



**Politecnico
di Torino**

Master of Science in Computer Engineering

Master Degree Thesis

Analysis of Security Configuration for IDS/IPS

Supervisors

prof. Riccardo Sisto

prof. Fulvio Valenza

dott. Daniele Bringhenti

Candidate

Andrea TRISOLINO

ACADEMIC YEAR 2022-2023

Abstract

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are critical components in ensuring the security of computer networks and systems. With the constant evolution of cyber threats, it is essential to understand the functionalities, benefits, and limitations of these systems. This Thesis provides a comprehensive overview of IDS and IPS, including their history, key features, and implementation strategies.

The Thesis begins by introducing the concepts of intrusion detection and prevention, to the differences between the two systems. It presents the historical development of IDS and IPS, from their early origins to the current state-of-the-art solutions. The Thesis delves into the various types of IDS (e.g., network-based, host-based, and hybrid) and IPS (e.g., inline, out-of-band, and hybrid), examining their strengths and weaknesses.

It will be discussed how these systems work and it explores key challenges faced by IDS and IPS, including false positives, false negatives, and evasion tactics employed by attackers.

Having clarified the role of IDS/IPS systems in the field of cyber security, the framework VEREFOO will be presented to better clarify the aim of this Thesis.

By offering a comprehensive understanding of VEREFOO, the state of the art and the future application of that; the importance of the framework will be evident at once.

After showing the different alternatives of IDS and IPS products on the market, the two chosen for the project will be explained.

The Thesis concludes with the implementation of the code that will enable the VEREFOO framework to be able to collect data from the two systems introduced, in order to distribute configurations to Firewalls and guarantee a high level of security posture within a scope perimeter.

Acknowledgements

A tutte le notti di lavoro e ai pomeriggi alla scrivania,
a tutti i momenti in cui l'unica voce che ti conforta è la tua.

Alle persone uniche che ho perso,
alle perdite che ho guadagnato,
a tutti i lati di Mondo rimandati.

Agli anni passati sporco di polvere di carbone
seppur non in miniera.
Al cinismo di azioni inumane.
Ai pesci in faccia,
ai bastoni fra le ruote,
dietro le ginocchia.

A chi ha sempre creduto in me
anche solo a parole.
A chi continua ad essere al mio fianco.
A mia madre,
a mio padre, a zio Tonino
a Ilaria.

A testa alta, cammino e poi corro.

Contents

List of Figures	7
1 Introduction	9
1.1 Background information on IDS and IPS	10
1.2 Historical context and evolution	11
1.3 Importance of IDS and IPS in network security	11
2 Intrusion Detection System (IDS)	13
2.1 Definition and overview	13
2.1.1 Network-based IDS (NIDS)	13
2.1.2 Host-based IDS (HIDS)	14
2.1.3 Hybrid IDS (HIDS/NIDS)	14
2.2 Prevention methods and techniques	14
2.2.1 Rule-based IDS	14
2.2.2 Signature-based IDS	15
2.2.3 Behavior-based IDS	15
2.2.4 Anomaly-based IDS	15
2.3 Other IDS categorizations	16
2.3.1 Software-based IDS	16
2.3.2 Hardware-based IDS	17
2.3.3 Hybrid IDS	17
2.4 Components of an IDS	17
2.5 Features of IDS	18
2.6 Limitations of IDS	20

3	Intrusion Prevention System (IPS)	22
3.1	Definition and overview	22
3.1.1	Network-based IPS	23
3.1.2	Host-based IPS	23
3.1.3	Virtual IPS	23
3.1.4	Wireless IPS	23
3.1.5	Inline IPS	23
3.2	Prevention methods and techniques	24
3.2.1	Signature-based detection	24
3.2.2	Behavior-based detection	24
3.3	Other IPS categorizations	24
3.3.1	Software-based IPS	25
3.3.2	Hardware-based IPS	25
3.3.3	Hybrid IPS	26
3.4	Components of an IPS	26
3.5	Features of IPS	27
3.6	Limitations of IPS	29
4	Comparison between IDS and IPS	32
5	Suitable scenarios for IPS/IDS	35
6	Limitations of IDS/IPS exploited by an attacker	36
7	Implementation of IDS and IPS	38
7.1	Best practices for implementing an IDS/IPS	38
7.2	Challenges in implementing IDS and IPS	39
8	OWASP	42
8.1	What is OWASP	42
8.2	OWASP Top Ten	42
8.3	IDS-IPS and OWASP	46
9	IDPS Case Studies	47
10	Future of IDS and IPS	49
10.1	Emerging trends in IDS and IPS	49
10.2	Limitations of IDS and IPS technologies in the evolving cyber landscape	51

11 IDPS comparison	53
12 Firewall and IDS/IPS	71
13 VEREFOO	73
14 MiddleVerefoo Project	78
15 Conclusions	85
Bibliography	87

List of Figures

11.1 Snort logo	53
11.2 Snort dashboard	56
11.3 Suricata logo	56
11.4 Suricata dashboard	58
11.5 Zeek logo	58
11.6 Zeek dashboard	59
11.7 Security Onion logo	60
11.8 Security Onion dashboard	61
11.9 OSSEC logo	61
11.10 OSSEC example dashboard	62
11.11 Cisco FirePower logo	62
11.12 Cisco FirePower dashboard	64
11.13 FortiGuard logo	64
11.14 FortiGate dashboard	65
11.15 EasyIDS logo	66
11.16 EasyIDS example dashboard	66
11.17 Untangle logo	67
11.18 Untangle dashboard	68
11.19 AIDE logo	69
11.20 Samhain logo	69
13.1 Allocation Graph input	74
13.2 Service Graph functions input	74
13.3 Network Security Requirements input	75
13.4 Firewall Allocation Scheme output	76
13.5 Firewall rules output	76
14.1 IDPS comparison overview	79
14.2 Host-based IDS comparison overview	80

Chapter 1

Introduction

Quoting a well-known cyber security expression: "Companies today are divided into two groups: those that have been attacked and know it, and those that have been attacked and don't know it yet."

The worldwide technological trend has led to ever-increasing digitization with the aim of reducing the ecological impact of old information management, but more importantly, making data available anytime, anywhere.

Goods and services companies, but also banks, hospitals and, in general, public administration systems, are increasingly connection-oriented, giving users the ability to access them remotely for any service.

The amount of data exchanged every day has more than doubled, and with it the cyberattacks designed to steal that data by using it for malicious purposes.

In this context, firewalls play a key role because they are designed, by definition, to separate two environments with different levels of security: the internal company perimeter and the Internet.

According to Gartner, the most notable American technological research and consulting firm, through 2023, 99% of firewall breaches will be caused by firewall misconfiguration ("The Dangers of Firewall Misconfigurations and How To Avoid Them" - Nov 16 2020 - akamai.com).

One of the major criticalities of these devices is the continuous need to update them with new configurations to ensure that only lawful and reliable traffic lands in the corporate perimeter, leaving everything else out, guaranteeing higher levels of security.

This is the goal set for VEREFOO: design the best FWs allocation and FW rules for a specific network^{1 2}.

¹D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Automated Firewall Configuration in Virtual Networks", IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1559-1576, 2023

²D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, "Automation for network security configuration: state of the art and research trends", ACM Computing, 2023

VEREFOO is a framework whose acronym stands for Verified Refinement and Optimized Orchestration.

Currently, VEREFOO works for packet filter firewalls but it will be ready for other security functions as well.

The functionalities currently available on VEREFOO will be extended, by creating an interface with IDPS devices.

Given the high number of exposed Internet services to support the protection of the corporate perimeter, an additional line of defense has been introduced in the detection of illegitimate access: the IDS and IPS, Intrusion Detection System and Intrusion Prevention System, respectively, or IDPS for short.

These appliances or software tools are engaged to detect unauthorized access to local networks or computers, through analysis of network traffic or events on different hosts, and based on priori defined security rules. The IDS will detect exploiting databases, libraries and attack signatures, while IPS will work to prevent them.

The purpose of this Thesis is to illustrate how IDPS work, the differences between the various types, and more importantly, how to integrate such devices into the VEREFOO ecosystem.

1.1 Background information on IDS and IPS

Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are security technologies that monitor network traffic or host activities searching for signs of unauthorized access or malicious activity.

IDSs typically consist of several components working together to monitor, analyze, and respond to potential security threats or policy violations in a network or system.

They can be generally classified into three main types: Network-based IDS (NIDS), Host-based IDS (HIDS), and Hybrid IDS (HIDS/NIDS). NIDSs monitor network traffic for signs of malicious activity, while HIDSs monitor host activities for signs of intrusion attempts. Hybrid IDSs combine the capabilities of both NIDS and HIDS and can provide a more comprehensive view of network security.

In addition to detecting potential security threats, IPSs can also take immediate action to prevent or block those threats from succeeding, like blocking traffic, terminating or resetting connections.

IPS can also operate at the network or host level and they can be used to protect against a wide range of attacks, including denial-of-service attacks (DoS), buffer overflows, and SQL injection attacks.

IPS are also classified into three main types: Network-based IPS (NIPS), Host-based IPS (HIPS), and Hybrid IPS (HIPS/NIPS).

The primary difference between IDS and IPS is that IDS are designed to detect and alert on suspicious activity, while IPS can detect and also prevent attacks in real-time.

1.2 Historical context and evolution

The history of Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) date back to the early days of computer networking in the 1980s. At that time, the primary focus of network security was on securing the perimeter of a network. Firewalls were the primary security tool used to protect networks from external threats.

However, as networks became more complex and sophisticated, it became clear that perimeter-based security was not enough to protect against all types of attacks. Hackers and cybercriminals started to develop more advanced and sophisticated techniques to bypass perimeter defenses and gain access to sensitive data.

The first IDS appeared in the late 1980s and early 1990s as a response to this realization. These systems were typically host-based and relied on signatures or rules to detect known attacks. They were limited in their ability to detect new or unknown threats and often suffered from high false positive rates.

As networking technology continued to evolve, NIDS (Network-based IDS) emerged, which were designed to monitor network traffic in real-time and detect malicious activity. These systems were capable of detecting a wider range of attacks and were more effective than host-based IDSs.

In the late 1990s and early 2000s, IPS technology emerged as the next step in network security. IPS were designed to not only detect malicious activity but also to take immediate action to block or prevent it from occurring. IPSs can operate at the network or host level and can be used to protect against a variety of attacks, including denial of service attacks, buffer overflows, and SQL injection attacks.

As the threat landscape continues to evolve, IDS and IPS technologies have also evolved to keep up with new and emerging threats. Machine learning and AI are being used to improve the accuracy and effectiveness of IDSs and IPSs, while cloud-based IDS and IPS solutions are becoming more popular as Organizations move their data to the cloud.

The historical context and evolution of IDS and IPS demonstrate the ongoing need for network security technologies that can adapt to new and emerging threats. As the threat landscape continues to evolve, IDSs and IPSs will continue to evolve and improve to ensure that Organizations can protect their networks from the growing range of cyber threats.

1.3 Importance of IDS and IPS in network security

The importance of IDS and IPS in network security cannot be overstated. With the increasing sophistication of cyber threats, it has become essential for Organizations to implement effective security measures to protect their networks and data. Hackers and cybercriminals are constantly developing new techniques and tools to bypass traditional security measures, making it crucial for Organizations to have advanced security technologies like IDS and IPS in place.

IDSs and IPSs provide several benefits to Organizations, including:

- **Detects and alerts on suspicious activities:** IDSs and IPSs can detect and alert on suspicious activities in real-time, allowing security teams to respond quickly to potential security threats.
- **Provides visibility into network traffic:** IDSs and IPSs can provide detailed insights into network traffic, allowing security teams to identify potential security threats and take proactive measures to prevent them from escalating.
- **Helps Organizations comply with regulatory requirements:** Many regulatory frameworks require Organizations to implement IDSs and IPSs as part of their security measures. Compliance with these regulations can help Organizations avoid penalties and protect their reputation.
- **Prevents attacks in real-time:** IPSs can take immediate action to prevent attacks from succeeding, helping to protect sensitive data and prevent costly data breaches.
- **Provides an additional layer of security:** IDSs and IPSs provide an additional layer of security beyond traditional perimeter defenses like firewalls, helping to protect against a wider range of threats.

Chapter 2

Intrusion Detection System (IDS)

2.1 Definition and overview

An Intrusion Detection System (IDS) is a security technology designed to monitor network traffic or host activities for signs of unauthorized or malicious activity. IDSs are designed to detect a variety of security threats, including malware, unauthorized access attempts, and suspicious network traffic.

The alert typically includes information about the type of attack, the affected system, and the severity of the attack. There are three main types of Intrusion Detection System (IDS): Network-based IDS (NIDS), Host-based IDS (HIDS), and Hybrid IDS (HIDS/NIDS). Each type of IDS has its own strengths and weaknesses and can be deployed in different ways to protect against specific types of security threats.

- **2.1.1 Network-based IDS (NIDS)**

They are designed to monitor network traffic for signs of malicious activity. NIDSs are typically deployed at strategic points within a network, such as at the network perimeter, switches, or routers. NIDSs can identify network-based attacks, such as denial-of-service (DoS) attacks, port scans, and other suspicious network activity.

Suppose an attacker is attempting to exploit a vulnerability in a web application that is hosted on a web server within an Organization's network. The attacker sends a series of HTTP requests to the web application, attempting to inject SQL code into the requests to gain unauthorized access to the underlying database. The network IDS is configured to monitor traffic on the network segment that includes the web server. The IDS is also configured with a rule that detects SQL injection attacks based on a specific pattern of HTTP requests. When the attacker sends the SQL injection requests to the web application, the IDS detects the pattern of network traffic that matches the SQL injection rule. The IDS generates an alert, which is sent to the security team for further investigation.

- **2.1.2 Host-based IDS (HIDS)**

They are designed to monitor activity on a specific host or endpoint device, such as a server or workstation. HIDSs can detect attacks that originate from within a network, such as malware infections or unauthorized access attempts. HIDSs can also detect system-level attacks, such as file integrity violations, unauthorized changes to system files, and other suspicious activities.

- **2.1.3 Hybrid IDS (HIDS/NIDS)**

They combine the capabilities of both NIDS and HIDS and can provide a more comprehensive view of network security. Hybrid IDSs can detect both network-based and host-based attacks, providing a more complete picture of the security posture of a network. Hybrid IDSs can also provide more accurate and reliable alerts by correlating data from both NIDS and HIDS components.

