Gustas Skarbalius

Open Source Security Information and Event Management System

Bachelor's thesis

Bachelor of Engineering

Information Technology

2025

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland
University of Applied Sciences

Degree title	Bachelor of Engineering
Author	Gustas Skarbalius
Thesis title	Open-source security information and event management system
Year	2025
Pages	40 pages
Supervisor	Matti Juutilainen

ABSTRACT

Open-source solutions play a vital role in addressing growing cybersecurity challenges and offering cost-effective, flexible, and community-supported tools. However, the application, configuration, and maintenance of such systems often pose significant complexities, requiring tailored approaches to meet the unique needs of organizations and effectively address cybersecurity concerns.

The objective of this thesis was to design and develop an open-source security information and event management (SIEM) system using SecurityOnion, SGUIL, Packet Tracer, and Oracle VM VirtualBox. The study sought to create a system that improves network security and enables continuous monitoring, and, thus, to provide insight into the effective use of open-source cybersecurity tools.

Qualitative methods were used to analyze existing challenges with open-source systems and benchmark the performance and reliability of SecurityOnion and associated tools. The development process included creating a virtualized network environment, configuring SIEM components, and integrating monitoring features to identify, analyze, and respond to security threats. The system was designed with an emphasis on transparency, adaptability, and efficiency to cater to diverse use cases.

The key outcome of this thesis was a fully functional and user-friendly SIEM system capable of detecting and addressing suspicious packets while offering a higher level of network security. By providing a comprehensive user guide and demonstrating the system's capabilities, the thesis underscored the value of open-source solutions in advancing cybersecurity practices and ensuring information security.

Keywords: Security Onion, packet tracer, Oracle VM VirtualBox, SGUIL, SIEM, network security

CONTENTS

1 INTRODUCTION

This section introduces the topic, outlines the problem inherent in costly commercial SIEM systems, and presents the thesis objective of evaluating an open-source alternative to paid security systems. Also, the significance and feasibility of Security Onion and SGUIL within a virtual environment are discussed. Finally, the research methods are summarized and the data sources are introduced.

1.1 Research Objective

With the rapid increase in cybersecurity threats, organizations face the challenge of safeguarding their IT infrastructure without exceeding budgetary constraints. Commercial Security Information and Event Management (SIEM) solutions, such as Splunk or ESET, provide effective monitoring and threat detection capabilities but involve substantial licensing costs and vendor dependency. These limitations restrict accessibility, especially for smaller businesses and educational institutions. Open-source solutions, however, offer a viable alternative by reducing costs and promoting flexibility and customization (Deploy Your Own Open Source SIEM, 2023).

Open-source solutions, however, offer a viable alternative by reducing costs and promoting flexibility and customization. Among these, Security Onion and SGUIL stand out, featuring integrated monitoring, alert management, and threat detection functionalities that are usually reserved for costly commercial platforms. (Deploy Your Own Open Source SIEM, 2023).

This thesis aims to assess the effectiveness, usability, and practicality of an open-source SIEM system comprising Security Onion and SGUIL within an Oracle VM VirtualBox environment. By simulating realistic network threats and evaluating the system’s ability to detect and respond to security events, this thesis seeks to demonstrate that a correctly configured open-source SIEM can offer similar threat detection and log analysis capabilities to commercial solutions without the associated high costs. The basic setup logic used for this virtualized

environment is illustrated in a simplified form in Figure 1, offering a visual approximation of how Security Onion operates within a VirtualBox-managed lab environment.

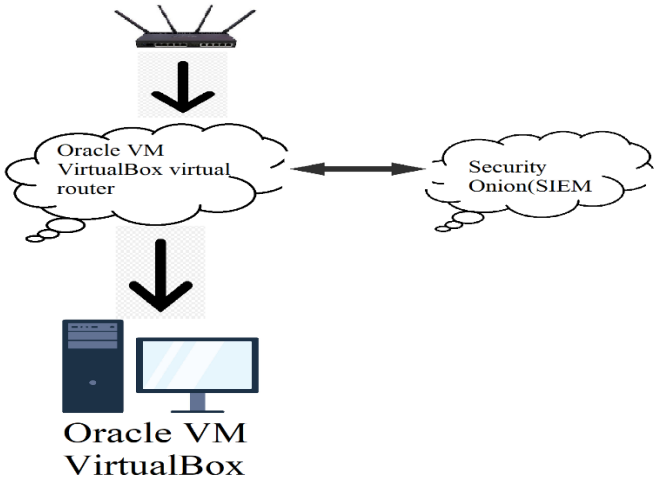

Figure 1. Approximation of SIEM layout

The thesis encompasses practical setup documentation, system configuration, performance evaluation, and risk assessment to provide comprehensive insights and recommendations. Ultimately, this thesis aims to validate that open-source SIEM tools can deliver robust, accessible, and cost-effective cybersecurity capabilities suitable for resource-constrained environments.

1.2 Research Materials and Methods

This thesis prioritizes accessible, adaptable, and transparent experimental approaches, focusing on tools such as Security Onion, SGUIL, Oracle VM VirtualBox, and Cisco Packet Tracer. Security Onion provides comprehensive capabilities, including intrusion detection, log management, and anomaly detection, using integrated components such as Suricata and Elasticsearch. SGUIL complements this by offering detailed event visualization and alert analysis capabilities.

Oracle VM VirtualBox creates a controlled environment, facilitating secure and iterative experimentation without real-world risks. Complementing this setup, Cisco Packet Tracer visually aids in the theoretical mapping of network

structures, enhancing practical validation in the virtual environment. The methodology adopted is qualitative and exploratory, characterized by iterative phases of configuration, simulation, and evaluation, allowing dynamic adaptation based on observed results and practical insights.

In sum, this study encompasses identifying technical requirements, justifying tool selection, establishing a virtualized lab environment, configuring and calibrating monitoring tools, conducting simulations of benign and malicious network activity, analyzing outcomes, and documenting findings comprehensively. This approach mirrors real-world cybersecurity operations where adaptability and iterative refinement are essential for managing evolving threats. Thus, this thesis highlights the perspectives of usability flexibility, and responsiveness with reference to modern cybersecurity design.

1.3 Risk Assessment

Assessing the deployment and operation risks associated with open-source SIEM systems is crucial to understanding their effectiveness and sustainability. Technical risks include concerns about compatibility, false positives, and resource demands, particularly within virtualized setups.

Due to manual configurations and community-driven documentation, open-source tools may pose additional setup risks and potential security misconfigurations.

Operational risks encompass human error, training gaps, and inconsistent monitoring practices. The complexity inherent in Security Onion and SGUIL requires knowledgeable personnel adept in Linux systems and intrusion detection systems, increasing the likelihood of errors or overlooked security threats.

Environmental risks relate to organizational changes, network adjustments (example, topology changes requiring reconfiguration of Security Onion’s sensor nodes), or shifts in external threat landscapes that could render the system less effective. Without consistent institutional support and regular updates, an open-

source SIEM could quickly become obsolete. (Deploy Your Own Open Source SIEM, 2023; A Review on Security Onion Tools for Intrusion Detection, 2021.)

By acknowledging these risks, the assessment made as part of this study aims to provide a balanced perspective on deploying open-source SIEMs, emphasizing the need for robust documentation, regular training, and meticulous system management to mitigate identified challenges.

2 THEORETICAL FRAMEWORK

This section outlines the theoretical basis for the design and implementation of an open-source Security Information and Event Management (SIEM) system. The framework aims to connect foundational cybersecurity concepts to the specific aims of configuring and testing an open-source SIEM setup. While this system is applied in practice it must be grounded in clearly defined principles that explain and justify the use of specific tools and configurations.

The framework is established upon key concepts that support the selection and deployment of tools such as SecurityOnion, SGUIL, Oracle VM VirtualBox, and Packet Tracer. Each of these tools operates within a larger conceptual space involving network monitoring, data visualization, intrusion detection, and virtualization.

Moreover, this section explains how the thesis contributes to ongoing academic and practical conversations about the strengths and challenges of open-source cybersecurity technologies.

2.1 Conceptual Basis and Technologies

The conceptual basis of this thesis is based on the basic principles of network security, digital forensics, and systems engineering. SIEM systems are founded on the idea of a central security hub that gathers and correlates data from diverse sources—including log files, network traffic captures, and intrusion alerts—to identify unusual or dangerous activity across a network. In essence, a SIEM

merges the capabilities of Security Information Management (SIM) (long-term data storage and analysis) with Security Event Management (SEM) (real-time monitoring and alerting). By combining these two functions, a SIEM enables both historical analysis and live oversight of network events, much like a command center monitoring both recorded footage and live camera feeds. This dual approach greatly boosts situational awareness, shortens incidence and improves the traceability of security events by providing a consolidated view of an organization's security posture. (What is SIEM? n.d.)

