electronics12204299

[14] Wireshark Foundation, "Wireshark: Network Protocol Analyzer," accessed: Mar. 6, 2025, available: https://www.wireshark.org/

[15] VirusTotal, "VirusTotal," accessed: Mar. 18, 2025, available: https://www.virustotal.com/gui/home/upload

[16] Google, "Google Rapid Response (GRR) Incident Response Framework," accessed: Mar. 21, 2025, available: https://github.com/google/grr

[17] AT&T Cybersecurity, "OSSIM: Open Source Security Information Management," accessed: Mar. 22, 2025, available: https://cybersecurity.att.com/products/ossim

[18] MalwareBazaar Database, "carolina-happy-mike-friend," accessed: Mar. 23, 2025, available: https://bazaar.abuse.ch/sample/f91d55e43648111bbfc06c0dba5ba80eb90b0ebc2a6dab6691f25aacc5c0ee2e/

[19] MalwareBazaar Database, "uranus-hamper-kilo-bacon," accessed: Mar. 23, 2025, available: https://bazaar.abuse.ch/sample/2d04d802438ae93b095acfbb87cf5760bfaf1bbd300a609d6941a6861bcc68a7/

[20] MalwareBazaar Database, "fanta-carbon-fruit-cold", accessed: Mar. 23, 2025, available: https://bazaar.abuse.ch/sample/a7550088fec6eaeab1fc5effb0dc9f003739c7707b55a462d1e283f0f610da55/

[21] Mandiant FLARE (FireEye), "FakeNet-NG — Next-Generation FakeNet (tool)," accessed: Mar. 23, 2025, available: https://github.com/mandiant/flare-fakenet-ng

[22] Wazuh, "Wazuh — Open Source Security Monitoring," accessed: Mar. 24, 2025, available: https://wazuh.com/

[23] Splunk, "SIEM: Security Information & Event Management Explained" accessed: Oct. 16, 2025, available: https://www.splunk.com/en_us/blog/learn/siem-security-information-event-management.html

[24] IBM, "IBM QRadar", accessed: Oct. 16, 2025, available: https://www.ibm.com/products/qradar