NIDSs are typically used to protect against external threats, while HIDSs are used to protect against internal threats. Hybrid IDSs can provide a more comprehensive view of network security but may require more resources to deploy and manage. There are many software and hardware IDS available on the market today, each with its own strengths and weaknesses.

In addition to these main types of IDS, there are also specialized IDSs designed to address specific security threats or environments. For example, Wireless IDSs (WIDS) are designed to monitor wireless networks for security threats, while Virtual IDSs (VIDS) are designed to monitor virtualized environments.

2.2 Prevention methods and techniques

IDSs use a variety of techniques to detect security threats, including the use of rules, signatures, behavior and anomaly detection:

- **2.2.1 Rule-based IDS**

They use customized rules to identify potential threats based on specific network activities or behaviors. Is more flexible for new or previously unknown attacks, as rules can be updated or created as new attacks are discovered. Each rule defines a specific pattern of behavior that is characteristic of a particular type of attack. For example, a rule may specify a particular sequence of network packets that is indicative of a buffer overflow attack. When network traffic or system activity matches a predefined rule, the IDS generates an alert indicating that an attack may be in progress.

- **2.2.2 Signature-based IDS**

They use a database of known attack signatures to identify potential threats. They are similar to Rule-based but more specific as each signature is unique to a particular type of attack. Signature-based detection involves comparing the network traffic or system activity against a database of known attack signatures. An attack signature is a unique pattern of data that is characteristic of a particular type of attack. For example, a signature for a SQL injection attack may include specific SQL commands or syntax, but also a content of e-mail subject lines, file hashes, so based on IoCs (Indicator of Compromise).

When an IDS receives network traffic or system activity, it compares the data against its database of attack signatures. If a signature match is found, the IDS generates an alert indicating that an attack may be in progress. Signature-based detection is easy to implement and can detect known attacks with a high degree of accuracy. Additionally, signature databases can be updated frequently to include new attack signatures as they are discovered. However, limits are on detections of unknown attacks or attacks that use novel techniques that do not match any known attack signatures. Attackers can also evade signature-based detection by using techniques such as obfuscation, which alters the attack signature to make it difficult for the IDS to detect. To overcome these limitations, IDSs may use other detection methods in addition to signature-based detection, such as anomaly-based detection or behavior-based detection.

- **2.2.3 Behavior-based IDS**

They analyze the behavior of users or systems to detect suspicious activity. This method is based on the assumption that attackers may behave differently from legitimate users or systems and that deviations from expected behavior may indicate an attack. Behavior-based detection is useful for detecting attacks that rely on insider threats, social engineering, or other non-technical means. Once a baseline of expected behavior for users or systems is set, the IDS monitors user or system behavior to identify potential suspicious activity. Behavior deviations may include unusual login patterns (login attempts from unusual geographic locations or unusual time of the day), unusual file access patterns, or other suspicious behavior.

- **2.2.4 Anomaly-based IDS**

They use machine learning and statistical analysis to detect deviations from normal network behavior, which may indicate a security threat. In this case, the IDS first establishes a baseline of normal behavior by analyzing network traffic or system activity over a period of time. The IDS then compares subsequent network traffic or system activity against the baseline to identify deviations from normal behavior. Deviations may include unusual network

traffic patterns (e.g. unusually large amounts of data being transferred between systems or infrequent protocols or ports), unusual system activity (e.g. suspicious file modifications or unusual process activity) or other suspicious behavior. This kind of IDS can resolve the limitation of detecting previously unknown attacks or attacks that use novel techniques that do not match any known attack signatures. Surely, it can be more complex to implement than signature-based detection, since it requires establishing and maintaining a baseline of normal behavior. It may also generate false negatives if an attacker is able to mimic normal behavior closely enough to avoid detection

2.3 Other IDS categorizations

Intrusion Detection System (IDS) are available in both software and hardware, each with its own advantages and disadvantages. Here is an overview of the software and hardware versions of IDS:

2.3.1 Software-based IDS

They are designed to run on a variety of operating systems, including Windows, Linux, and macOS. They can be installed on dedicated hardware or run on virtual machines. Some software-based IDSs are open-source, while others are proprietary. They can be configured to monitor network traffic in real-time, analyze logs, and detect potential security threats. Software-based IDSs are generally more cost-effective than hardware-based systems and can be easier to deploy and manage. Following are some of the most famous software IDSs.

- **Snort**: is one of the most popular open-source IDSs. It is highly customizable and can be used to monitor network traffic for a wide range of threats. Snort uses a signature-based detection approach and can be easily integrated with other security solutions.
- **Suricata**: another popular open-source product that is known for its high-performance and scalability. It offers both signature-based and behavior-based detection methods and can be used to monitor network traffic in real-time.
- **OSSEC**: is an open-source HIDS system that is designed to monitor individual hosts for potential threats. It offers real-time alerts and can be used to detect a wide range of threats, including malware, rootkits, and unauthorized access.
- **Bro**: is an open-source NIDS system that uses a high-level scripting language to analyze network traffic. It offers both signature-based and behavior-based detection methods and can be used to monitor network traffic in real-time.

2.3.2 Hardware-based IDS

They are designed to run on dedicated hardware, such as a server or appliance. They are typically designed for high-traffic environments or in situations where network performance is critical, hardware-based where they may provide better performance and reliability than software-based IDS and can handle large volumes of network traffic. Hardware-based IDSs often come with specialized hardware components, such as network interface cards (NICs), that are optimized for high-speed data processing. Hardware-based IDS may also be used in specialized environments, such as industrial control systems or critical infrastructure, where specialized hardware is required to monitor network traffic or system activity. They are not as widely used today as software-based IDS. This is because hardware-based IDS typically require dedicated hardware appliances to monitor network traffic, which can be expensive and difficult to scale. Following are some of the most famous hardware IDSs.

- **Cisco FirePower:** IDS hardware-based designed for high-performance environments. They offer advanced threat detection capabilities and can be easily integrated with other Cisco security solutions.
- **Palo Alto Networks:** it offers a range of hardware-based IDSs, including the PA-5200 Series and the PA-800 Series. These systems are designed to provide advanced threat detection and prevention capabilities and can be easily integrated with other Palo Alto Networks security solutions.
- **Fortinet:** it offers a range of hardware-based IDSs, including the FortiGate Series and the FortiAnalyzer Series. These systems are designed to provide advanced threat detection and prevention capabilities, as well, and can be easily integrated with other Fortinet security solutions.
- **Juniper Networks:** it offers a range of hardware-based IDSs, including the SRX Series and the Sky Advanced Threat Prevention (ATP) solution. Advanced threat detection and prevention capabilities are provided and they can be easily integrated with other Juniper Networks security solutions

2.3.3 Hybrid IDS

This system is a combination of both software-based and hardware-based IDSs. Hybrid IDSs can offer the best of both worlds, with the flexibility and ease of configuration of software-based systems and the performance and scalability of hardware-based systems. Hybrid IDSs can be particularly useful for Organizations that have high-performance requirements but also need the flexibility to configure the system to their specific needs.

2.4 Components of an IDS

An IDS (Intrusion Detection System) typically consists of several components that work together to monitor network or system activity for signs of a security threat.

The main components of an IDS include:

- **Sensors:** are the devices or software agents that capture network or system activity. They collect data from network traffic, system logs, or other sources and send it to the IDS for analysis.
- **Analysis engine:** is the core component of the IDS that processes the data collected by the sensors. It applies detection algorithms, rules, and other techniques to identify patterns of activity that may indicate a security threat.
- **Alerting system:** the alerting system generates alerts when the analysis engine detects activity that matches a known attack pattern or violates a security policy. Alerts may be sent to a security team or other stakeholders for further investigation.
- **Reporting system:** the reporting system generates reports that provide insights into the security posture of the Organization. Reports may include information about detected threats, attack trends, or other security-related metrics.
- **Management console:** it provides a user interface for configuring and managing the IDS. It allows administrators to configure sensors, rules, and other settings, view alerts and reports, and perform other administrative tasks.
- **Centralized storage:** it is used to store data collected by the sensors, including network traffic, system logs, and other security-related data. The storage can be used for forensic analysis, incident response, or other purposes.

2.5 Features of IDS

Alerting

IDS alerting features extend functionalities about:

- **Real-time alerts:** IDSs can send real-time alerts to security personnel through email, SMS, or other notification methods, enabling rapid response to potential threats.
- **Customizable alert thresholds:** to minimize false positives and ensure that security teams focus on high-priority threats, IDS solutions offer configurable alert thresholds. Organizations can set the sensitivity level and choose the types of alerts they want to receive.
- **Alert prioritization:** some IDS solutions prioritize alerts based on their severity, enabling security teams to address the most critical issues first.

Reporting

Reporting for IDSs is covered by following:

- **Detailed event logs:** when an alert is triggered, IDSs typically provide detailed event logs that offer insights into the nature of the suspicious activity, such as source and destination IP addresses, timestamps, and other relevant information.
- **Visualizations and dashboards:** IDSs often include visualizations and dashboards to help security teams quickly assess the current threat landscape, identify trends, and monitor the overall security posture of the Organization.
- **Scheduled reports:** some IDS solutions allow for the creation of scheduled reports, which can be automatically generated and sent to relevant stakeholders at regular intervals. These reports can provide insights into long-term trends and help

Organizations assess the effectiveness of their security measures.

- **Compliance reporting:** Organizations can meet various compliance requirements by generating reports tailored to specific regulations and standards.

Integration and Correlation

Other security tools can be interfaced with IDS solutions allowing a more comprehensive view of the Organization's security landscape:

- **Integration with SIEM systems:** by integrating with Security Information and Event Management (SIEM) systems, IDS data can be correlated with other security events and logs, providing a more holistic understanding of potential threats.

- **Threat intelligence feeds:** some IDS solutions can incorporate external threat intelligence feeds, enhancing their detection capabilities and ensuring that they stay up-to-date with the latest attack patterns and indicators of compromise (IoCs).

Organizations and regulatory requirements

Many regulatory frameworks require Organizations to implement security controls and monitoring processes to protect sensitive data and systems. IDSs can help Organizations meet these requirements by providing real-time monitoring of network traffic and host activities, generating alerts for potential security threats, and maintaining logs of security incidents.

Some examples of regulatory requirements that IDSs can help Organizations comply are:

- **HIPAA:** the Health Insurance Portability and Accountability Act (HIPAA) requires healthcare Organizations to implement security controls and monitoring processes to protect electronic patient health information (ePHI).

- **PCI-DSS:** the Payment Card Industry Data Security Standard (PCI-DSS) requires Organizations that accept credit card payments to implement security controls and monitoring processes to protect cardholder data. IDSs can help Organizations comply with PCI-DSS generating alerts for any suspected or actual data breaches involving cardholder data.

- **GDPR:** the General Data Protection Regulation (GDPR) requires Organizations that process personal data of European Union (EU) residents to implement security controls and monitoring processes to protect personal data. Alerts can be generated for any suspected or actual data breaches involving personal data.

IDSs can also help Organizations comply with other regulatory requirements, such as the Sarbanes-Oxley Act (SOX), the Federal Information Security Management Act (FISMA), and the National Institute of Standards and Technology (NIST) Cybersecurity Framework.

To ensure that IDS solutions actually support compliance with various regulatory requirements, Organizations should follow best practices, such as:

- **Risk assessment:** conduct a thorough risk assessment to identify potential threats and vulnerabilities in the Organization's IT systems. This assessment will help determine the appropriate IDS implementation to address specific risks and satisfy regulatory requirements.

- **Policy development:** establish clear policies and procedures related to the use of IDS, including incident response plans, system configuration guidelines, and ongoing maintenance requirements.

- **System integration:** integrate IDS with other security tools, such as Security Information and Event Management (SIEM) systems, firewalls, and endpoint protection solutions, to ensure comprehensive protection and visibility across the Organization's IT environment.
- **Compliance reporting:** leverage IDS reporting capabilities to generate reports tailored to specific regulatory requirements, demonstrating the Organization's compliance with these standards.

2.6 Limitations of IDS

False positives and false negatives

Systems like IDS can generate two types of errors: false positives and false negatives. False positives occur when the IDS system generates an alert for an event that is not actually a security threat, while false negatives occur when the IDS system fails to generate an alert for an event that is a security threat.

False Positives can occur for several reasons, including:

- **Incorrectly configured rules or signatures:** IDSs use rules and signatures to identify potential security threats. If these rules or signatures are not configured correctly, the IDS system may generate false positives.
- **Network or system anomalies:** IDSs use anomaly detection to identify potential security threats. If network or system anomalies occur, the IDS system may generate false positives.
- **Environmental factors:** IDSs can be affected by environmental factors, such as network latency or packet loss. These factors can cause the IDS system to generate false positives.

False positives can be a significant problem for Organizations, as they can lead to unnecessary alerts and create additional workload for security analysts. To reduce false positives, IDSs should be configured correctly and regularly updated to include new and emerging threats.

False negatives can occur for several reasons:

- **Outdated rules or signatures:** because use rules and signatures used, in the case of outdated rules or signatures, the IDS system could misdetect a cyber issue.
- **Encrypted traffic:** IDSs may have difficulty detecting security threats in encrypted traffic, as they cannot inspect the contents of encrypted data.
- **Network or system anomalies:** in the case of network or system anomalies, the IDS system may fail to detect security threats.

False negatives can be a significant problem for Organizations, as they can leave the Organization vulnerable to cyberattacks.

Limitations to prevent attacks

One of the main reasons for the limited ability of IDSs to prevent attacks is that they are passive systems. IDSs do not actively block network traffic or host activities, but instead, they monitor network traffic and host activities for signs of potential security threats. This means that IDSs can only generate alerts when they detect

potential security threats, but they cannot actively prevent those threats from occurring.

Another reason for the limited ability of IDSs to prevent attacks is that they can only detect known threats. IDSs can only detect threats that are included in their database or that match specific rules or anomalies. This means that IDSs may not be able to detect new or unknown threats, which could leave the Organization vulnerable to cyberattacks. Advanced threats, such as zero-day exploits and targeted attacks, are designed to evade traditional security measures and they are often difficult to detect using standard detection methods, so there will not be detected.