Security Onion exemplifies this theoretical approach by integrating multiple open-source applications that each handle a layer of security: intrusion detection (e.g. Suricata), detailed traffic analysis (e.g. Zeek), and log management (e.g. Elastic Stack). This layered security design improves reliability and early threat detection by ensuring that if one tool misses a threat, another can detect it. SGUIL complements Security Onion by providing a user-friendly graphical interface to visualize alerts and analyze raw data in context. In other words, SGUIL acts as an analyst's dashboard, consolidating complex network data into readable alerts and summaries, which simplifies the inherently complex task of interpreting network traffic patterns. (Deploy Your Own Open Source SIEM, 2023.)

From a technological perspective, virtualization using Oracle VM VirtualBox provides a secure, isolated environment for controlled cybersecurity experimentation. By running servers and networks inside virtual machines, testers are protected from real-world consequences—any mistakes or attacks are contained within the virtual lab and can be reverted easily using snapshots. This aligns with virtualization theory which states that isolating systems in software containers enables rapid recovery from errors or cyber-attacks and encourages experimentation without harming actual infrastructure. Cisco Packet Tracer complements this setup by allowing users to visually model and "dry-run" the network design. In practice, before configuring the real virtual machines, the network topology can be drawn and tested in Packet Tracer to validate IP addressing, routing, and network layouts. This theoretical pre-validation of

configurations helps detect design issues early before deploying the setup in VirtualBox. (Oracle VM VirtualBox Overview, n.d.)

The theoretical framework of this study incorporates core open-source principles: transparency (the source code and processes are open for inspection), community collaboration (many contributors improving the tools), and rapid vulnerability management (security issues are identified and patched quickly by the community). While open-source tools might lack some of the user interface polish or turnkey convenience of commercial products, they offer greater flexibility and can achieve comparable functionality when configured correctly. In summary, the key concepts in this thesis comprise effective data correlation across sources, a layered defense strategy, human-centered usability (ensuring the tools can be used and understood by people in training), controlled experimentation in isolated environments, and the transparency afforded by open-source development. These principles form the conceptual basis that links the chosen tools to the thesis objectives.

This thesis acknowledges established concepts such as layered monitoring and event correlation, and implemented in a realistic, accessible context. For instance, the use of virtual machines offers an accessible alternative for learners lacking access to expensive infrastructure, translating virtualization theory into tangible educational experiences.

Similarly, while SIEM is primarily used in enterprise-scale deployments, its practical value becomes evident through free-of-charge, open-source components such as Security Onion and SGUIL, demonstrating robust capabilities suitable for small-scale environments. Adopting open-source tools embodies an ideology of accessibility, removing barriers imposed by proprietary licenses and corporate control. Ultimately, this emphasizes the significance of facilitating cybersecurity learning without extensive resources.

It is crucial to acknowledge the theoretical constraints and limitations inherent to open-source SIEM solutions. Scalability issues arise because open-source

systems, although effective in smaller environments, can face performance challenges in large-scale deployments without extensive customization. Reliance on community-based support presents another limitation, as open-source models do not always guarantee the availability of assistance, posing potential risks in security-critical contexts.

Additionally, the complexity and steep learning curves of open-source systems represent a significant practical challenge. While theoretically accessible, these tools typically demand considerable technical proficiency, creating usability gaps, particularly for novices or small organizations. Tool fragmentation further complicates open-source SIEM deployments, as manual integration of separate tools introduces risks of inconsistency and configuration errors.

Lastly, the absence of formal validation or standardized certifications can create challenges for institutions with strict compliance requirements. Even if open-source SIEM solutions are effective in practice, not having recognized certifications may hinder their adoption in regulated settings. Recognizing these limitations ensures a balanced, realistic perspective on the potential and boundaries of open-source SIEM systems.

2.2 Legal, Organizational, and Standardization Context

Implementing cybersecurity systems—especially those that monitor and log data—requires careful consideration of legal, regulatory, and organizational standards. Even though this thesis uses only simulated data in a lab environment, it is still framed by real-world ethical and legal requirements to underscore the importance of practice. In a real-world setting, SIEM deployments must comply with data protection laws and respect privacy, so this thesis considers these aspects as guiding parameters despite the use of simulated data.

Legal considerations include compliance with national and international data protection laws, most notably in the EU such as General Data Protection Regulation (GDPR) and Lithuania’s national cybersecurity and data protection

statutes. These laws emphasize proper data management practices, secure storage of sensitive information, and strict protection of user privacy. In the field of cybersecurity, it is always recommended to acknowledge these principles. For instance, any logging of personal data must be secured and possibly anonymized to comply with GDPR's privacy requirements. Additionally, all software tools employed in this study (Security Onion, SGUIL, Oracle VirtualBox, and Packet Tracer) come with their own licensing terms. Intellectual property rights dictate that these tools must be used in accordance with their licenses, and that proper credit and acknowledgment be given. (GDPR, 2016; Lithuanian Cybersecurity Law, 2018.)
Adherence to these prinpciples ensures that the thesis not only meets the legal requirements but also demonstrates ethical software use.

In addition to laws, adherence to well-known organizational security standards promotes the credibility of the thesis. A comprehensive compliance with a standard such as ISO/IEC 27001 which covers all aspects of information security management is beyond the scope of the thesis, but the study aims to comply with its key principles. For example, the implemented user account setup follows basic access control principles, and the procedure for examining alerts incorporates elements of incident response protocols—both are central tenets of ISO 27001. In addition, the study acknowledges guidance from the U.S. NIST Cybersecurity Framework. In particular, the "Detect" and "Respond" functions of the NIST Framework correspond to core SIEM functions, such as to detect threats in real time and to enable response through investigation of alerts. (International Organization for Standardization, ISO/IEC 27001, 2022; NIST Cybersecurity Framework, 2018). By referencing these standards, the thesis aims to ensure that the open-source SIEM implementation is conceptually in line with industry best practices.

Finally, ethical responsibility is carefully followed in this thesis. All experiments are conducted in a controlled environment isolated from any real networks, ensuring that no actual users or systems are put at risk. This sandbox approach promotes responsible cybersecurity research practices. When simulating attacks

(such as malware or scans), experiments are done in a contained manner. By adhering to ethical guidelines—such as not exporting any malicious traffic to the internet and obtaining proper authorization for all test activities—the study stays within the boundaries of acceptable practice. This ethical stance shows that one can explore and learn about cyber attacks and defenses hands-on without crossing into unethical or illegal territory.

2.3 The Role of Open-Source SIEM in Practical Cybersecurity Training

Open-source SIEM systems offer more than just threat detection tools; they double as valuable hands-on learning platforms that bridge theoretical knowledge and practical skills. Security Onion and SGUIL, in particular, illustrate this educational value. These tools allow learners to engage directly with realistic security operations—such as monitoring network traffic and investigating alerts—within a controlled setting. All the while, students benefit from the flexibility of open-source software which can be modified or extended to suit learning goals and the extensive support of the open-source community (documentation, user forums, and shared configurations). In short, using Security Onion and SGUIL in a lab help to simulate textbook concepts, reinforcing classroom lessons with interactive experience. (What Is SIEM?)

A major educational advantage of open-source SIEMs is their accessibility and low cost. Proprietary SIEM solutions such as Splunk or QRadar often entail expensive licenses or hardware appliances which can put them out of reach for universities or small training programs. In contrast, open-source tools can be downloaded and used freely. This cost-effectiveness means that even institutions with tight budgets can give students direct experience with professional-level security software. For example, instead of merely reading about the operation of a SIEM, students can effectively deploy Security Onion at no cost and experiment with it firsthand. This lowers the threshold for practical cybersecurity training. (Mohamed, S., Yunos, Z., & Razak, M. F. A., 2022.)

Open-source SIEM platforms also encourage experimentation and active learning. The customizable architecture of Security Onion and SGUIL allows students and instructors to adjust configurations, simulate various cyber threats, and test specific defensive responses with minimal restrictions. This flexibility promotes engagement by letting learners ask “what if?” and then actually see the consequences. For instance, an instructor can challenge students to modify an intrusion detection rule in Security Onion and observe how it changes the alerts generated during a simulated attack. Such hands-on trials help students understand cybersecurity concepts because they can observe the cause and effect in real time.

Another advantage of open-source SIEMs is the extensive community support available through documentation, forums, and knowledge-sharing platforms. Novices can easily find guides, tutorials, and discussions addressing common issues. For students, this means that solutions to problems encountered during lab exercises often may already be available online. Access to real-life scenarios and troubleshooting tips from experienced professionals turns the learning process into a collaborative experience. In a classroom setting, learners can use online forums to ask questions and receive answers concerning a Security Onion configuration issue—much as a junior analyst in an organization would consult more experienced colleagues. This community-driven support not only helps overcome the initial learning curve but also mirrors the collaborative problem-solving practices of professional cybersecurity teams.