Anomaly detection techniques can detect unusual behavior or traffic patterns, but they may generate a large number of false positives and may not be effective against targeted attacks that are designed to blend in with normal traffic.

To overcome all these limitations, Organizations should use a multi-layered approach to network security, which includes a combination of preventive and detective controls. Preventive controls, such as firewalls, antivirus software, and access controls, are designed to prevent security threats from occurring in the first place. By using a combination of preventive and detective controls, Organizations can improve their overall security posture and reduce the risk of cyberattacks.

Best practice for false positive detections

To reduce false positive, IDS rules and signatures must be regularly review and adjusted. Rules must be specific to the threats relevant to the environment and eliminate overly broad or outdated rules. Updating the rules and signatures frequently ensures the system is kept up to date with the latest threat intelligence.

Many IDS have configurable sensitivity levels that determine how strictly the system analyzes network traffic. Lowering the sensitivity can help reduce false positives but may increase the risk of false negatives. Striking the right balance between sensitivity and specificity is crucial to optimize the IDS's performance.

Another good advice is to implement the anomaly-based detection as it learns and adapts to the normal behavior of the network. By establishing a baseline of normal activity, the system can be more accurate in identifying deviations that may indicate a genuine threat.

The use of the stateful protocol analysis examines network traffic in the context of the specific protocols being used and their expected states. This method can help reduce false positives by detecting attacks that exploit protocol vulnerabilities or violate protocol specifications, which may not be identified by signature-based detection alone.

With the allowlist can be created and maintained a list of known benign activities, IP addresses, or applications that the IDS should not flag as malicious. This can help reduce false positives by explicitly allowing trusted traffic and reducing the chances of misclassification.

As fundamental, it remains the training security personnel because an experienced analyst can better understand the nuances of the system and make more informed decisions when adjusting settings, rules, and signatures.

Chapter 3

Intrusion Prevention System (IPS)

3.1 Definition and overview

An Intrusion Prevention System (IPS) is a network security device that is designed to identify and prevent potential security threats in real-time. IPSs are similar to Intrusion Detection System (IDS) in that they monitor network traffic and host activities for signs of potential security threats. However, IPSs are more proactive and have the ability to actively block traffic or host activities that are deemed to be security threats.

IPSs use a variety of detection methods and techniques to identify potential security threats, including signature-based detection, rule-based detection, anomaly detection, and behavioral analysis. When an IPS system detects a potential security threat, it can take several actions to prevent the threat from causing harm, such as blocking traffic from a specific IP address or quarantining a compromised host.

IPSs can be deployed at various points in the network, including at the perimeter, within the data center, and on endpoints. IPSs can also be integrated with other security technologies, such as firewalls, IDSs, and Security Information and Event Management (SIEM) systems.

IPSs provide several benefits to Organizations looking to enhance their network security, including:

- **Real-time threat prevention:** IPSs can prevent potential security threats in real-time, providing Organizations with the opportunity to respond quickly and mitigate the impact of a security incident.
- **Enhanced network visibility:** IPSs can provide Organizations with enhanced visibility into network traffic and host activities, allowing them to identify potential security threats and track user activity.
- **Compliance with regulatory requirements:** IPSs can also help Organizations comply with regulatory requirements related to network security and data protection.
- **Reduced workload for security personnel:** IPSs can automate the detection

and prevention of potential security threats, reducing the workload for security personnel and allowing them to focus on more complex security tasks.

There are several types of Intrusion Prevention System (IPS) that Organizations can use to protect their networks and data. These include the following:

- **3.1.1 Network-based IPS**

They are deployed at the network perimeter and inspect traffic as it enters and exits the network. These systems use various detection methods, including signature-based detection, rule-based detection, and anomaly detection, to identify potential security threats. When a potential threat is detected, the network-based IPS can take action to block the traffic, quarantine the host, or perform other actions to prevent the threat from causing harm.

- **3.1.2 Host-based IPS**

They are deployed on individual hosts, such as servers or endpoints, and monitor host activities for signs of potential security threats. These systems can detect and prevent threats that may bypass network-based IPSs, such as attacks that originate from within the network, on an internal asset. Host-based IPSs can also provide granular visibility and control over host activity, allowing Organizations to enforce policies and protect sensitive data.

- **3.1.3 Virtual IPS**

They are deployed in virtualized environments, such as cloud environments or virtual private networks (VPNs), and monitor network traffic and host activities for signs of potential security threats. Virtual IPSs can provide the same level of protection as traditional network-based IPSs, but they are designed to operate in virtualized environments and can scale up or down as needed.

- **3.1.4 Wireless IPS**

They are designed to protect wireless networks from potential security threats. These systems can monitor wireless traffic for signs of potential threats, such as rogue access points or unauthorized wireless devices. Wireless IPSs can also enforce policies and prevent unauthorized access to wireless networks.

- **3.1.5 Inline IPS**

They are deployed inline with network traffic and actively inspect and block traffic that is deemed to be a security threat. These systems can provide real-time threat prevention and can block traffic before it reaches its intended destination. Inline IPSs can also provide enhanced visibility into network traffic and host activities.

3.2 Prevention methods and techniques

IPSs work by using various prevention methods and techniques to detect and block potential security threats. There are two main prevention methods used by IPSs: signature-based and behavior-based detection.

3.2.1 Signature-based detection

They use pre-defined signatures or patterns to identify known threats. The IPS system compares network traffic or host activities against a database of signatures (known patterns of attacks or malicious activities), and if a match is found, the system takes action to prevent the threat from causing harm. Signature-based detection is effective for detecting known threats, such as malware or specific attack patterns.

3.2.2 Behavior-based detection

They use machine learning and artificial intelligence to analyze network traffic or host activities and identify abnormal or suspicious behavior. The IPS system builds a baseline of normal behavior and flags any activity that deviates from the baseline. Behavior-based detection is effective for detecting new or unknown threats that may not be detected by signature-based detection.

IPSs use various prevention techniques to block potential security threats. These techniques include:

- **Blocking:** is the most common technique used by IPSs to prevent potential security threats. When a potential threat is detected, the IPS system can block traffic from a specific IP address, quarantine a compromised host, or perform other actions to prevent the threat from causing harm.
- **Quarantine:** is a technique used by IPSs to isolate compromised hosts or devices from the network. When a potential threat is detected, the IPS system can quarantine the affected host or device, preventing it from communicating with other devices on the network.
- **Alerting:** is a technique used by IPSs to notify security personnel of potential security threats. When a potential threat is detected, the IPS system generates an alert, which can be sent to security personnel via email, SMS, or other communication channels.
- **Redirection:** is a technique used by IPSs to redirect traffic to a different destination. When a potential threat is detected, the IPS system can redirect traffic to a different destination, such as a honeypot, where the traffic can be further analyzed without causing harm to the network.

3.3 Other IPS categorizations

Intrusion Prevention System (IPS) are available in both software and hardware versions, each with its own advantages and disadvantages. Here is an overview of

the software and hardware versions of IPS:

3.3.1 Software-based IPS

They are designed to run on a variety of operating systems, including Windows, Linux, and macOS. They can be installed on dedicated hardware or run on virtual machines. Some software-based IPSs are open-source, while others are proprietary. They can be configured to monitor network traffic in real-time, analyze logs, and prevent potential security threats. Software-based IPSs are generally more cost-effective than hardware-based systems and can be easier to deploy and manage. Following are some of the most famous software IPSs.

- **Snort:** is an open-source IPS (also) system that is widely used in the industry. It uses signature-based detection and can be customized to fit the specific needs of an Organization.
- **Suricata:** is another open-source IPS system that is known for its performance and scalability. It offers both signature-based and behavior-based detection methods and can be used to prevent potential security threats in real-time.
- **OSSEC:** is an open-source Host-based IPS system that is designed to monitor individual hosts for potential threats. It offers real-time alerts and can be used to detect a wide range of threats, including malware, rootkits, and unauthorized access.

3.3.2 Hardware-based IPS

They are designed to run on dedicated hardware, such as a server or appliance. They are typically designed for high-performance environments and can handle large volumes of network traffic. Hardware-based IPSs often come with specialized hardware components, such as network interface cards (NICs), that are optimized for high-speed data processing. Hardware-based IPSs are often more expensive than software-based systems and can be more difficult to deploy and manage. Hardware-based IPSs can provide better performance than software-based systems because they offload the processing and analysis of network traffic from the host server. This can improve network performance and reduce the impact of IPS on host performance.

Following are some of the most famous hardware IPSs.

- **Cisco FirePower:** Cisco FirePower is a family of hardware-based IPSs that are designed for high-performance environments. They offer advanced threat

prevention capabilities and can be easily integrated with other Cisco security solutions.

- **Palo Alto Networks:** Palo Alto Networks offers a range of hardware-based IPSs, including the PA-5200 Series and the PA-800 Series. These systems are designed to provide advanced threat prevention capabilities and can be easily integrated with other Palo Alto Networks security solutions.
- **Fortinet:** Fortinet offers a range of hardware-based IPSs, including the FortiGate Series and the FortiAnalyzer Series. These systems are designed to provide advanced threat prevention capabilities and can be easily integrated with other Fortinet security solutions.
- **Juniper Networks:** Juniper Networks offers a range of hardware-based IPSs, including the SRX Series and the Sky Advanced Threat Prevention (ATP) solution. These systems are designed to provide advanced threat prevention capabilities and can be easily integrated with other Juniper Networks security solutions.

3.3.3 Hybrid IPS

It is a combination of both software-based and hardware-based IPSs. Hybrid IPSs can offer the flexibility and ease of configuration of software-based systems and the performance and scalability of hardware-based systems. Hybrid IPSs can be particularly useful for Organizations that have high-performance requirements but also need the flexibility to configure the system to their specific needs. Also IPSs comes both as software and hardware.

3.4 Components of an IPS

An IPS typically consists of several components that work together to provide comprehensive protection. These components include:

- **Sensors:** are responsible for monitoring and capturing network traffic data. They are placed at strategic points within the network, such as near network switches, routers, or firewalls, to ensure maximum visibility. In an IPS, sensors are usually in-line with network traffic, allowing them to actively monitor and intervene when necessary.
- **Analysis Engine:** processes the network traffic data captured by the sensors and employs various techniques, such as signature-based detection, anomaly-based detection, or stateful protocol analysis, to identify potential threats.
- **Policy Engine:** the policy engine is responsible for defining the rules and configurations that determine how the IPS should respond to detected threats. These rules may include actions such as blocking IP addresses, adjusting firewall rules, terminating sessions, or generating alerts.
- **Response and Prevention Mechanisms:** When the IPS identifies a potential threat, it takes appropriate actions as defined by its policy engine. These actions can include blocking traffic, resetting connections, modifying packet contents, or

notifying security personnel.

- **Management Interface:** The management interface allows administrators to configure, monitor, and manage the IPS. This interface provides visibility into the system's operation, including alerts, logs, and reporting features. It also enables administrators to update the IPS's rules, policies, and signatures.

3.5 Features of IPS

Alerting

IPS solutions generate alerts when suspicious or malicious activities are detected, enabling security teams to take swift action.

- **Real-time alerts:** IPSs can send real-time alerts to security personnel via email, SMS, or other notification methods, allowing for rapid response to potential threats.

- **Customizable alert thresholds:** to minimize false positives and ensure that security teams focus on high-priority threats, IPS solutions offer configurable alert thresholds. Organizations can set the sensitivity level and choose the types of alerts they want to receive.

- **Alert prioritization:** some IPS solutions prioritize alerts based on their severity, enabling security teams to address the most critical issues first.

Reporting

IPS solutions provide comprehensive reporting capabilities, offering insights into security incidents and trends.

- **Detailed event logs:** when an alert is triggered, IPSs typically provide detailed event logs that offer insights into the nature of the suspicious activity, such as source and destination IP addresses, timestamps, and other relevant information.

- **Visualizations and dashboards:** IPSs often include visualizations and dashboards to help security teams quickly assess the current threat landscape, identify trends, and monitor the overall security posture of the Organization.

- **Scheduled reports:** some IPS solutions allow for the creation of scheduled reports, which can be automatically generated and sent to relevant stakeholders at regular intervals. These reports can provide insights into long-term trends and help Organizations assess the effectiveness of their security measures.

- **Compliance reporting:** many IPS solutions include features to help Organizations meet various compliance requirements by generating reports tailored to specific regulations and standards, such as the General Data Protection Regulation (GDPR) or the Payment Card Industry Data Security Standard (PCI DSS).

Integration and Correlation

IPS solutions can be integrated with other security tools to provide a more comprehensive view of the Organization's security landscape:

- **Integration with SIEM systems:** by integrating with Security Information and Event Management (SIEM) systems, IPS data can be correlated with other security events and logs, providing a more holistic understanding of potential threats.

- **Threat intelligence feeds:** some IPS solutions can incorporate external threat intelligence feeds, enhancing their detection capabilities and ensuring that they stay

up-to-date with the latest attack patterns and indicators of compromise (IoCs).

Organizations and Regulatory Requirements

Organizations worldwide face various regulatory requirements that mandate the implementation of robust security measures, including the use of Intrusion Prevention System (IPS). These regulations aim to protect sensitive data, ensure the integrity and availability of IT systems, and promote a strong security posture. Below is an overview of some key Organizations and regulatory requirements related to IPS:

- **Payment Card Industry Data Security Standard (PCI DSS):** PCI DSS is a set of security standards designed to ensure the secure handling of cardholder data by merchants, payment processors, and other entities involved in processing credit card transactions. Requirement 11 of PCI DSS mandates the use of intrusion detection and prevention systems to monitor and protect the cardholder data environment.

- **Health Insurance Portability and Accountability Act (HIPAA):** HIPAA is a US federal law that sets standards for the protection of sensitive patient health information. While HIPAA does not explicitly require the use of IPS, it mandates the implementation of appropriate technical safeguards to protect electronic protected health information (ePHI). An IPS will detect and prevent any unauthorized access or other potential threats to ePHI.

- **Federal Information Security Management Act (FISMA):** FISMA is a US federal law that sets security requirements for federal agencies and their contractors. FISMA requires Organizations to implement a comprehensive security program, including the use of intrusion detection and prevention systems, to protect the confidentiality, integrity, and availability of federal information systems.

- **General Data Protection Regulation (GDPR):** GDPR is a European Union regulation that sets guidelines for the protection of personal data for EU citizens. While GDPR does not explicitly mention IPS, it requires Organizations to implement appropriate technical and Organizational measures to ensure a high level of security for personal data. Detections/Incidents will be created about any potential threats to personal data.

- **International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) 27001:** ISO/IEC 27001 is a globally recognized standard for information security management systems (ISMS). It provides a systematic approach to managing sensitive information, including the implementation of intrusion detection and prevention systems to monitor and protect the Organization's information assets.

To ensure that IPS solutions effectively support compliance with various regulatory requirements, Organizations should follow best practices, such as:

- **Risk assessment:** conduct a thorough risk assessment to identify potential threats and vulnerabilities in the Organization's IT systems. This assessment will help determine the appropriate IPS implementation to address specific risks and satisfy regulatory requirements.

- **Policy development:** establish clear policies and procedures related to the use of IPS, including incident response plans, system configuration guidelines, and ongoing maintenance requirements.

- **System integration:** integrate IPS with other security tools, such as Security Information and Event Management (SIEM) systems, firewalls, and endpoint

protection solutions, to ensure comprehensive protection and visibility across the Organization's IT environment.

- **Compliance reporting:** leverage IPS reporting capabilities to generate reports tailored to specific regulatory requirements, demonstrating the Organization's compliance with these standards.

3.6 Limitations of IPS

False positive and false negative

Similarly with IDS, false positives and false negatives are two common issues that can occur with Intrusion Prevention System (IPS) detections too.

False positives occur when the IPS system uses overly aggressive detection rules or when there is a misconfiguration in the IPS system. False positives can be problematic because they can result in legitimate traffic being blocked, which can impact network performance and availability. For those reasons, for OT environment, the use of IDS is preferred in order to avoid significant impact on the production: an IPS could affect the productivity just because a misconfiguration, for example, and this is unacceptable.

False negatives occur when the IPS system fails to identify a potential security threat. This can happen when the IPS system's detection methods are not comprehensive enough to identify all types of potential security threats, or when the IPS system is unable to keep up with the volume of network traffic. False negatives can be problematic because they can result in security threats being missed, which can lead to data breaches or other negative consequences.

IPSs rely on deep packet inspection to analyze network traffic and identify potential security threats, which can impact network performance, particularly in high-volume traffic environments.

Limitation to prevent attacks

Some limitations of an IPS in preventing attacks adequately are a consequence of configurations chosen or the type of the traffic flowing:

- **Signature-based detection limitations:** signature-based detection methods rely on known patterns or signatures of attacks to identify malicious activities. While this approach is effective against known threats, it cannot detect zero-day attacks or new variants of malware that have not yet been documented. As a result, signature-based IPS may fail to prevent previously unknown threats.

- **Anomaly-based detection limitations:** anomaly-based detection methods identify suspicious activity by comparing network traffic against a baseline of normal behavior. This approach can detect unknown threats but is prone to false positives, as legitimate network activity may sometimes deviate from the baseline. Additionally, if the baseline is not comprehensive or up-to-date, the IPS may fail to detect actual threats.

- **Resource-intensive:** IPS solutions can be resource-intensive, consuming significant processing power and bandwidth to analyze network traffic in real-time. This may lead to performance degradation, especially for high-traffic networks or during peak times. In some cases, the IPS may be unable to keep up with the traffic

volume, allowing some threats to slip through.

- **Encryption challenges:** encrypted network traffic can limit the effectiveness of an IPS because it cannot inspect the contents of encrypted packets. As a result, malicious activities concealed within encrypted traffic may bypass the IPS undetected. While some IPS solutions offer SSL/TLS inspection capabilities, this feature may introduce privacy concerns and performance overheads.
- **Evasion techniques:** attackers may employ various evasion techniques to bypass IPS detection, such as fragmentation, obfuscation, or tunneling. These tactics can make it difficult for the IPS to accurately identify and block malicious traffic, allowing the attack to succeed.
- **Configuration and maintenance challenges:** IPS solutions require proper configuration and regular maintenance to remain effective. Inaccurate configurations or outdated signatures and rules can result in decreased prevention capabilities. Additionally, keeping up with the ever-evolving threat landscape can be time-consuming and resource-intensive for security teams.
- **False positives and negatives:** as discussed earlier, IPS solutions are not immune to false positives and negatives. False positives can lead to operational disruptions, while false negatives can allow actual threats to go undetected. Balancing the trade-off between false positives and negatives is a significant challenge for IPS effectiveness.

Best practice for false positive detection

Minimizing false positive detections in Intrusion Prevention System (IPS) is essential for maintaining efficient security operations and reducing the impact on legitimate network activities. The following best practices can help Organizations reduce false positives and enhance the overall effectiveness of their IPS:

- **Regularly update signatures and rules:** ensure the IPS has latest versions of threat signatures and rules. This helps the system accurately identify known threats and reduces the likelihood of falsely identifying legitimate network traffic as malicious.
- **Fine-tune detection settings:** customize the setting to be aligned with Organization's specific network environment and risk profile. Anomaly detection thresholds must to be adjusted ad-hoc, custom rules considering the context, and signature-based detection settings must be refined to minimize false positives without compromising threat detection capabilities. Identify patterns of false positives and continuously fine-tune the IPS settings to improve accuracy.
- **Implement multiple detection techniques:** a combination of signature-based, anomaly-based, and behavior-based detection methods can improve the accuracy of threat detection. This multi-layered approach helps reduce false positives by providing a more comprehensive view of network traffic and potential threats.
- **Allowlisting and segmentation:** allowlisting trusted network traffic, applications, and IP addresses to reduce the likelihood of false positives. Additionally, segment own network into zones with different security levels, which can help prevent unnecessary IPS alerts by allowing only the required traffic to pass through.
- **Integrate with other security tools:** the integration between IPS and the other security solutions (Managed Detection and Response (MDR), Endpoint Protection Platform (EPP), etc.) improves visibility and context, enabling more accurate threat detection and reducing false positives.

Continuous monitoring and performance analysis: monitor and analyze regularly the performance of the IPS, reviewing the generated alerts and actions taken.

- **Educate the security team:** security team must be trained and knowledgeable about the IPS, its features, and the latest threat landscape. This will help them make informed decisions when configuring and fine-tuning the system, reducing the likelihood of false positives.

- **Periodic testing and validation:** periodic tests and validation exercises, such as penetration testing or red team exercises, are essential to evaluate the effectiveness of the IPS and its ability to detect threats accurately. These tests can help identify areas where false positives may be occurring and inform adjustments to the system settings.

- **Establish a feedback loop:** there needs a strong feedback loop between the security team and other stakeholders, such as network administrators and application developers, to share insights on false positives. This collaboration can lead to better understanding and adjustments that can help reduce false positives.

Chapter 4

Comparison between IDS and IPS

Similarities between IDS and IPS

Despite some differences in their functionality showed in the next paragraph, they share several similarities, as outlined below:

- **Threat detection:** both IDS and IPS are designed to monitor network traffic and detect potential security threats, such as unauthorized access, malware, or other malicious activities. Their primary goal is to identify suspicious behavior or patterns in the network traffic to help maintain a secure network environment.

- **Signature-based detection:** both systems employ signature-based detection methods, which rely on known patterns or signatures of attacks to identify malicious activities. This approach is effective against known threats but has limitations in detecting zero-day attacks or new malware variants.

- **Anomaly-based detection:** both IDS and IPS can use anomaly-based detection methods, which identify suspicious activity by comparing network traffic against a baseline of normal behavior. This approach can detect unknown threats but may be prone to false positives due to variations in legitimate network traffic.

They can also use Access Control Lists (ACLs), which allow or block traffic based on predefined rules. ACLs can be used to block traffic from known malicious IP addresses or to restrict access to specific resources.

- **Alert generation:** both systems generate alerts upon detecting potential security threats. These alerts provide valuable information about the nature of the detected threat, its source, and other relevant details, allowing security teams to take appropriate action in response to the identified incidents.

- **Integration with other security tools:** both IDS and IPS can be integrated with other security solutions, such as Security Information and Event Management (SIEM) systems, firewalls, and endpoint protection platforms.

This integration improves visibility and context, enabling more accurate threat detection and a comprehensive security posture.

- **Monitoring and analysis:** both systems require continuous monitoring and analysis to maintain their effectiveness. Security teams need to regularly review the generated alerts, update signatures and rules, and fine-tune detection settings to ensure optimal performance and threat detection accuracy.

- **Configuration and maintenance:** proper configuration and regular maintenance are essential for both IDS and IPS to remain effective in detecting and preventing threats. Inaccurate configurations or outdated signatures and rules can

result in decreased prevention capabilities and an increased likelihood of false positives and negatives.

Differences between IDS and IPS

Differences between those two systems are the following:

- **Response to threats:** the primary difference between IDS and IPS lies in their response to detected threats. An IDS is a passive system that only monitors and detects potential threats, generating alerts for security teams to respond to. In contrast, an IPS is an active system that not only detects threats but also takes preventive action to block or mitigate them in real-time.

In the case of a DoS attack, for example, the IDS would identify an abnormally high volume of incoming traffic from multiple sources, which matches the signature or pattern of a DDoS attack. But since the IDS is a passive system, it does not take any direct action to stop the attack. Instead, it generates an alert to inform the security team of the potential threat.

The detection management would be in care of the security team that receives the alert and manually investigates the incident. They may take actions such as blocking the attacking IP addresses, adjusting firewall rules, or implementing traffic filtering to mitigate the attack.

In the case of the IPS, because it is an active system that can take direct action to prevent or mitigate the attack in real-time, it could automatically block the attacking IP addresses, adjust firewall rules, or implements traffic filtering to limit the impact of the attack on the targeted network resources.

Concurrently, the IPS generates an alert to inform the security team of the detected threat and the actions taken to mitigate it.

The security team would be alert by the IPS about the final action and they can then review the alert and follow up with any additional actions or investigations, if necessary.

- **Network placement and performance:** IDS solutions are deployed in passive mode or out-of-band, meaning they monitor a copy of the network traffic without directly affecting the actual traffic. They receive mirrored traffic from network taps, span ports, or network packet brokers. This approach allows IDS to analyze network traffic and detect threats without introducing latency or reducing throughput, as it does not interact with the live traffic.

Since the IDS operates independently from the actual traffic flow, its resource usage and processing capabilities have minimal impact on network performance. This passive monitoring ensures continuous threat detection with little to no disruption of network operations.

IPS solutions are deployed in active mode or in-line, meaning they are placed directly in the path of network traffic. They intercept, analyze, and process network packets in real-time to identify and prevent threats. When an IPS detects a potential threat, it can take immediate action, such as blocking, modifying, or rerouting the traffic.

Due to the in-line placement and active operation, an IPS system can have a more significant impact on network performance, particularly in terms of latency and throughput. As the IPS processes and analyzes network traffic in real-time, it can introduce delays, especially if it is not adequately optimized or configured.

Furthermore, the IPS must be capable of handling the full bandwidth of the network traffic to avoid becoming a bottleneck.

Additionally, the resource usage and processing capabilities of an IPS system can directly affect network performance. If the IPS is overwhelmed by the volume of traffic or the complexity of the analysis, it may struggle to keep up with network demands, further impacting performance.

- **False positives and operational disruption:** false positives in an IDS are instances where the system generates an alert for an event that it mistakenly identifies as malicious, even though the activity is actually benign. Since IDS operates in passive mode, it only monitors and analyzes network traffic without taking any direct action to prevent or mitigate the detected threats. The impact of false positives in an IDS is mainly on the workload of the security team, who must review and investigate these alerts to determine their validity.

While false positives in an IDS can lead to wasted resources and increased response times for security teams, they do not directly disrupt network operations or affect legitimate traffic.

In contrast, false positives in an IPS can have a more significant impact on network operations due to the system's active mode of operation. When an IPS incorrectly identifies benign traffic as malicious, it may take preventive actions, such as blocking, modifying, or rerouting the traffic. These actions can directly disrupt network operations and affect legitimate users and services.

Since an IPS operates in-line and actively interacts with network traffic, false positives can lead to service outages, reduced application performance, and other operational issues. These disruptions can be costly and time-consuming to diagnose and resolve, especially if the root cause is not immediately apparent.

Chapter 5

Suitable scenarios for IPS/IDS

Strength and weakness of both systems are clear explained in past section of this paper.

Because their dissimilarities, there are different scenarios where one is more suitable than the other one, for example, an IDS should better than an IPS for:

- **Internal Network Monitoring:** IDS can be effective at detecting insider threats, policy violations, or compromised devices within the Organization's internal network, where the primary goal is monitoring and alerting rather than active prevention.
- **Historical Analysis and Forensics:** IDS is well-suited for collecting and analyzing historical network data, which can be invaluable for forensic investigations, incident response, and fine-tuning security policies.
- **Attack Detection in High Availability Environments:** in environments where maintaining network uptime and minimizing latency is critical, IDS provides passive monitoring without directly affecting network traffic. This can be especially useful in detecting attacks on systems that cannot tolerate any performance impact or disruptions due to false positives.
- **Compliance and Regulatory Requirements:** some Organizations may be required to implement IDS to meet specific regulatory or compliance requirements that mandate continuous network monitoring and threat detection. IPS may be more suitable instead for:
- **Real-time Threat Prevention:** IPS is better suited for scenarios where real-time prevention of attacks is crucial, such as protecting public-facing web applications or critical infrastructure. By actively blocking or mitigating threats, an IPS can help minimize the damage caused by attacks.
- **Zero-day Attack Mitigation:** IPSs often include advanced features like protocol analysis, traffic normalization, and behavioral analysis, which can help identify and block zero-day attacks that may not have known signatures.
- **Network Segmentation and Access Control:** IPS can be used to enforce network segmentation and access control policies, preventing unauthorized access and the lateral movement of attackers within the network.
- **Protection against Automated Attacks:** IPS is particularly effective against automated attacks, such as botnets and worms, where real-time prevention is essential to minimize the spread and impact of the threat.

Chapter 6

Limitations of IDS/IPS exploited by an attacker

Evasion tactics are techniques used by attackers to bypass Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) to avoid detection and carry out their malicious activities. These tactics exploit weaknesses and limitations in IDS/IPS technologies, making it challenging for security systems to identify and prevent attacks. Some common evasion tactics employed by attackers include:

- **Fragmentation:** attackers can fragment packets into smaller pieces, which makes it difficult for an IDS/IPS to reassemble and inspect them. This tactic exploits the fact that many IDS/IPS solutions struggle to process fragmented packets efficiently or accurately.