Using open-source SIEM tools in an educational lab adds a high degree of realism to training. Students can simulate authentic cybersecurity scenarios—ranging from network intrusions and malware outbreaks to insider misuse of credentials—within virtual labs. Responding to these simulated incidents using Security Onion and SGUIL provides useful experience. It allows learners to practice standard operational workflows such as detecting an alert, investigating it via logs and packet captures, and documenting the incident in a stress-free environment. Such realistic simulations very likely prepare students for professional cybersecurity roles better than abstract exercises and help them

recognize common attack patterns and learn systematic responses, skills that are difficult to master by attending or reviewing the documentation of case studies.

An often-overlooked benefit of deploying SIEM tools in training is the development of soft skills such as problem-solving, critical thinking, and teamwork. Setting up and using Security Onion with SGUIL is not a rote task—it requires troubleshooting configuration issues, interpreting ambiguous alerts, and prioritizing responses to multiple concurrent events. These challenges closely resemble real-life situations where security analysts must work together to solve problems under time pressure. As students navigate such hands-on difficulties in the lab, for example, trying to identify why logs are not appearing as expected, or collaborating to trace a simulated attack across systems, they can sharpen their analytical thinking and learn to communicate effectively with team members. This helps them become more versatile and employable after graduation.

Open-source SIEM projects inherently support interdisciplinary learning. By working through a SIEM deployment, students engage with multiple domains: networking (configuring network interfaces and understanding traffic flow), system administration (installing and managing Linux servers and services), and software development/scripting (adjusting configuration files or parsing log data). Furthermore, discussions among students naturally extend to legal and ethical considerations—for instance, who is allowed to view certain logs and how long data should be retained under privacy laws. This allows students to improve their technical competence while also learning to see the bigger picture: concerning such as data privacy, ethical hacking limits, and legal implications of monitoring, which are crucial topics in real-world cybersecurity jobs (Mohamed, Yunos & Razak, 2022.)

Educational institutions also benefit from using flexible open-source tools. The cybersecurity landscape evolves rapidly, with new threats and attack techniques emerging every semester, and open-source communities tend to respond quickly to these changes by patching vulnerabilities and adding detection rules often within days of a new threat being identified. In contrast, proprietary solutions

might have slower update cycles or require costly upgrades. By leveraging open-source SIEM tools, instructors can keep their curriculum relevant without waiting for vendor updates. For example, when a new type of malware appears, instructions using Security Onion can update the system's rules or add a community-created detection script immediately. This agility ensures that students can work on up-to-date scenarios and learn the latest practices, rather than working with outdated examples (Mohamed, Yunos & Razak, 2022.)

In summary, incorporating tools such as Security Onion and SGUIL into academic programs can greatly contribute to cybersecurity training. They offer affordable access to enterprise-grade technology, realistic simulations of cyber incidents, and adaptable environments for a range of exercises. Supported by vibrant user communities, these open-source platforms help to that learners are not in isolation but can benefit from collective knowledge and contribute with own findings. Overall, these tools prepare students with essential practical skills, from configuring systems to analyzing attacks, and give them a broader view of industry practices and challenges. This combination of hands-on skill-building with community-supported knowledge is why open-source SIEMs are increasingly favored in cybersecurity training (Deployment of Honeypot and SIEM Tools for Cyber Security Education Model In UITM)(Mohamed, Yunos & Razak, 2022.)

2.4 Review of Related Research

In order to situate this thesis within the broader cybersecurity landscape, it is crucial to examine related academic works, practical implementations, and case studies focusing on open-source SIEM systems and cybersecurity education. This section identifies relevant themes and illustrates how this thesis aligns with current research efforts while addressing specific gaps.

Numerous academic studies and technical projects have explored deploying open-source SIEM tools (including Security Onion, Suricata, Zeek, and the ELK stack) in laboratory and institutional settings. It has been shown that Security Onion can perform on par with commercial alternatives under controlled

conditions, provided it is properly configured and maintained, although it may sometimes require additional computing resources or fine-tuning. This is an important finding: it suggests that organizations on a limited budget can still achieve robust threat detection and log analysis capabilities without purchasing expensive licenses. These prior results strongly support the rationale for the approach of this thesis for choosing free, community-supported software as the backbone of a SIEM implementation. (Open Source Software Security, 2019.)

SGUIL's functionality and its usefulness in network security monitoring have been specifically examined. For example, (Bejtlich, [2004]) details SGUIL's interface and capabilities, concluding that SGUIL adds significant value for network analysis training. These studies highlight how SGUIL's design—integrating alert data with detailed packet logs—makes it well suited for hands-on cybersecurity education. In practice, this means students can use SGUIL to pivot from an alert directly into contextual data (such as viewing the raw packets that triggered an intrusion alert), which is an effective way to learn incident investigation. Such discussion seems to confirm that SGUIL is not only a powerful tool for professionals but also an excellent teaching instrument for network security monitoring concepts.

Virtualization technologies (notably Oracle VM VirtualBox) are now a staple in cybersecurity research and training because they allow safe, realistic network simulations without requiring physical hardware. When virtualization is combined with network visualization tools such as Cisco Packet Tracer, the educational benefits further increase. Packet Tracer provides a clear diagrammatic view of the network topology and data flows, which helps learners conceptualize what the VirtualBox-hosted SIEM environment is doing behind the scenes. In this thesis, that same best practice was adopted by using VirtualBox to host the operational SIEM system and Packet Tracer to design and explain the network. This combination not only mirrors methods used in other research and training programs but also maximizes the learning value of the thesis implementation. (Oracle VM VirtualBox Overview, n.d.)

In recent years, cybersecurity educators have launched various initiatives aimed at improving practical learning and comprehension. Programs such as CyberPatriot (a national youth cyber-defense competition in the US) and Youth4Cyber (a European initiative focused on youth cybersecurity awareness) exemplify this trend. They use scenarios and tools—often built on open-source platforms—to simulate real-world cybersecurity incidents in a competition or lab format. While those programs are large in scale and formally structured, this thesis shares a similar philosophy on a smaller scale and the thesis's approach is aligned with the idea of broadening access to cybersecurity education: making hands-on learning accessible to anyone possessing a computer, by using free tools and virtual environments. In essence, the thesis echoes what these initiatives demonstrate – that effective cybersecurity training does not always require enterprise-grade infrastructure or costly software, simply a thoughtful integration of open-source resources and practical scenarios. (What is CyberPatriot? N.d., ECSO Youth4Cyber 2016)

Despite the growing popularity of open-source SIEM tools in education institutions and industry, there is still a lack of comprehensive, step-by-step documentation describing the building of a fully integrated and easily replicable SIEM system using exclusively free tools. In many cases, practitioners or researchers have to piece together information from various sources to have an end-to-end solution. This thesis addresses this shortcoming by providing detailed documentation of a small-scale SIEM deployment. The emphasis throughout is on replicability and accessibility. In other words, this thesis aim to server a manual for recreating such system. By openly presenting the exact configuration steps, network design, and lessons learned, this thesis aims to lower the barrier for students, educators, or small organizations interested in implementing their own affordable SIEM lab.

It should be clarified that the approach in this thesis is practical and educational, rather than theoretical. This study does not aim to introduce new detection algorithms or innovative security frameworks. Instead, its value lies in offering a structured example that combines relevant theoretical concepts with practical

configuration steps and pedagogical considerations. The thesis builds on existing research and implementation practices, presenting them in a format that may be more accessible and applicable to educational or small-scale settings. While not intended as a definitive model, the outcome illustrates one possible approach to deploying open-source SIEM systems in a virtual lab environment. By documenting both the actions taken and the rationale behind key decisions, this thesis may provide a helpful point of reference for those seeking to implement similar solutions for learning or introductory monitoring purposes.

3 DESIGN AND IMPLEMENTATION OF THE SYSTEM

This section focuses on the practical creation and structure of the open-source Security Information and Event Management (SIEM) system developed for this thesis. It outlines the technical steps, design logic, and implementation process from conceptual planning through to final system configuration. The objective is to demonstrate not only how the system was built, but also why specific decisions were made—always taking into account accessibility, functionality, and pedagogical clarity.

The design relies on four core tools:

- **SecurityOnion** – the backbone of the SIEM system, responsible for data collection, traffic analysis, and alerting.
- **SGUIL** – the graphical interface used to analyze and visualize alerts, offering user-friendly interaction with network events.
- **Oracle VM VirtualBox** – a virtualization platform used to simulate the test environment.
- **Cisco Packet Tracer** – a network simulation tool and used to design and visualize the logical network topology and flow of data.

These tools were selected based on their availability, community support, and practical relevance to small- and medium-scale cybersecurity needs.

3.1 Comparison of SIEM System Interfaces (ESET, Splunk, Graylog, ELK Stack, and SGUIL)

In selecting a suitable SIEM interface, ESET, Splunk, Graylog, ELK Stack, and SGUIL were analyzed across cost, source openness, intended uses, usability, and overall advantages. Each interface has distinct characteristics relevant to various organizational needs (Table 1). To better understand these differences, each interface is introduced in detail below.

ESET a proprietary SIEM solution known for strong monitoring and threat detection capabilities. ESET is designed to integrate seamlessly with the rest of the ESET security product ecosystem. (About ESET). In practice, this means an organization already using ESET antivirus or endpoint protection might find value in its SIEM for centralized management. However, ESET's advantages come at a cost. Its licensing fees are high, and it offers limited flexibility for customization compared to open-source alternatives. These factors can be significant drawbacks for smaller organizations or educational labs which often cannot afford ongoing license costs or need more adaptable systems. The typical ESET management interface showcasing its dashboard and alert view is presented in Figure 7 (Appendix 1).

Splunk is another leading commercial SIEM, Splunk is renowned for its powerful log indexing engine, advanced analytics, and automation capabilities in incident response. Splunk can ingest massive volumes of data and automatically correlate events, making it a viable choice for large enterprises that need to detect sophisticated threats across complex environments. Despite its effectiveness, Splunk shares the accessibility issues of other commercial tools: it is expensive to license and maintain. The high cost, along with the system's complexity (which often requires trained Splunk engineers or administrators), puts it out of reach for many smaller organizations and most educational settings. Figure 8 (Appendix 1) shows a Splunk interface example, illustrating the detailed dashboards and search query capabilities that Splunk provides. (What is Splunk?)

Graylog is an open-source log management platform noted for its powerful search and filtering capabilities and its extensible API for integrations. In its free community edition, Graylog allows organizations to collect and analyze log data without licensing costs (a paid enterprise edition adds more features and support). This makes Graylog an attractive starting point for those who need centralized log analysis on a budget. However, Graylog is primarily a general log management tool rather than a specialized security incident platform. It lacks built-in threat intelligence or advanced correlation rules that dedicated SIEMs have. In a pure security monitoring context, this means Graylog might require additional plugins or custom rules to reach the same level of real-time alerting. Figure 9 (Appendix 1) shows the Graylog dashboard interface, which emphasizes log data streams and search filters, reflecting Graylog's focus on broad log analysis rather than security-specific features. (What is Graylog?)

The ELK Stack (Elastic Stack)—comprising Elasticsearch, Logstash, and Kibana (often now called the Elastic Stack with the addition of Beats)—is a very popular open-source solution for log and event data analysis. Its strength lies in broad data visualization and search: logs from virtually any source can be ingested (via Logstash or Beats), stored and indexed (in Elasticsearch), and visualized on rich dashboards (via Kibana). This modular design is extremely powerful and flexible, capable of scaling to handle large datasets and diverse data types (system logs, application logs, network events, etc.). However, with great power comes complexity. Standing up an ELK Stack for security monitoring requires substantial expertise in each component and careful tuning to avoid being overwhelmed by data. Maintaining performance and useful signal-to-noise ratio in ELK can be challenging without dedicated effort. Figure 10 (Appendix 2) provides a Kibana dashboard from an ELK Stack deployment, demonstrating complex visualizations and search queries, and indicating the configuration effort required. (What is Elastic-Stack?)

Table 1. SIEM Comparisons

SIEM System	Cost	Open Source	Primary Uses	Usability	Major Advantages
ESET	$60/year per device	No	Comprehensive Security Monitoring	High usability, easy integration with ESET products	Integrated ecosystem, strong security features
Splunk	Unpublished/Variable	No	Extensive Log Analysis & Incident Response	Complex but powerful analytics interface	Advanced automated analytics, enterprise-level capabilities
Graylog	Free (basic), Paid (advanced)	Partial	General Log Management & Analysis	High usability, intuitive interface	Flexible API, limited in open-source form
ELK Stack	Demo (Free), Paid Tiers	Partial	Broad Data Visualization & Analytics	Highly flexible, moderate complexity	Modular structure, scalable, requires expertise
SGUIL	Free	Yes	Specialized Real-time Network Security Monitoring	User-friendly, straightforward integration	Real-time alerting, best fit based on personal experience with SGUIL

SGUIL Integrated as part of Security Onion, SGUIL (pronounced “squeal”) is a specialized open-source interface for real-time network security monitoring. Unlike the more generalized tools above, SGUIL is purpose-built for security analysts, providing consoles to review intrusion detection alerts alongside the raw packet data related to those alerts. Being open-source and cost-free, SGUIL can be installed easily as part of Security Onion without any licensing barriers. Its user interface, while more rudimentary than such as Splunk or Kibana, is

straightforward and focused on security events, which reduces complexity for beginners. These traits make SGUIL especially well-suited for educational use and small-scale operations. Students and instructors can quickly can be quickly deployed and immediately start observing real-time alerts, for example, from Suricata and investigate those alerts by conducting in-depth analysis into packet captures. In section 3.5 Figure 5 illustrates the SGUIL interface in action, displaying network events and alerts in real-time. This dedicated focus on network security data—rather than general logs—is what sets SGUIL apart other interfaces. (Network Security Monitoring with Sguil, 2004.)

Based on this comparative evaluation (Table 1), SGUIL emerged as the most advantageous selection, offering notable cost-effectiveness, targeted security monitoring capabilities, ease of integration, and accessibility.

3.2 Planning the Architecture

The first stage of the implementation was to design a simplified yet functionally relevant network architecture that would serve the needs of a small organization or an educational lab. The design had to be practical – including only essential components – and representative of typical scenarios in which a SIEM might be deployed for a small enterprise or training environment.

Cisco Packet Tracer was used to create a detailed visual mock-up of the intended network. This simulated network diagram included common components: a few workstations (to represent end-user computers), a dedicated server, a router to handle internal traffic routing, and a firewall to enforce security policies at the network's edge. The Packet Tracer diagram served as both a conceptual guide and a practical blueprint for the subsequent VirtualBox setup. By designing the network on Packet Tracer first, ensured that when it came time to configure the virtual machines, there was a clear reference for how all network devices and connections should be connected. Figure 2 presents an early prototype of the network layout created in Packet Tracer. While this sketch is a simplification and not every detail was exactly replicated, it was invaluable in

shaping the full implementation by providing an initial vision of the lab architecture.

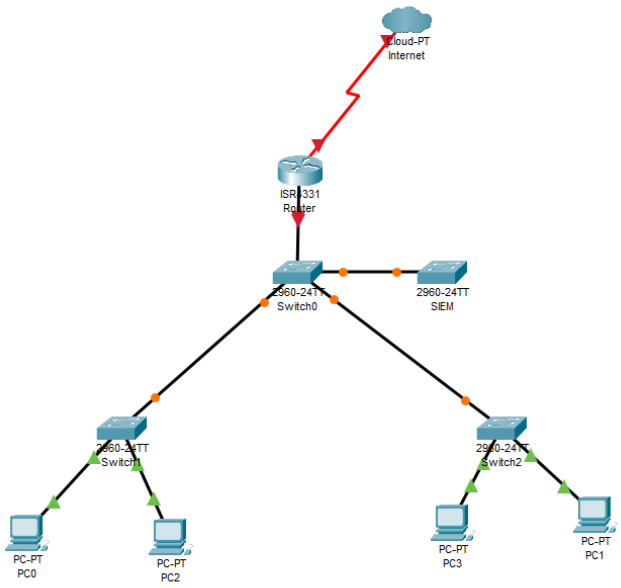

Figure 2. First sketch/prototype of SIEM Architecture

The decision to keep the network architecture intentionally minimalist was strategic. The decision mirrors the operational constraints often faced by small organizations or academic institutions that typically have limited hardware resources and a small IT team. In this design, all core services and hosts are contained within a single, clearly defined virtual environment. This setup strikes a balance between simplicity and essential functionality. By avoiding an overly complex network, potential points of failure are reduced and the system is made more robust and much easier to understand and manage. At the same time, the design still provides all essential capabilities for effective monitoring included components such as clients, a server, and a firewall are included to simulate real network traffic and security events. This streamlined approach also aids replicability: the lab can be recreated without specialized hardware or an intricate configuration, which is ideal for teaching scenarios.