- **Encryption:** by encrypting the payload of malicious traffic, attackers can hide the true nature of the content, making it challenging for signature-based IDS/IPS to detect known attack patterns. This tactic is particularly effective when combined with encrypted communication channels, such as HTTPS or VPN connections.

- **Obfuscation:** attackers can use various techniques to obfuscate the malicious code or traffic, such as encoding, compressing, or altering the payload's syntax. These modifications can make it difficult for signature-based IDS/IPS to match the payload to known attack patterns.

- **Protocol violations:** some attackers exploit ambiguities or vulnerabilities in network protocols to bypass IDS/IPS. For example, they might craft packets with incorrect header information, manipulate packet timing, or use overlapping TCP segments. These tactics can cause IDS/IPS to misinterpret or ignore the malicious traffic.

- **Low and slow attacks:** attackers can evade detection by sending malicious traffic at a slow rate or in small bursts, making it difficult for IDS/IPS to identify the attack. These tactics often target application layer vulnerabilities and can bypass threshold-based detection mechanisms.

- **Insertion and evasion:** attackers can manipulate network traffic in such a way that the IDS/IPS and the target system interpret the traffic differently. For example, an attacker might send overlapping packets with contradictory information, causing the IDS/IPS to discard the packet while the target system processes it.

- **Using legitimate services:** attackers can leverage legitimate services, such as cloud storage, social media, or content delivery networks, to host and distribute

their malicious payloads. This tactic makes it difficult for IDS/IPS to distinguish between legitimate and malicious traffic, leading to potential false negatives.

By Organization side must be deployed a multi-layered defense strategy, which may include:

- regularly updating IDS/IPS signatures and rules;
- employing anomaly-based detection techniques to identify unusual patterns in network traffic;
- incorporating advanced inspection technologies, such as deep packet inspection (DPI) and stateful protocol analysis;
- implementing encryption and decryption capabilities in IDS/IPS solutions;
- integrating IDS/IPS with other security systems, such as firewalls, Endpoint Protection Platforms (EPP), and Security Information and Event Management (SIEM) systems;
- regularly monitoring and analyzing network traffic to identify emerging threats and improve security policies.

Chapter 7

Implementation of IDS and IPS

7.1 Best practices for implementing an IDS/IPS

Implementing those systems requires careful planning and preparation to ensure that the system is configured properly and integrated adequately with other security technologies. Some best practices for implementing an IDS/IPS with planning and preparation include:

- Define the scope and goals: before implementing systems, it is important to define the scope and goals of the system. This includes identifying the types of threats will be used to detect, the systems and networks that will be monitored, and the expected performance and availability of the system.
- Understand own Network: a thorough understanding of the network topology, assets, and traffic patterns is essential for configuring and tuning them. This knowledge helps to identify critical systems, prioritize monitoring, and set appropriate rules and thresholds for alerting.
- Conduct a risk assessment: a risk assessment can help identify potential vulnerabilities and threats to the network and provide guidance on the types of threats should be configured to detect. A risk assessment can also help prioritize the implementation of the technologies based on the level of risk associated with different systems and networks.
- Select the Right IDS Type: choose between a Network-based or a Host-based based on the Organization's specific needs and security goals. The first one monitors network traffic, while the second examines activities on individual hosts. Hybrid solutions that combine both approaches may also be appropriate in some cases.
- Use tap or span ports for network-based sensors: Network-based sensors should be connected to tap or span ports on network devices to monitor network traffic without disrupting network operations. It is important to ensure that tap or span ports are configured correctly to provide accurate and complete network traffic.

- Determine the appropriate number of sensors: the number of sensors required will depend on the size and complexity of the network being monitored. Generally, more sensors are required for larger and more complex networks. However, it is important to balance the number of sensors with the cost of the system and the workload of security analysts.
- Properly Place Sensors: place sensors strategically throughout the network to maximize visibility into potential threats. They must be located at critical network points, at choke points, in front of critical assets, and within network segments to detect both external and internal threats. If the network is segmented, sensors should be placed at the boundaries between network segments to monitor traffic between segments. This can help detect lateral movement of attackers within the network.
- Integrate with Other Security Systems: for a more comprehensive security posture, integrate them with other security tools, such as firewalls, Security Information and Event Management (SIEM) systems, and Endpoint Detection and Response (EDR) solutions. This allows for better correlation of events, improved threat intelligence, and more efficient incident response.
- Configure them to capture and analyze the appropriate network traffic: they should be configured to capture and analyze the appropriate network traffic to detect potential threats. This includes configuring them to monitor specific network segments, protocols, and ports, and to filter out unnecessary traffic.
- Define alerting and reporting criteria: it is important to define alerting and reporting criteria, including the types of alerts that will be generated, who will receive alerts, and how alerts will be investigated and responded to. This includes defining thresholds for alert generation and tuning them to reduce false positives.
- Conduct regular testing and validation: regular testing and validation is essential to ensure that the system is working properly. This includes conducting penetration testing and vulnerability assessments, as well as testing the response procedures to ensure that the system is prepared to respond to potential threats.

7.2 Challenges in implementing IDS and IPS

One of the biggest challenges in implementing Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) is managing the resource requirements of these systems. As wrote, IDS and IPSs can be resource-intensive, requiring significant processing power, memory, and storage to correctly monitor and protect networks against cyber threats. Some of the challenges related to resource requirements when implementing IDS and IPS include:

Hardware requirements: IDS and IPSs require powerful hardware to successfully monitor and protect networks. This includes servers, switches, and routers that are capable of handling large volumes of network traffic and processing data

in real-time.

Network bandwidth requirements: IDS and IPSs can generate a significant amount of traffic on the network, particularly if they are configured to capture and analyze all network traffic. This can impact network performance and require additional bandwidth to support the system.

Storage requirements: IDS and IPSs generate a large amount of data, including log files, alerts, and network traffic captures. This data must be stored for analysis and reporting, which can require significant amounts of storage space.

Processing requirements: IDS and IPSs require significant processing power to analyze network traffic and generate alerts. This can impact overall system performance and require additional processing power to support the system.

Management and maintenance requirements: IDS and IPSs require ongoing management and maintenance, including software updates, signature and rule updates, and regular testing and validation. This can be time-consuming and require dedicated personnel to manage the system in a correct manner.

Another key challenge in implementing Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) is the complexity of configuration.

IDS and IPSs can be very complex, requiring detailed configuration to ensure that they are effective in detecting and preventing intrusions:

- Configuring the system to detect and prevent the appropriate types of threats: IDS and IPSs must be configured to detect and prevent the specific types of threats that the Organization is most likely to face. This requires a detailed understanding of the Organization’s network infrastructure and the types of threats that are most common in their industry or geographic region.
- Setting appropriate thresholds for alerts and blocking: IDS and IPSs generate a large number of alerts and blocking events, which can quickly overwhelm security teams. Setting appropriate thresholds for alerts and blocking is critical to ensure that the system is effective in detecting and preventing threats, without generating an excessive number of false positives. The Company must consider a large amount of alert and event data, which can be challenging to manage.
- Tuning the system to reduce false positives: false positives can be a significant burden on security teams, as they require additional investigation and can distract from real threats. Tuning the system to reduce false positives requires a detailed understanding of the network infrastructure and the types of traffic that are normal for the Organization.
- Integrating the system with other security technologies: IDS and IPSs are most effective when they are integrated with other security technologies. This requires detailed configuration to ensure that the systems are working together effectively and efficiently. Moreover, systems must be compatible to ensure effective integration so that all systems are working together correctly. This requires careful selection of the appropriate systems and vendors, and thorough testing to ensure compatibility.

Consequently, the Company must consider a large amount of data that could be resulted.

- Ensuring that data is shared securely and confidentially: sharing data between IDS and IPSs and other security technologies must be done securely and confidentially to protect sensitive information. This requires the implementation of appropriate security controls to ensure that data is not compromised during transmission or storage. For all logs (that are the inputs/outputs of security devices), such as successful logins, failed logins, and logouts, in conjunction with the associated timestamp, must be ensured principles of integrity and inalterability, as well as local data regulation requirements, using hashing techniques and secure network tunnels.
- Managing signature and rule updates: IDS and IPSs rely on signatures and rules to detect and prevent threats. Managing signature and rule updates can be complex, particularly if the Organization is using multiple systems from different vendors and if the Organization has a wide perimeter.

Chapter 8

OWASP

8.1 What is OWASP

The Open Web Application Security Project (OWASP) is a nonprofit Organization dedicated to improving the security of software, particularly web applications. Founded in 2001, OWASP aims to create a global community of security professionals, developers, and Organizations that collaborate on creating open-source resources, tools, and guidelines to enhance the security of web applications. OWASP offers a wealth of resources, including a OWASP Top Ten, a regularly updated list of the ten most critical web application security risks. The Top Ten project is a widely recognized resource that helps Organizations prioritize and address the most common and impactful web application vulnerabilities.

It offers also best practices and guidelines with a variety of documentation on secure coding practices, security testing, and other aspects of web application security. These resources offer practical advice and guidance to help developers and Organizations build more secure web applications.

The following functions are also available on the web-site:

- **Open-source tools:** OWASP develops and maintains several open-source security tools to assist with various aspects of web application security, such as vulnerability scanning, penetration testing, and secure coding.
- **Educational resources:** OWASP offers training materials, workshops, and conferences focused on web application security, providing opportunities for professionals to learn and share knowledge.
- **Local chapters and events:** OWASP has local chapters worldwide that organize meetings, presentations, and networking opportunities for professionals interested in web application security.

8.2 OWASP Top Ten

The OWASP Top Ten is a list of the most critical web application security risks, which is periodically updated to reflect the current threat landscape:

1. Broken object level authorization
2. Broken authentication
3. Broken object property level authorization
4. Unrestricted resource consumption
5. Broken function level authorization
6. Server side request forgery
7. Security misconfiguration
8. Lack of protection from automated threats
9. Improper asset management
10. Unsafe consumption of APIs

1. Broken object level authorization

Object level authorization, typically implemented at the code level for user validation, is a control method to restrict access to objects. When authorization at the object level is not properly enforced, it can expose systems. Such a vulnerability was uncovered at Uber by sending API requests including user phone numbers to get access to tokens and manipulating systems.

Attack vectors: attacks exploit API endpoints by manipulating object IDs that are sent within a request. This issue is unfortunately fairly common in API-based applications when server-side components do not track the full client state but rely more on object IDs.

Security weakness: authorization and access controls are complex. Even with proper protocols and configurations, developers sometimes forget to use authorization checks before accessing sensitive objects. These states do not play well with automatic testing.

2. Broken authentication

Authentication endpoints are vulnerable to a number of risks, including brute force attacks, credential stuffing, weak encryption keys, and connections to other microservices without requiring authentication.

Attack vectors: because these endpoints may be accessible to people outside an Organization, there are several potential threats. It's easy to fail to fully protect the entire boundary for authentication or implement the proper security protocols.

Security weakness: OWASP points to two specific issues with endpoint authentication:

- a lack of protection mechanisms that include extra levels of protection;
- incorrect implementation of authentication mechanisms or using the wrong mechanism for applications.

3. Broken object property level authorization

When accessing an object via an API, users must be validated to ensure they have

the authority to access certain object properties. Broken authorization at the object property level can allow unauthorized users to access and change objects.

Attack vectors: threat actors exploit vulnerable API endpoints to read, change, add, or delete object property values for objects that should not be available to attackers.

Security weakness: even when developers provide validations for user access to functions and objects, they may not validate if users are allowed to access specific properties within objects.

4. Unrestricted resource consumption

Without restrictions on API requests, attackers sending multiple requests or flooding resources can implement denial of service (DoS) attacks and also cause financial damage for those using pay-per-request billing.

Distributed denial of service (DDoS) attacks have grown significantly over the past two years, up as much as 60%.

Attack vectors: APIs can be exploited by sending multiple, concurrent requests to APIs that do not limit interactions.

Security weakness: APIs often do not limit activities such as execution timeouts, maximum allowable memory, the number of operations in client requests, or implementing third-party spending limits. Even with logging, it's easy for malicious activity to go unnoticed in the early stages.

5. Broken function level authorization

When function level authorization allows users to access administrative endpoints, they can perform sensitive actions.

Attack vectors: attackers can uncover API flaws because they are more structured and predictable in access methodology, and then they can send legitimate API calls to endpoints that they should not be able to access. In some cases, it can be as simple as guessing the endpoint URL and changing “users” to “admins” in strings.

Security weakness: modern applications contain plenty of roles, groups, and complex user hierarchies. Users may have different roles for different areas or objects, so it can be challenging to monitor.

6. Server side request forgery

Server side request forgery (SSRF) can happen when an API fetches a remote resource without first validating the URL supplied by users. Servers can be used as proxies to hide malicious activity. Researchers recently found four such instances of SSRF vulnerabilities with Azure API management, which have since been patched.

Attack vectors: attackers find an API endpoint that receives a universal resource identifier (URI) and force the application to send a request to an unexpected destination — even when destinations are protected via a firewall or VPN.

Security weakness: application development often includes accessing URIs provided by the client, and server-side data retrieval generally is not logged or monitored.

7. Security misconfiguration

Hardening security for the API stack should be a top priority for developers, but

permissions are often improperly, or inconsistently, applied across cloud services. In other cases, security patches and software are out of date. There have been several high-profile instances where companies failed to protect their cloud resources properly, such as the United States Army Intelligence and Security Command, and in that case the unprotected data included some files classified as top secret.

Attack vectors: threat actors actively search for unpatched flaws and unprotected files or directories, and they attack common endpoints to map systems and gain unauthorized access. Discrepancies in the way requests are handled and processed leave attack vectors open.

Security weakness: misconfigurations can happen at any level from network to application. Legacy options and unnecessary services can also create additional attack pathways.

8. Lack of protection from automated threats

Cybercriminals and other threat actors are increasingly evolving their tactics, and APIs are prime targets.

Automation is cheap and widely available on the dark web. The APIs themselves may not have flaws or bugs, but the underlying business flow may be vulnerable to excessive activity.

Attack vectors: attackers learn API models and business flows and then exploit them using automated tools. For example, the use of automated tools and botnets can bypass rate limiting by spreading requests over IP addresses.