Despite its simplicity, the design is sufficiently robust to support a variety of comprehensive testing scenarios. It is possible to capture and analyze network traffic and trigger security alerts, analyze by simulating attacks, and monitor data

flows between devices in real time. Importantly, all of this takes place in a fully simulated network sandbox. Which means none of the potentially malicious activities such as port scans or malware simulations escape into any external network – a critical safety factor. This closed-loop setup allowed the conduction of a thorough experimentation and learning process a controlled environment. Such an approach is ideal for educational purposes and entry-level security training because students can observe the effects of cyber threats and defensive measures firsthand without endangering any real systems.

3.3 Setting Up the Virtual Environment

In order to host all virtual machines used in this study, Oracle VM VirtualBox was employed due to its flexibility, reliability, and widespread acceptance within both educational and professional environments. The selected virtual environment comprised:

- A primary virtual machine running SecurityOnion, based on the widely used Ubuntu Linux distribution.
- Two supplementary virtual machines acting as typical client endpoints, running on standard Windows and Linux operating systems.

This mix of one SIEM server and two client VMs provides a simple but effective test network, with multiple operating systems for diversity of traffic. Windows and Linux systems tend to produce different log and traffic patterns, which is useful for realistic testing. This also mirrors common small network setups of one server and a few clients.

Networking between the virtual machines was configured using VirtualBox Host-Only Adapter networks. Figure 3 illustrates the adapter settings in the configuration. This setup provides a critical layer of isolation: all the VM network traffic is confined to a private virtual network that includes the host machine, and no traffic is routed out to the physical internet or the broader LAN. In practical terms, this means the Security Onion VM and the two client VMs can communicate with each other (for example, clients send traffic that the Security Onion sensor detects) but they cannot reach any external servers or be reached from the system. This isolation is crucial for security experimentation as it

guarantees that even if a simulated malware or attack tries to spread, it cannot escape the lab environment. The isolation also ensures that external scans or internet noise do not interfere with controlled tests.

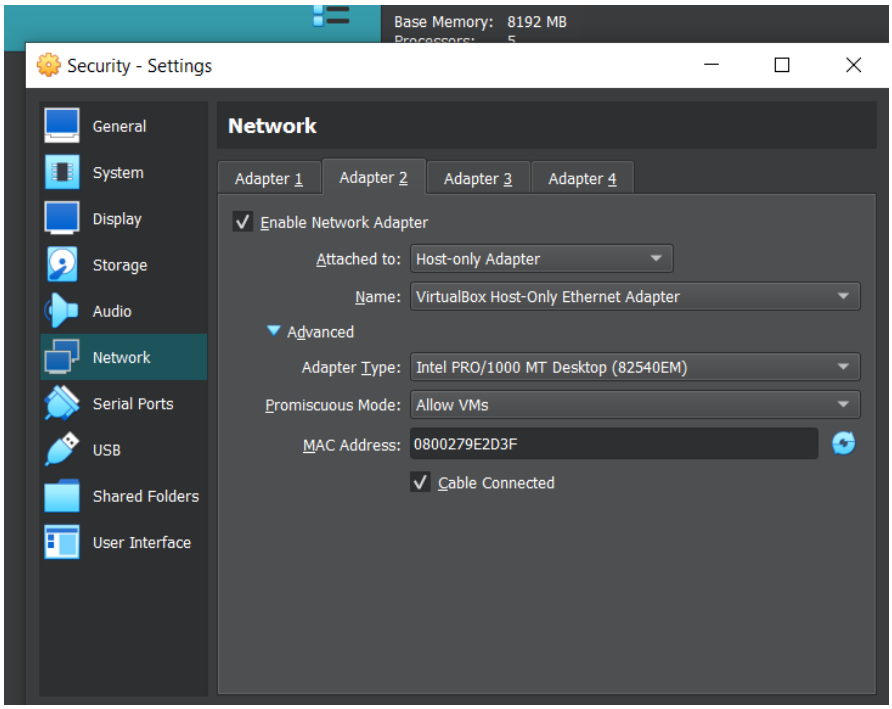

Figure 3. Adapter Settings

As a precaution, regular snapshots were taken of the virtual machines at various points during setup and testing. This snapshot strategy meant that if a configuration change went wrong or an update caused instability, it was possible to quickly revert the VM to a previous stable state. Such an approach significantly enhanced the efficiency of our experimentation. Instead of troubleshooting at length or rebuilding the environment from scratch after a major error, it was possible to roll back in a matter of seconds and resume work. This is a common best-practice in virtual lab implementations because it provides a safety net that encourages learning by trial and error knowing that mistakes are reversible. In this case, it saved time and ensured that minor setbacks did not derail the overall thesis progress.

3.4 Installing and Configuring SecurityOnion

Security Onion was installed on its dedicated virtual machine using the distribution's built-in setup wizard. The installation process is straightforward but time-consuming. It involves steps such as selecting the server role, setting up network interfaces, and choosing which components to enable. Each step required careful attention to detail and instance, it had to be ensured that the correct network interface was designated for monitoring. However, the wizard made the process approachable even for new to SIEM installations. Figure 4 illustrates the Security Onion desktop after installation, confirming the successful setup of the system.

Figure 4. SecurityOnion desktop

The Configuration involved several key steps to ensure comprehensive monitoring and effective management of security operations :

- Defining network interfaces clearly for efficient monitoring and administrative management.
- Selecting and enabling critical security monitoring tools such as Suricata for intrusion detection, Zeek for detailed network traffic analysis, and Wazuh for robust log correlation.

- Intergrating Elasticsearch and Kibana, which provided additional analytical and visualization capabilities to enhance the monitoring experience.

It should be noted that while installing Security Onion was not technically difficult, it did demand patience. After the initial setup, the system performs data indexing, particularly after Elasticsearch is enabled and the indexing of logs, begins, and downloads updates for its rulesets such as those for Suricata and Wazuh. These steps can be time-consuming. During the setup made for this study, there were stretches of several minutes where the system was updating or building indices. However, waiting periods are normal and essential for the platform to achieve optimal stability and responsiveness. Allowing the system to fully update and index before heavy use is considered a best practice as it ensures that all detection rules are up-to-date and that search queries in tools such as Kibana run quickly. Breaks were scheduled and other tasks were performed while these long operations completed, underscoring that an effective setup sometimes requires careful monitoring rather than constant action.

3.5 Integrating SGUIL for Alert Analysis

The SGUIL interface was seamlessly integrated as part of the initial SecurityOnion setup. SGUIL served as a primary graphical tool, enabling detailed analysis and clear visualization of security alerts and raw packet data captured within the network, as shown in Figure 5.

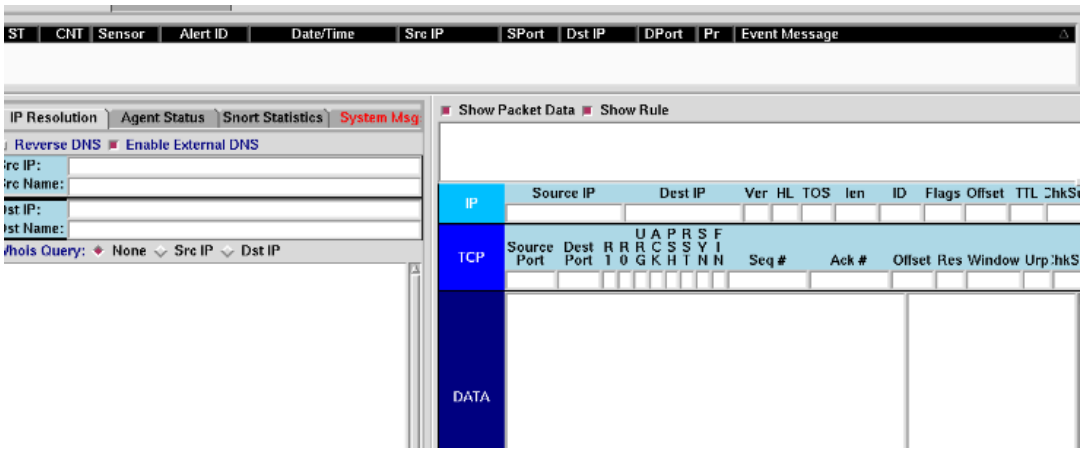

Figure 5. SGUIL interface

The interface's practical effectiveness was assessed through a series of simulated alerts reflecting common security incidents, including:

- Ping floods designed to test response capability.
- Unauthorized port scanning activities.
- Suspicious HTTP network traffic.

These scenarios effectively demonstrated SGUIL's strengths in managing and interpreting security events in real-time, emphasizing its suitability for practical educational exercises and small-scale cybersecurity monitoring. Figure 6 illustrates SGUIL alerts generated during a simulated Trojan virus infection, demonstrating how threats are captured and displayed within the interface.

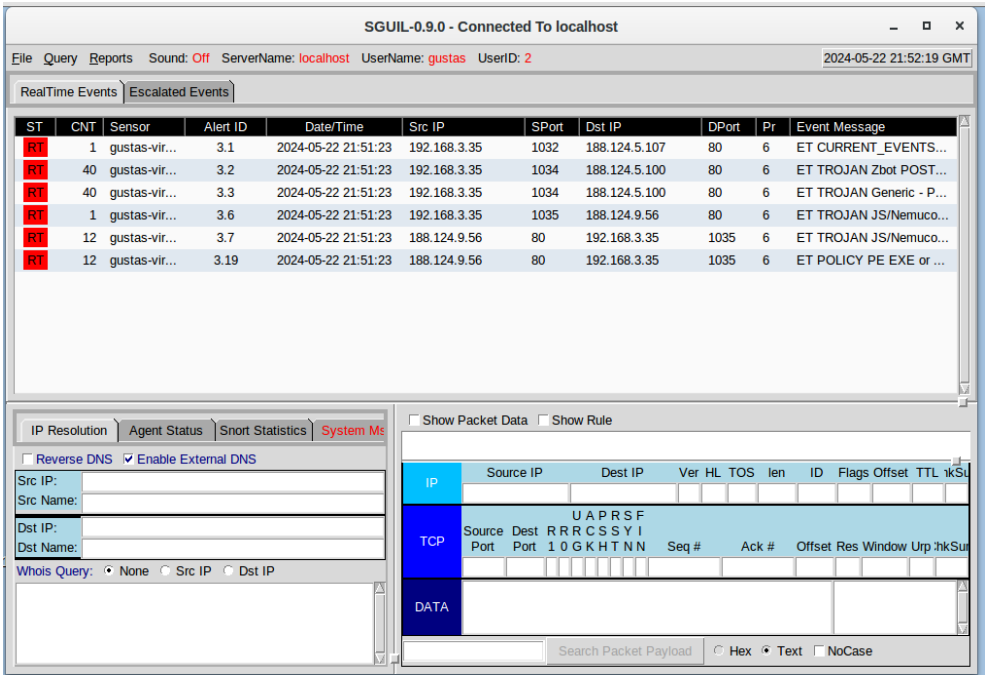

Figure 6. SGUIL Alerts

3.6 Generating and Testing Network Events

The practical testing of the system's response capabilities involved deliberate threat simulations launched from client virtual machines, encompassing typical security threats such as :

- Conducting network scans using commonly available tools such as Nmap.
- Executing DNS spoofing simulations to observe detection accuracy.
- Simulating abnormal and unauthorized login attempts.

These controlled experiments allowed each simulated event to be traced comprehensively through the SIEM processing pipeline. Specifically, threats were detected by Suricata, logged meticulously by Zeek, correlated effectively by Wazuh, and visualized clearly and practically in SGUIL. This full-cycle testing provided a thorough evaluation of the integrated monitoring system's responsiveness and reliability.

3.7 Challenges Encountered

Throughout the thesis, several practical challenges emerged. These issues are summarized as follows :

At times, there were compatibility issues between the host operating system and Oracle VM VirtualBox. A kernel driver error on VirtualBox was encountered following a host operating system update, which prevented the virtual machines from starting. This required troubleshooting steps such as reinstalling VirtualBox drivers and adjusting specific configuration settings. Such hiccups are not uncommon when using virtualization software on various host OS versions. (Microsoft, Dynamic Host Configuration Protocol.)

Occasionally, the virtual machines lost network connectivity with each other. This was observed when the Security Onion virtual machine stopped receiving traffic from the client machines, or when clients were unable to communicate with each other. Each time, the fix was to reset the VirtualBox Host-Only Network adapter or reboot the VMs, which restored connectivity.

SGUIL exhibited noticeable delays when processing multiple simultaneous alerts, highlighting performance constraints related to the system’s simplified architecture.

Despite these challenges, none of them impaired the system’s overall functionality. A stable and practical virtual lab environment was successfully

delivered that effectively demonstrated fundamental principles of security monitoring.

3.8 Summary

In summary, the implemented open-source SIEM system effectively achieved its educational and practical objectives. The final setup was intentionally streamlined, yet it was able to clearly demonstrate core SIEM operations: from real-time traffic analysis and alert generation to basic incident investigation and response. By prioritizing simplicity, ease of replication, and practical utility throughout the design, a lab environment was created. This system serves as a solid introduction to open-source cybersecurity tools by lowering the barrier for students and small organizations to deploy and explore a functioning SIEM. It is particularly well-suited for educational use and training scenarios. For instance, in a classroom or workshop setting, participants can use the same setup to learn how to detect and analyze attacks. Moreover, small-scale security operations such as a startup or nonprofit organization with limited resources could adopt this model to improve their network monitoring on a shoestring budget. The fact that this thesis provides hands-on experience and insight into practical cybersecurity practices highlights the potential value of open-source solutions in similar contexts. The process described in this study demonstrates that, with careful planning, effective network security monitoring can be implemented a modest scale even in environments with limited resources.

4 RESULTS AND DISCUSSION

This section summarizes the outcomes of the experimental part of the study and evaluates them in light of the thesis’s objectives providing a clear and realistic assessment of limitations and implications in the broader context of cybersecurity education and open-source system deployment.

The primary objective of this thesis was to develop and test an open-source Security Information and Event Management (SIEM) system using SecurityOnion, SGUIL, VirtualBox, and Packet Tracer. The system was

successfully built in a virtual lab and responded appropriately to various simulated threats. From installation to alert analysis, every major component of the SIEM pipeline operated as intended.

One of the undisputed results was the system’s ability to detect and log activity using built-in tools such as Suricata and Zeek. The alerts were timely and contextually rich, and SGUIL's interface enabled an efficient interpretation of those alerts. While this confirms that the tools function correctly, it also demonstrates that visibility into network behavior is achievable, even in a simplified lab environment. In addition to detection, the combination of real-time event tracking and historical packet inspection allowed for basic incident reconstruction. This reflects core SIEM functionality and validates the system’s use in entry-level security environments, such as classrooms or small businesses.

Despite the positive outcomes, several limitations emerged during testing. First, system performance was modest; resource consumption occasionally caused delay, especially in SGUIL when multiple alerts were triggered. Second, the learning curve for configuration and integration of components such as Wazuh or Elastic dashboards was steep. Third, certain log entries required manual inspection or additional correlation that was not automated by the system. These observations are similar to the findings made in previous research. Which confirms that open-source systems, while flexible, often require more user intervention and system knowledge compared to commercial solutions.

However, results demonstrate that a functioning SIEM system can be built entirely from free and open-source tools. For institutions or individuals without access to commercial platforms, this is significant. It shows that foundational cybersecurity practices—such as traffic monitoring, alert generation, and incident tracking—can be carried out effectively using tools available to the public. Furthermore, the simplicity of the test network reinforces that such a setup does not need to be overly complex to be educational or functional. Although the thesis does not offer technical innovation, its primary contribution lies in describing

processes that facilitate and replicability. This study presents a functioning example of a cost-efficient and transparent approach to network security.

Building and testing the system reinforced several key lessons. First, even simple setups can offer strong learning outcomes. Virtual environments are sufficient for most baseline testing and experimentation. Second, while open-source solutions are not always easy to implement, they offer a high degree of adaptability. Third, the thesis highlights the importance of critical thinking and practical troubleshooting while aiming to balance theoretical understanding with hands-on experience. Also, it emphasizes the importance of patience and attention to detail, particularly in contexts where multiple tools are required to interact reliably.

In conclusion, the results achieved in this thesis validate the concept and establish a working baseline for future implementations. While it is not a comprehensive enterprise-grade SIEM, it is a fully functional educational tool and a gateway to more thorough exploration in the field cybersecurity.