Security weakness: the challenge here is that each request may appear legitimate, so it will not be identified as an attack. However, these automated attacks can flood systems and prevent legitimate users from access.

9. Improper inventory management

APIs across applications can be quite complex and interwoven. Connectivity with third parties increase threat exposure, and often multiple versions of APIs may be left running that are unmanaged. Outdated or missing documentation can make it challenging to keep track of everything.

Attack vectors: attackers may access older API versions or endpoints that are unpatched. They may also gain access through third parties.

Security weakness: a lack of inventory or asset management can lead to a host of problems, including unpatched systems. API hosts may be exposed through microservices, which make applications independent in many cases. A lack of a systematic and documented way to deploy, manage, and retire APIs can lead to different security weaknesses.

10. Unsafe consumption of APIs

When working with well-known third parties and suppliers, generally data received are trusted and might employ less stringent security standards. Yet, if threat actors can breach third parties, they may be able to cause damage through APIs connected to. Today, as many as half of data breaches occur because of third-party connectivity.

Attack vectors: the exploitation of security flaws in APIs occurs when developers trust — but do not verify and fully protect — endpoints that interact with APIs. For example, they may not place appropriate limitations on resources, validate

redirects, or validate/sanitize data requests from APIs before processing.

Security weakness: security weaknesses often arise when weaker security models are applied to API integrations, especially in areas such as transport security, input validation, data validation, authentication, and authorization. This exposes Organizations to unauthorized access and malicious injections.

8.3 IDS-IPS and OWASP

A significant correlation exists between OWASP and IDS/IPS and it is clarified below:

Informing detection rules: IDS/IPS use various detection techniques to identify malicious activities. By understanding the OWASP Top Ten risks and other web application security best practices, security teams can develop more accurate and effective detection rules tailored to web application attacks for their IDS/IPSs. This can improve an Organization's ability to detect and prevent web-based threats.

Enhancing threat intelligence: OWASP provides valuable information on common web application vulnerabilities and attack techniques. By incorporating this knowledge into their IDS/IPS, Organizations can improve their overall threat intelligence and better understand the web application risks they face.

Securing web applications: IDS/IPS are just one layer of defense in an Organization's security posture. By following OWASP guidelines and recommendations for secure coding and web application security, Organizations can reduce the attack surface and make it more difficult for attackers to exploit vulnerabilities. A more secure web application can reduce the number of alerts generated by the IDS/IPS, allowing security teams to focus on more critical issues.

Improving incident response: when an IDS/IPS detects a potential web application attack, having knowledge of OWASP best practices can help security teams better understand the attacker's techniques and objectives. This can enable faster and more effective incident response and remediation.

Measuring security effectiveness: OWASP resources can be used as a benchmark to evaluate the effectiveness of an Organization's web application security measures, including the performance of their IDS/IPS. By comparing the Organization's security posture against OWASP recommendations, security teams can identify gaps and areas for improvement.

Chapter 9

IDPS Case Studies

There are many case studies of Organizations that have implemented Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). These case studies offer valuable lessons learned regarding common challenges and solutions when implementing IDS and IPS.

Challenge: false positives and alert fatigue.

One of the most common challenges with IDS and IPSs is the generation of false positives and alert fatigue. This can be addressed by fine-tuning the system to reduce false positives, setting appropriate alert thresholds, and implementing automation to reduce the workload on security teams.

Challenge: resource requirements.

IDS and IPSs can be resource-intensive, requiring significant processing power, memory, and storage. This can be addressed by carefully assessing the network and the types of threats that the system is being implemented to address and investing in appropriate hardware and infrastructure.

Challenge: complexity of configuration.

IDS and IPSs can be complex, requiring detailed configuration to ensure that they are effective in detecting and preventing intrusions. This can be addressed by conducting a thorough assessment of the network, working with vendors to ensure that the system is configured correctly, and regularly testing and validating the system.

Challenge: integration with other security technologies.

IDS and IPSs are most effective when they are integrated with other security technologies. However, integrating these systems can be complex and challenging. This can be addressed by carefully selecting appropriate systems and vendors, configuring the systems to work together properly, and implementing appropriate security controls to ensure that data is shared securely and confidentially.

Following examples of Organizations that have successfully implemented IDS and IPSs.

IBM: IBM implemented a global IDS system to monitor its network for potential threats. The system was designed to be scalable and flexible, allowing it to easily adapt to changes in the network. IBM also implemented automation to reduce the workload on security teams and improve response times.

NASA: NASA implemented an IPS system to protect its critical systems from cyber threats. The system was configured to detect and prevent specific types of threats, and was integrated with other security technologies, such as firewalls and

antivirus software.

NASA also implemented a comprehensive testing and validation program to ensure that the system was effective in preventing intrusions.

University of Alabama: The University of Alabama implemented an IDS system to monitor its network for potential threats. The system was configured to reduce false positives and generate alerts only for high-priority threats. The university also implemented automation to reduce the workload on security teams and improve response times.

Netflix: Netflix implemented an IPS system to protect its streaming service from cyber threats. The system was designed to detect and prevent specific types of threats, and was integrated with other security technologies, such as firewalls and SIEMs. Netflix also implemented automation to reduce the workload on security teams and improve response times. The system was successful in protecting Netflix's streaming service from cyber threats and has been in operation for several years.

Chapter 10

Future of IDS and IPS

10.1 Emerging trends in IDS and IPS

One emerging trend in IDS and IPS is the use of machine learning (ML) and artificial intelligence (AI) to enhance the effectiveness of these systems.

Machine learning and AI can be used in several ways to improve IDS and IPS, including:

- **Improved threat detection:** ML and AI can be used to improve the accuracy of threat detection by analyzing large volumes of data and identifying patterns and anomalies that may indicate a cyber attack. By analyzing network traffic, system logs, and other data sources, ML and AI algorithms can identify potential threats that may be missed by traditional signature-based detection methods.

- **Reduced false positives:** ML and AI can also be used to reduce the number of false positives generated by IDS and IPSs. By analyzing data and identifying patterns, ML and AI algorithms can distinguish between legitimate traffic and potential threats, reducing the number of false positives and reducing the workload on security teams.

- **Improved response times:** ML and AI can be used to improve response times by automating certain tasks, such as alert triage and incident response. By using ML and AI algorithms to analyze data and generate alerts, security teams can respond to threats more quickly and accurately.

- **Adaptive security:** ML and AI can be used to create adaptive security systems that can learn and adapt to new threats over time. By analyzing data from past threats and identifying patterns, ML and AI algorithms can be used to create models that can proactively detect and prevent new and emerging threats.

Nowadays there are some concrete examples of how ML and AI are being used in IDS and IPS:

Cisco: Cisco's Next-Generation Intrusion Prevention System (NGIPS) uses machine learning to improve threat detection and reduce false positives. The system analyzes network traffic to identify potential threats and uses ML algorithms to distinguish between legitimate traffic and potential threats, reducing the number of false positives generated by the system.

Darktrace: Darktrace's Enterprise Immune System uses AI to create an adaptive security system that can learn and adapt to new threats over time. The system

analyzes network traffic and system logs to identify potential threats and uses AI algorithms to create models that can proactively detect and prevent new and emerging threats.

Fortinet: Fortinet's FortiGate IPS uses ML to improve threat detection and reduce false positives. The system analyzes network traffic to identify potential threats and uses ML algorithms to distinguish between legitimate traffic and potential threats, reducing the number of false positives generated by the system.

Another emerging trend in Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) is the use of **cloud-based solutions**. Cloud-based IDS and IPS offer several benefits over traditional on-premises solutions, including improved scalability, flexibility, and cost-effectiveness.

Cloud-based IDS and IPS can leverage the scale and processing power of the cloud to analyze large volumes of data and identify potential threats more quickly and accurately than traditional on-premises solutions.

Another parameter impacted is the reduced latency due to their location closer to the network edge, and also improving response times. This is particularly important for Organizations with distributed networks or geographically dispersed users. Scalability, flexibility, and cost-effectiveness are strong point of cloud-based solution compared with traditional on-premises solutions, as they eliminate the need for expensive hardware and infrastructure.

Additionally, cloud-based solutions typically offer pay-as-you-go pricing models, allowing Organizations to only pay for the resources they use.

More and more Companies is offering those kind of products and following are some examples:

- **Amazon Web Services (AWS):** AWS offers a cloud-based IDS and IPS solution called Amazon GuardDuty. The solution uses machine learning and threat intelligence to identify potential threats and provides real-time alerts and remediation recommendations.

- **Microsoft:** Microsoft offers a cloud-based IPS solution called Azure Firewall. The solution provides centralized network security management and can be integrated with other Microsoft security products, such as Azure Security Center and Azure Sentinel.

- **Google Cloud:** Google Cloud offers a cloud-based IDS and IPS solution called Cloud IDS. The solution uses threat intelligence to detect potential threats and provides real-time alerts and remediation recommendations.

From the perspective of integration with other security technologies, emerging trends are regarding **Security Orchestration, Automation, and Response (SOAR)**. SOAR platforms can be used to integrate IDS and IPS with other security technologies, such as firewalls, SIEMs, and endpoint protection platforms. This can enable automated incident response workflows, reducing the workload on security teams and improving response times.

The same is about **Endpoint Detection and Response (EDR)**.

Moreover, as more Organizations are moving to the cloud, integration with Cloud Security Posture Management (CSPM) solutions is becoming increasingly important. By integrating IDS and IPS with CSPM solutions, Organizations can improve

threat detection and response in cloud environments and reduce the risk of misconfigurations and other cloud security issues.

Also, the “world” of **Identity and Access Management (IAM)**, that control access to critical systems and data is becoming more integrated with IDS and IPSs, by monitoring user activity and identifying potential threats more precisely.

Most notorious cases and vendors have already considered this:

- **Palo Alto Networks**: Palo Alto Networks offers an integrated security platform, including IDS and IPS, that can be integrated with other security technologies, such as firewalls, SIEMs, and endpoint protection platforms. The platform also includes automation and orchestration capabilities, enabling automated incident response workflows.

- **Cisco**: Cisco’s SecureX platform integrates IDS and IPS with other security technologies, such as firewalls and SIEMs, and provides a centralized view of security alerts and threats. The platform also includes automation and orchestration capabilities, reducing the workload on security teams and improving response times. -

IBM: IBM Security offers an integrated security platform, including IDS and IPS, that can be integrated with other security technologies, such as SIEMs and IAM solutions.

The platform also includes automation and orchestration capabilities, enabling automated incident response workflows.

10.2 Limitations of IDS and IPS technologies in the evolving cyber landscape

In addition to the limitations already wrote in previous paragraph about specific technologies, there are others resulting from the increasing sophistication of some types of attacks. Those end up to remark on precisely those boundaries.

Social engineering: APTs often use social engineering tactics to gain access to systems and data. These tactics may include phishing emails, spear-phishing, or other forms of social engineering that are difficult to detect and prevent using IDS and IPSs.

Insider threats: APTs may be carried out by insiders who have legitimate access to systems and data. These attackers may be difficult to detect and prevent using IDS and IPSs, as they may not exhibit the same types of behavior as external attackers.

This is true especially because not all insider threats are malicious in nature. For example, an employee may accidentally delete important data or introduce a virus into the network. IDS and IPSs may not be able to detect these types of threats, as they may not exhibit the same types of behavior as malicious insiders.

To address these limitations, Organizations may need to consider alternative security measures, such as network traffic analysis (NTA), or user and entity behavior analytics (UEBA). These solutions can be used in conjunction with IDS and IPS, that may not be able to monitor user behavior at a granular enough level to detect insider threats, to provide a more comprehensive security posture that is better equipped to detect and prevent APTs.

Alternatively, Next-generation IDS and IPS technologies use behavioral analysis to detect and prevent advanced cyber threats. Behavioral analysis involves monitoring user behavior, network traffic, and other factors to identify patterns that may indicate a potential threat.

This approach can be more effective at detecting advanced threats that traditional signature-based detection methods may miss.

The latter two are not the only weaknesses that might emerge from the future conversion of the attacks.

Adversarial attacks: Adversarial attacks involve manipulating or subverting machine learning algorithms to evade detection. Adversarial attacks can be particularly effective against IDS and IPSs that use machine learning algorithms to identify potential threats. Further exploration and study of adversarial attacks could help improve the robustness of IDS and IPSs and help them better withstand these types of attacks.

IoT security attacks: The Internet of Things (IoT) is a rapidly growing field that is expected to include billions of connected devices in the coming years. However, the security of these devices is often lacking, leaving them vulnerable to cyberattacks. Further exploration and study of IoT security could help improve the security of these devices and enable IDS and IPSs to better detect and prevent attacks on IoT devices.

Chapter 11

IDPS comparison

It is known that such security products are available in the market in both hardware and software formats. Considering the objective of this paper, it is considered important to present the software products in circulation that will then be evaluated in order to use them with VEREFOO.

Along with the product name, the hardware requirements necessary for the proper functioning of the product will also be specified.

Snort is an open-source Network Intrusion Detection System (NIDS) and Intrusion Prevention System (IPS) created by Martin Roesch in 1998.



Figure 11.1. Snort logo

It is designed to monitor network traffic in real-time, identify malicious packets, and detect various types of attacks and intrusions. Snort uses a rule-based language to describe traffic patterns and can analyze protocols, payloads, and headers of IP packets.

Snort uses a powerful and flexible rule-based language to define patterns of suspicious network traffic. Users can write their own rules or use a vast library of pre-existing rules provided by the Snort community. It is available for various operating systems, including Linux,

Windows, and MacOS, making it a versatile tool for organizations with diverse IT environments.

Snort can operate in three different modes:

- **Sniffer mode:** in sniffer mode, Snort simply captures the packets on the network and displays them on the console in a human-readable format. This mode is useful for basic network troubleshooting and analysis. To run Snort in sniffer mode, use the following command: "snort -vde".

- **Packet logger mode:** in packet logger mode, Snort captures packets and saves them to a file on disk for later analysis. This mode is useful for storing network traffic data for forensic purposes or for conducting in-depth analysis of network activity. To run Snort in packet logger mode, use the following command: \$snort -l <lo_directory> -b.