REFERENCES

Air & Space Forces Association. n.d. What is CyberPatriot – https://www.uscyberpatriot.org/pages/about/what-is-cyberpatriot.aspx

Bejtlich, R. 2004. Network Security Monitoring with Sguil – https://www.bsdcan.org/2004/papers/sguil.pdf

Bejtlich, R. 2004. Tao of Network Security Monitoring, The: Beyond Intrusion Detection – https://www.informit.com/articles/article.aspx?p=350390

Cisco. n.d. What Is SIEM? - Security Information and Event Management. – https://www.cisco.com/c/en/us/products/security/what-is-siem.html

ECSO Youth4Cyber. 2016. – https://ecs-org.eu/activities/youth4cyber/

European Union. 2016. General Data Protection Regulation (GDPR) - https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679

Eset. n.d About ESET - https://www.eset.com/lt/apie/

Fortinet. n.d. Trojan Horse Virus – https://www.fortinet.com/resources/cyberglossary/trojan-horse-virus

Fortinet. n.d. What is Splunk? – https://www.fortinet.com/resources/cyberglossary/what-is-splunk

International Organization for Standardization (ISO). 2022. ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements — https://www.iso.org/standard/27001

Lithuanian Republic. 2018a. Cybersecurity Law of the Republic of Lithuania (Kibernetinio saugumo įstatymas) — https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/TAIS.451177

Lithuanian Republic. 2018b. Republic of Lithuania Law on the Legal Protection of Personal Data (Asmens duomenų teisinės apsaugos įstatymas) — https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/bc0837f27f9511e89188e16a6495e98c

Medium. n.d. What is Graylog? – https://medium.com/@gouthamr102/graylog-5265748cace0

Microsoft. n.d. Dynamic Host Configuration Protocol (DHCP) – https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-top

Mohamed, S., Yunos, Z. & Razak, M. F. A. 2022. Deployment of Honeypot and SIEM Tools for Cyber Security Education Model In UITM – https://www.researchgate.net/publication/365509767_Deployment_of_Honeypot_and_SIEM_Tools_for_Cyber_Security_Education_Model_In_UITM

Nazar, M. 2021. A Review on Security Onion Tools for Intrusion Detection – https://www.researchgate.net/publication/359221249_A_Review_on_Security_Onion_Tools_for_Intrusion_Detection

Kaufman, M. 2017. Network Routing: Algorithms, Protocols, and Architectures – https://web-p-ebscohost-com.db.kaunokolegija.lt/ehost/detail/detail?vid=56&sid=67cbe087-1b97-4363-9683-2c99dadff369%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=1145266&db=e000xww

Mukherjee, A. 2020. Network Security Strategies: Protect Your Network and Enterprise Against Advanced Cybersecurity Attacks and Threats – https://web-p-ebscohost-com.db.kaunokolegija.lt/ehost/detail/detail?vid=26&sid=67cbe087-

1b97-4363-9683-2c99dadff369%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=2648388&db=e000xww

NIST. 2018. NIST Cybersecurity Framework (CSF) – https://www.nist.gov/cyberframework

NIST. 2020. NIST SP 800-53 Rev. 5 – Security and Privacy Controls for Information Systems – https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final

Li, Y., Ma L., Shen, L., Lv, J. 2019. Open Source Software Security Vulnerability Detection Based on Dynamic Behavior Features – https://web-p-ebscohost-com.db.kaunokolegija.lt/ehost/detail/detail?vid=34&sid=67cbe087-1b97-4363-9683-2c99dadff369%40redis&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#AN=138238861&db=asn

Oracle. n.d. Oracle VM VirtualBox Overview – https://www.oracle.com/assets/oracle-vm-virtualbox-overview-2981353.pdf

Security Onion. n.d. Security Onion Introduction – https://docs.securityonion.net/en/2.3/introduction.html

TechTarget. n.d. What is Elastic-Stack? – https://www.techtarget.com/searchitoperations/definition/Elastic-Stack

Premium Papers. 2025. Intrusion Detection System: Sguil's Viability as a Network Security Monitor – https://premium-papers.com/intrusion-detection-system-sguils-viability-as-a-network-security-monitor/

Yadav, B. 2023. Deploy Your Own Open Source SIEM With Security Onion – https://www.linkedin.com/pulse/deploy-your-own-open-source-siem-security-onion-bise-

SIEM Interfaces:

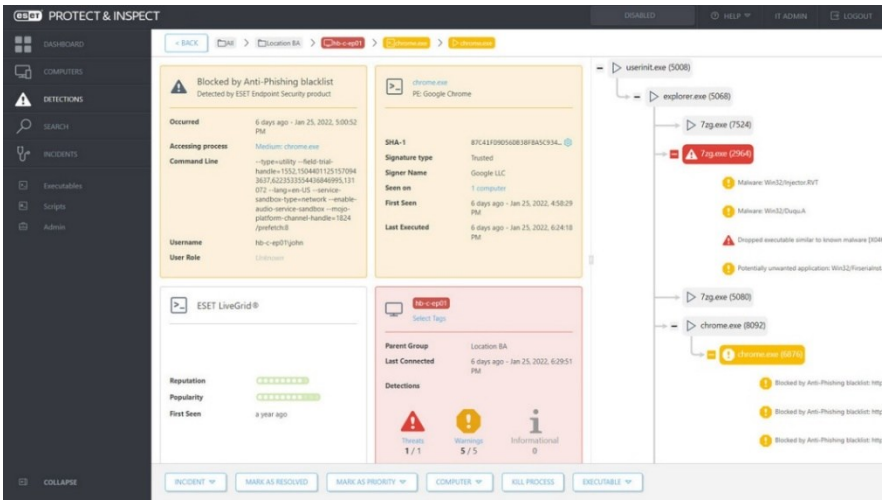

Figure 7. ESET Interface https://www.eset.com/my/business/solutions/security-management/

Figure 8. SPLUNK Interface https://www.splunk.com/en_us/products.html

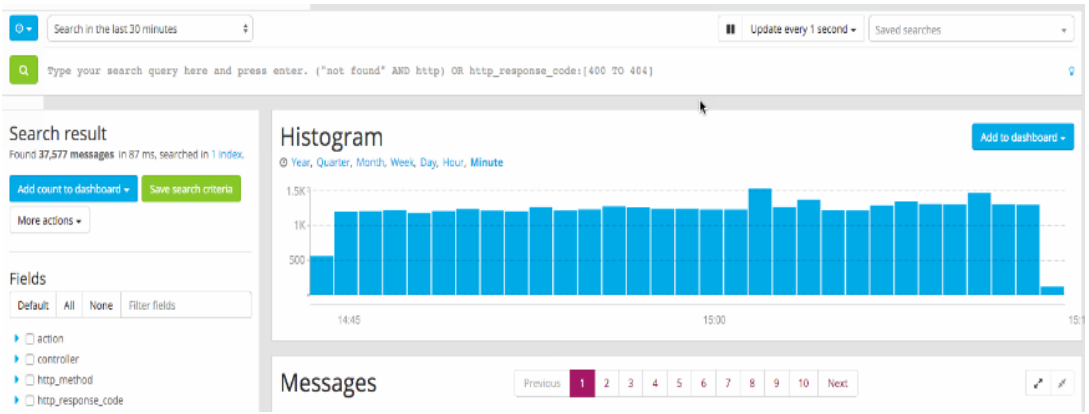

Figure 9. Graylog Interface https://graylog.org/post/graylog-security-the-affordable-siem-alternative/

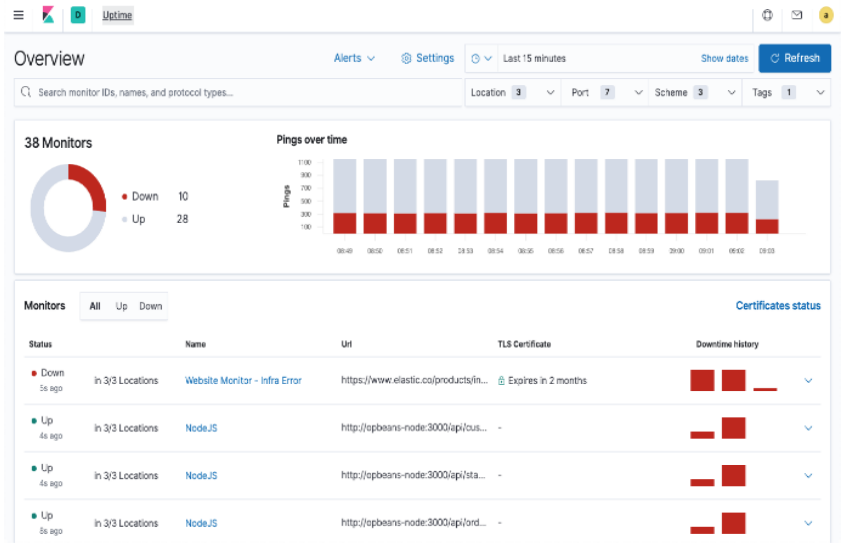

Figure 10. ELK Stack Interface https://www.elastic.co/elastic-stack

Setting up SecurityOnion terminal and checking services:

Figure 11. Desktop terminal

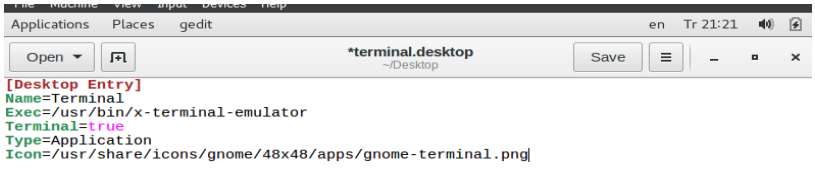

Figure 12. Terminal application

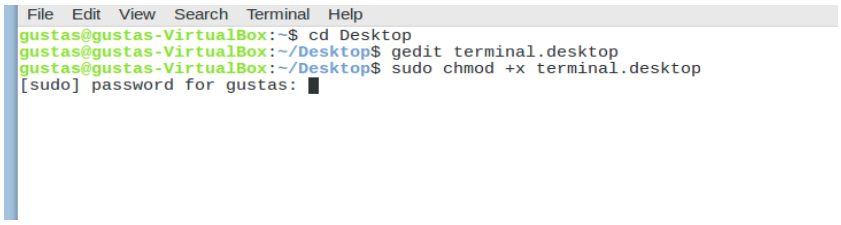

Figure 13. Opening terminal

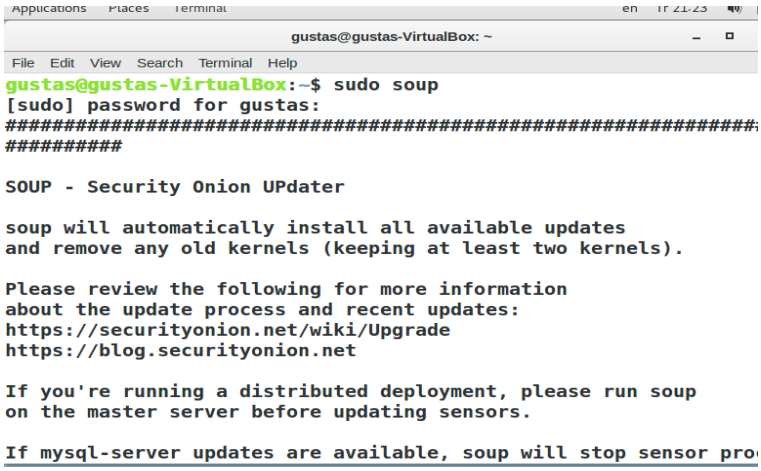

Figure 14. Updating services

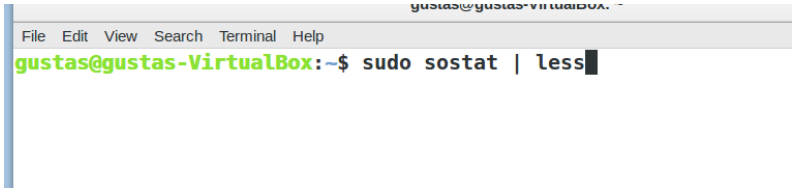

Figure 15. Service check

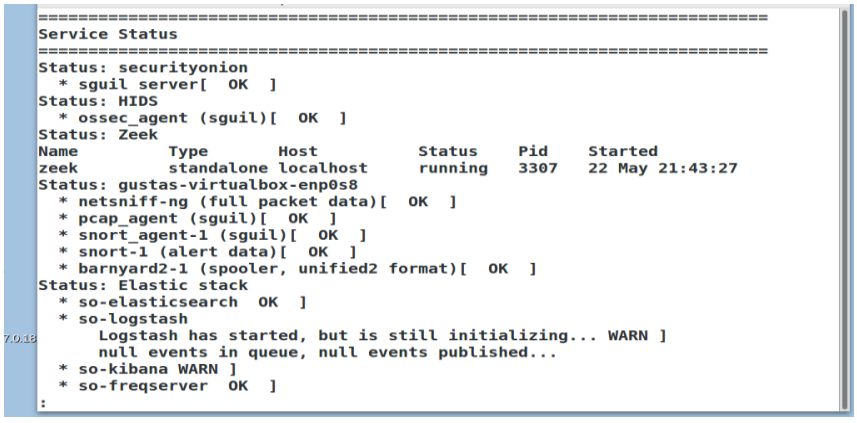

Figure 16. Services

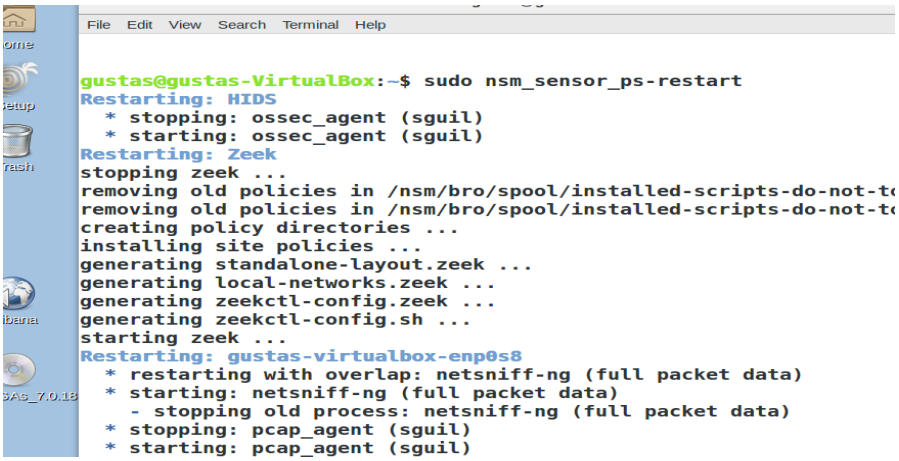

Figure 17. Service restart

Testing if SGUIL picks up test viruses:

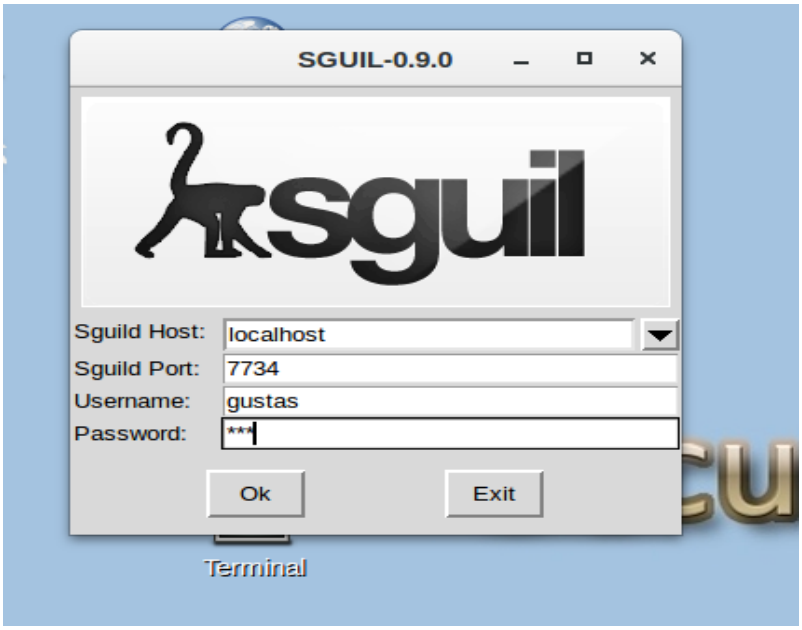

Figure 18. SGUIL user authentication

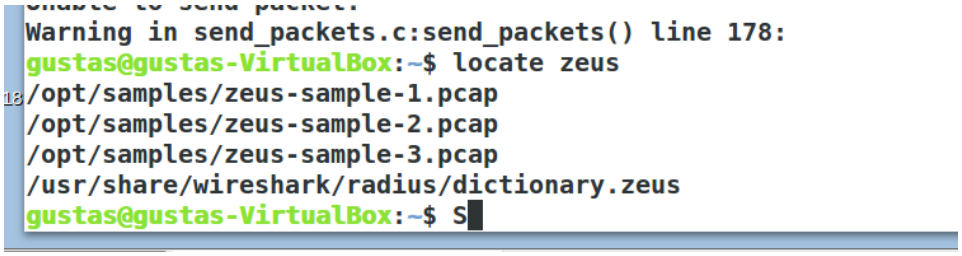

Figure 19. Finding Zeus trojan

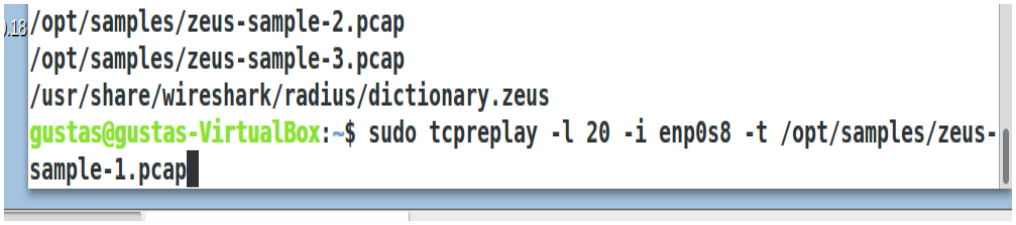

Figure 20. Using Zeus trojan 20 packets