- **Network IDS (NIDS) mode:** in NIDS mode, Snort analyzes network traffic in

real-time, comparing it against a set of predefined rules that describe known malicious traffic patterns. If a match is found, Snort generates an alert or takes other appropriate actions, such as logging the event or blocking the traffic. This mode is the most advanced and widely used mode, as it provides real-time detection and prevention of network-based attacks. To run Snort in NIDS mode, use the following command: `$snort -c <path_to_snort_configuration_file>.$`

Snort is capable of real-time traffic analysis and packet logging on the network. It performs protocol analysis, content searching/matching and detects a large amount of attacks and probes, such as buffer overflows, port scans, CGI attacks and attempts to breach the system.

In Network Intrusion Detection System (NIDS) mode, Snort actively analyzes network traffic in real-time, comparing it against a set of predefined rules that describe known malicious traffic patterns. If a match is found, Snort generates an alert and logs the event. Snort can analyze various network protocols, including IP, TCP, UDP, ICMP, and others. It can also inspect the payloads of packets to search for known signatures of malware or other malicious content.

Alerts can be logged in various formats, including Unified2, Fast and Full alert and Syslog.

Unified2 is a binary format designed for high-performance logging and is the recommended format for production environments. Unified2 logs can be processed by external tools like Barnyard2, which can convert them into other formats or forward them to other systems, such as a Security Information and Event Management (SIEM) system.

Fast alert is a format logs a single line per alert, including a timestamp, source and destination IP addresses, and a brief description of the alert. Fast alert logs are more concise and easier to read than full alerts but provide less detail.

Full alert is format logs more detailed information for each alert, including the complete packet header, the Snort rule that triggered the alert, and additional metadata.

About Syslog, Snort can also send alerts to the system's syslog daemon, which can forward them to remote syslog servers if necessary.

Real-time packet logging: in packet logger mode, Snort captures packets on the network and saves them to a file on disk for later analysis. This mode is useful for storing network traffic data for forensic purposes or for conducting in-depth analysis of network activity. These packet logs are typically saved in the "pcap" (packet capture) format, which can be later analyzed using tools like Wireshark or tcpdump. To enable packet logging, use the `-$l <log_directory>$` and `-b` options when starting Snort.

Moreover Snort can log decoded packet data in human-readable ASCII format. This can be useful for understanding the contents of packets during troubleshooting or analysis.

There can be set also a custom logging format using output plugins that can be used to log data in custom formats or to send data to external systems.

Snort supports various add-ons and plugins to extend its functionality, improve performance, and integrate with other systems. These extensions can be divided into two main categories: preprocessors and output plugins.

Preprocessors are modules that process and analyze packets before they are

passed to the detection engine. They can be used to decode protocols, normalize traffic, or perform other specialized tasks.

They are Frag3 (for fragmented IP packets); Stream5 (stateful TCP and UDP traffic analysis); HTTP inspection; SMTP and DCE/RPC traffic detection and analysis; etc.

Output plugins control how Snort logs data and alerts. They can be used to customize the output format, store data in external systems, or forward data to other tools.

Alert modes available are seven and they can be specified when executing Snort:

- **Fast:** when in fast mode, Snort alerts report the timestamp, send an alert message, show the source IP address and port, and the destination IP address and port. This mode is instructed using the `-A fast` flag.

- **Full:** additionally to the information printed in the fast mode, the full mode shows the TTL, packet headers and datagram length, service, ICMP type, window size, ACK and sequence number. The full mode is defined with the `-A full` flag, but this is the default alerts mode.

- **Console:** prints fast alerts in the console. This mode is implemented with the `-A console` flag.

- **Cmg:** this alerts mode was developed by Snort for testing purposes; it prints a full alert on the console without saving logs. The mode is implemented with the `-A cmg` flag.

- **Unsock:** this is useful to export alert reports to other programs through Unix sockets. The unsock mode is implemented using the `-A unsock` flag.

- **Syslog:** in syslog (System Logging Protocol) mode, Snort sends alert logs remotely; this mode is implemented by adding the `-s` flag.

- **None:** with this mode, Snort does not generate alerts.

A great value is given by the **Vulnerability Research Team (VRT)**.

Sourcefire, the company that originally developed Snort, created the team to enhance Snort's effectiveness by researching and developing new detection rules and signatures. The team was comprised of security experts who focused on discovering and analyzing vulnerabilities, exploits, and attack techniques to create high-quality detection rules for Snort.

In 2013, Cisco acquired Sourcefire, and the Vulnerability Research Team eventually became part of Cisco Talos, a larger threat intelligence organization within Cisco. Cisco Talos continues to maintain and develop Snort rulesets and provides two main types of rulesets for Snort:

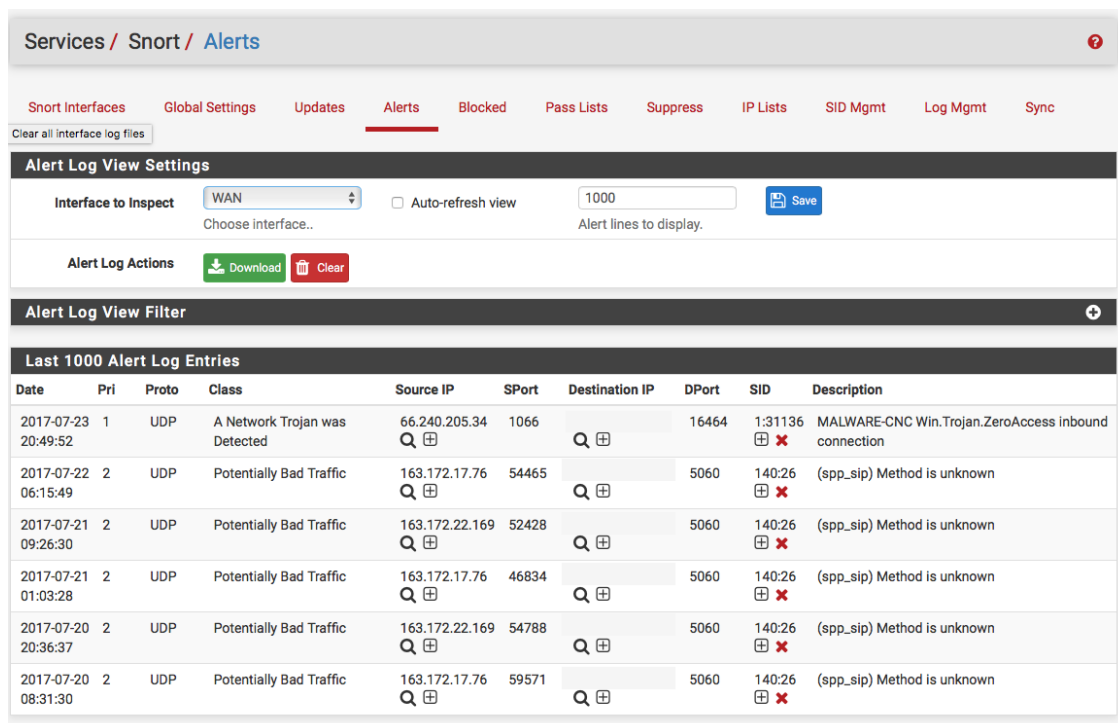
- **Talos (formerly VRT) ruleset** is a subscription-based, commercial ruleset that provides the most up-to-date and comprehensive detection rules for Snort.

Subscribers benefit from timely updates, professional support, and access to rules that protect against the latest threats and vulnerabilities. The Talos ruleset is recommended for organizations that require the highest level of security and protection.

- **Snort Community ruleset:** is a free, open-source ruleset maintained by the Snort community, Cisco Talos, and other contributors. While the community ruleset is not as comprehensive or up-to-date as the Talos ruleset, it still provides a valuable source of detection rules for Snort users. The Snort community ruleset is a good starting point for organizations and individuals who want to use Snort without subscribing to the commercial ruleset.

Software Requirements

Operating System: Linux, Windows, and macOS, but it is also deployed on Linux distributions such as Ubuntu, CentOS, or Security Onion.



The screenshot shows the Snort dashboard interface. At the top, there are navigation tabs: Services / Snort / Alerts. Below this, there are several menu items: Snort Interfaces, Global Settings, Updates, Alerts (selected), Blocked, Pass Lists, Suppress, IP Lists, SID Mgmt, Log Mgmt, and Sync. A 'Clear all interface log files' button is visible. The main section is titled 'Alert Log View Settings' and includes a dropdown for 'Interface to Inspect' (set to WAN), an 'Auto-refresh view' checkbox, and a text input for 'Alert lines to display' (set to 1000). Below this are 'Alert Log Actions' with 'Download' and 'Clear' buttons. The 'Alert Log View Filter' section is also visible. The main content area is titled 'Last 1000 Alert Log Entries' and contains a table with the following data:

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34	1066		16464	1:31136	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76	54465		5060	140:26	(spp_sip) Method is unknown
2017-07-21 09:26:30	2	UDP	Potentially Bad Traffic	163.172.22.169	52428		5060	140:26	(spp_sip) Method is unknown
2017-07-21 01:03:28	2	UDP	Potentially Bad Traffic	163.172.17.76	46834		5060	140:26	(spp_sip) Method is unknown
2017-07-20 20:36:37	2	UDP	Potentially Bad Traffic	163.172.22.169	54788		5060	140:26	(spp_sip) Method is unknown
2017-07-20 08:31:30	2	UDP	Potentially Bad Traffic	163.172.17.76	59571		5060	140:26	(spp_sip) Method is unknown

Figure 11.2. Snort dashboard

Suricata is an open-source product developed and maintained by the Open Information Security Foundation (OISF).



Figure 11.3. Suricata logo

Suricata inspects network traffic in real-time and uses a combination of signature-based detection, protocol analysis, and flow-aware heuristics to identify potential threats. It can be deployed on various platforms, including Linux, FreeBSD, and MacOS, and is compatible with both IPv4 and IPv6 networks.

Its key features are:

- **Signature-based Detection:** it supports the Snort VRT, Emerging Threats, and Emerging Threats Pro rule sets, which contain a large number of signatures for detecting known threats and attacks.

- **Protocol Analysis:** it can decode and interpret various network protocols, such as HTTP,

TLS/SSL, DNS, and SMB, allowing it to detect anomalies, policy violations, and malicious activity within these protocols.

Flow-aware Heuristics: by analyzing network flows and tracking the state of network connections, Suricata can detect and prevent certain types of attacks, such as

port scans, brute force attempts, and Distributed Denial of Service (DDoS) attacks.

Inline Intrusion Prevention: Suricata can operate in inline mode, allowing it to block malicious traffic in real-time by integrating with network devices, such as routers, switches, and firewalls.

File Extraction: Suricata can extract files from network traffic for further analysis, helping to identify and block the transfer of malware or other malicious content.

Scalability and Performance: Suricata is designed to take advantage of multi-core and multi-threaded processors, ensuring high performance even in large-scale and high-traffic networks. Some of the specific features of Suricata IDPS against cyberattacks include:

Signature-based Detection: Suricata supports multiple rule sets, such as Snort VRT, Emerging Threats, and Emerging Threats Pro, which contain thousands of signatures for detecting known threats and attack patterns. This helps identify and block known exploits, malware, and other cyberattacks.

Protocol Analysis: Suricata can decode and interpret numerous network protocols, including HTTP, TLS/SSL, DNS, and SMB. By analyzing these protocols, Suricata can detect anomalies, policy violations, and malicious activities often associated with cyberattacks.

Flow-aware Heuristics: Suricata analyzes network flows and tracks the state of network connections, enabling it to detect and prevent certain types of attacks, such as port scans, brute force attempts, and Distributed Denial of Service (DDoS) attacks.

Inline Intrusion Prevention: in inline mode, Suricata can block malicious traffic in real-time by integrating with network devices like routers, switches, and firewalls. This helps prevent threats from reaching their targets and causing damage.

File Extraction and Analysis: Suricata can extract files from network traffic for further analysis. This feature helps identify and block the transfer of malware or other malicious content while also aiding in forensic investigations.

Encrypted Traffic Analysis: Suricata can analyze encrypted traffic, such as TLS/SSL, and detect malicious activities hidden within encrypted communications. This is particularly useful for identifying threats that attempt to evade detection by encrypting their network traffic.

Customizable Rules: Suricata allows security professionals to create custom rules tailored to their environment, helping them detect and prevent threats specifically targeting their organization or industry.

Scalability and Performance: Suricata has been designed to take advantage of multi-core and multi-threaded processors, ensuring high performance even in large-scale and high-traffic networks. This helps maintain effective detection and prevention capabilities even as the volume of network traffic increases.

Suricata can be installed on various operating systems, including Linux, FreeBSD, and macOS. Linux is the most common platform for Suricata deployments due to its flexibility, performance, and wide range of compatible hardware.

As a general guideline, a minimum of 4 GB of RAM is recommended for small to medium networks. For larger deployments or more extensive rulesets, it is better using 8 GB or more.



Figure 11.4. Suricata dashboard

Zeek, formerly known as Bro, is an open-source IDPS distributed by International Computer Science Institute (ICSI) and the National Center for Supercomputing Applications (NCSA).



Figure 11.5. Zeek logo

It is designed to provide deep, context-rich insights into network traffic, enabling security professionals to detect and investigate a wide range of cybersecurity threats and incidents.

This IDPS is known for its powerful network analysis capabilities and flexible scripting language, which allows users to create custom scripts tailored to their specific environment and security requirements. It operates primarily as a passive network monitoring tool that extracts high-level metadata from network traffic, generating detailed logs and facilitating real-time analysis.

Some of the key features of Zeek IDPS include:

- **Protocol Analysis:** Zeek supports a wide range of network protocols, including HTTP, FTP, DNS, SSL/TLS, and many others. It can dissect and analyze these protocols, providing valuable information about network activity and potential security issues.
- **Extensible Scripting Language:** Zeek uses its own scripting language, which allows for extensive customization and flexibility. Users can create custom scripts to define their own rules, analyze specific traffic patterns, or perform other advanced network analysis tasks.
- **Logging and Reporting:** Zeek generates detailed logs of network activity, including connection logs, application layer logs, and custom logs defined by user.

scripts. These logs provide valuable data for threat hunting, incident response, and forensic analysis.

- **File Analysis:** Zeek can extract files from network traffic and analyze them for potential security threats, such as malware or other malicious content.

- **Real-time Analysis:** Zeek can process and analyze network traffic in real-time, providing security professionals with continuous visibility into network activity and potential threats.

- **Integration with other security tools:** Zeek can be easily integrated with other security tools, such as Security Information and Event Management (SIEM) systems or threat intelligence platforms, to enhance overall network security and threat detection capabilities.

For this product, there are no particular hardware requirements requested.

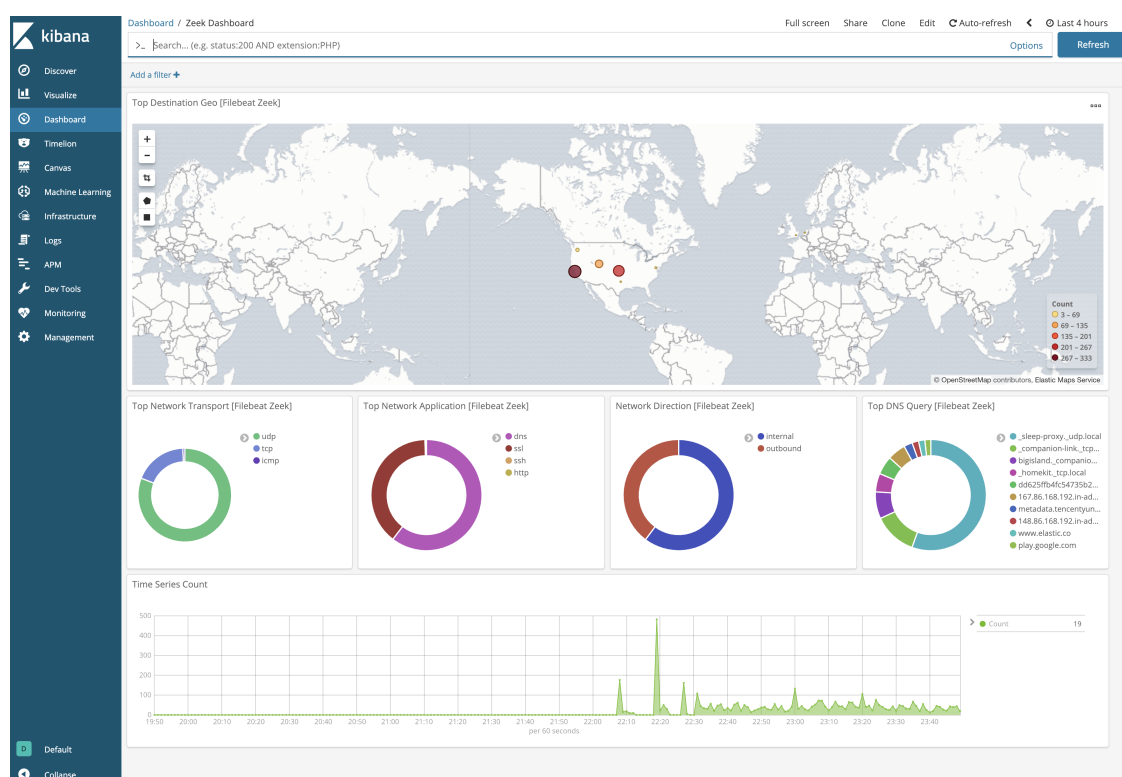


Figure 11.6. Zeek dashboard

Security Onion is a free and open-source Linux distribution tailored for network security monitoring, intrusion detection, and log management.

Created by Doug Burks and maintained by a community of security professionals, Security Onion provides a comprehensive platform to help organizations detect, investigate, and respond to various cybersecurity threats and incidents.

Security Onion integrates multiple security tools, including network intrusion detection system (NIDS), host intrusion detection system (HIDS), and log management solutions, into a cohesive and easy-to-use platform.

Security Onion simplifies the process of deploying, configuring, and managing these tools by providing a user-friendly interface and pre-configured settings. It

also includes various features for monitoring system performance, managing rule-sets, and customizing the platform to suit the unique needs of each organization.



Figure 11.7. Security Onion logo

In this suite are offered different features:

- **Intrusion Detection and Prevention:** Security Onion integrates Suricata and Snort, two powerful network intrusion detection and prevention systems (IDPS), which monitor network traffic and detect malicious activities based on predefined rules and heuristics.
- **Network Security Monitoring:** Zeek is included in Security Onion for deep network traffic analysis and generating detailed logs. These logs provide valuable information about network activity, enabling security teams to identify potential threats, anomalies, and policy violations.
- **Host Intrusion Detection:** Wazuh, a host intrusion detection system (HIDS), is integrated into Security Onion, monitoring system and application logs as well as file integrity on endpoints. This helps detect potential security threats and policy violations within the organization's IT infrastructure.
- **Log Management and Analysis:** The Elastic Stack (Elasticsearch, Logstash, and Kibana) is incorporated in Security Onion, providing powerful log management, processing, and visualization capabilities. This allows security teams to search, analyze, and visualize log data from various sources, enabling them to identify patterns and trends that may indicate security incidents or threats.
- **File Analysis:** Strelka, a file scanning framework, is included in Security Onion. It analyzes files extracted from network traffic, helping to identify potential security threats, such as malware or other malicious content.
- **Incident Response and Management:** TheHive and Cortex, integrated into Security Onion, provide a security incident response platform and an analysis engine. These tools enable security teams to manage, investigate, and respond to incidents efficiently and effectively.
- **Threat Intelligence:** Security Onion can be configured to incorporate threat intelligence feeds, enabling security teams to stay informed about the latest threats and vulnerabilities, and further enhancing detection and prevention capabilities.
- **Alert Management:** Security Onion includes alert management features to help security teams prioritize, investigate, and respond to security alerts generated by various tools within the platform.
- **Customization and Extensibility:** Security Onion allows organizations to customize its configuration and rulesets to better match their unique environment and security requirements. It also supports the development and integration of custom scripts, plugins, and additional tools, making it highly adaptable to specific needs.

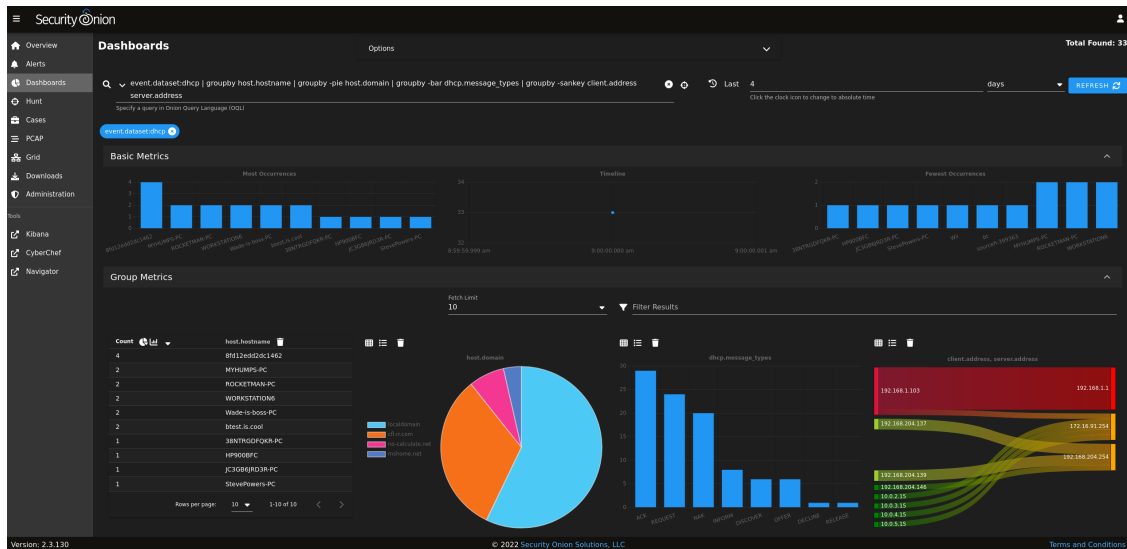


Figure 11.8. Security Onion dashboard

OSSEC (Open Source Security) is an open-source host-based intrusion detection system (HIDS) that performs log analysis, file integrity checking, policy monitoring, rootkit detection, and real-time alerting and active response.



Figure 11.9. OSSEC logo

It is a comprehensive security tool that helps organizations protect their critical infrastructure and sensitive data by monitoring and analyzing activity on their systems.

OSSEC is platform-independent, which means it can be used on various operating systems such as Linux, Windows, macOS, and Unix-based systems. The OSSEC project aims to provide a high-level security solution with a flexible architecture that can be easily customized and extended to fit the specific needs of an organization.

OSSEC has a client-server model, where agents are installed on the monitored systems and send information to a central server, which is responsible for analyzing the data and triggering alerts or response actions when necessary. The server can also be configured to operate in a standalone mode without agents, suitable for smaller environments or single-system monitoring.

Key feature of OSSEC can be listed as following:

- **Log analysis:** OSSEC can analyze logs from various sources, such as syslog, Event Viewer, Apache, MySQL, and many others. It can detect anomalies and suspicious activities, helping administrators identify potential security incidents.

- **File integrity monitoring:** OSSEC can monitor files and directories for changes, such as content modifications, permission alterations, or ownership changes. This helps ensure the integrity of critical system and application files.

- **Rootkit detection:** OSSEC can detect the presence of rootkits, which are malicious tools that try to hide their existence and activities on a compromised system.

- **Policy monitoring:** OSSEC can check the compliance of systems with predefined security policies, ensuring that configurations are in line with best practices and organizational requirements.

- **Real-time alerting and active response:** OSSEC can send real-time alerts when specific events or anomalies are detected. It can also execute automated responses, such as blocking an IP address or disabling an account, to mitigate potential threats.

OSSEC supports Linux, Windows, macOS, and various Unix-based operating systems.

The memory requirements depend on the size of the monitored environment and the complexity of the rules and decoders. For small environments with a few agents, 512 MB to 1 GB of RAM should be sufficient. For larger environments with many agents and extensive rule sets, 2 GB of RAM or more may be necessary.

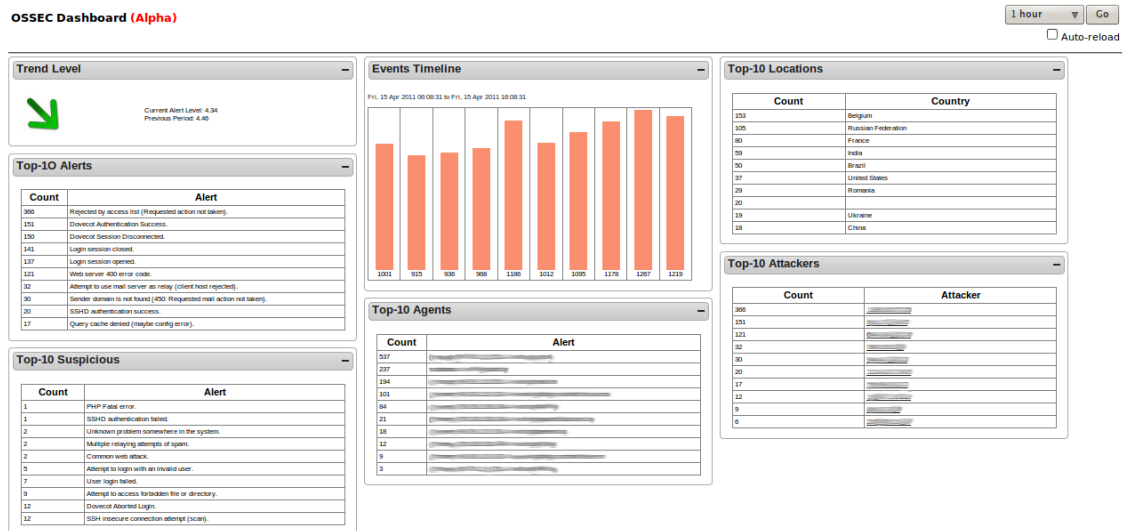


Figure 11.10. OSSEC example dashboard

Cisco FirePower is an integrated intrusion detection and prevention system (IDPS) developed by Cisco Systems, a leading global provider of networking and cybersecurity solutions.



Figure 11.11. Cisco FirePower logo

The Cisco FirePower IDPS is designed to protect networks and systems from a wide range of threats, including malware, exploits, and other malicious activities, by detecting and preventing them in real-time.

Cisco FirePower is part of Cisco's broader security portfolio and can be deployed as a standalone solution or integrated with other Cisco security products, such as Cisco ASA (Adaptive Security Appliance) firewalls and Cisco FirePower Threat Defense (FTD) systems.

This IDPS include:

- **Advanced Threat Detection:** Cisco FirePower uses a combination of signature-based and behavior-based detection techniques to identify known and unknown threats. It leverages Cisco Talos, the company's threat intelligence group, to stay up-to-date with the latest threat information and provide effective protection against emerging threats.
 - **Real-time Prevention:** it can actively block or mitigate threats in real-time, preventing them from causing harm to own network and systems. This includes blocking network connections, dropping malicious packets, or quarantining infected systems.
 - **Context-aware Analysis:** it can analyze network traffic and system activities in the context of the broader environment, providing more accurate and actionable insights. This includes understanding the relationships between users, devices, applications, and other network elements, which helps to reduce false positives and improve threat detection.
 - **File Trajectory and File Reputation:** Cisco FirePower can track the movement of files across the network and determine their reputation based on various factors, such as source, destination, and associated behaviors. This helps organizations identify potential threats, such as malware infections or data exfiltration attempts.
 - **SSL/TLS Decryption:** it can decrypt SSL/TLS encrypted traffic, enabling it to inspect and analyze the content of encrypted communications for potential threats. This helps organizations detect and prevent cyber attacks that leverage encryption to evade detection.
 - **Application Visibility and Control:** it can identify and control thousands of applications, providing organizations with granular visibility and control over their network traffic. This helps prevent unauthorized applications from running on the network, which can reduce the attack surface and protect against application-based attacks.
 - **URL Filtering:** Cisco FirePower can filter and control access to websites based on categories, user groups, or custom rules. This helps organizations prevent users from accessing malicious or compromised websites, which can reduce the risk of phishing attacks or drive-by malware infections.
 - **Security Intelligence:** Cisco FirePower can leverage external threat intelligence feeds, including Cisco Talos, to identify and block known malicious IP addresses, domains, and URLs. This helps organizations protect against known threats and stay ahead of emerging cyber attacks.
 - **Integration with Cisco Security Portfolio:** it can be integrated with other Cisco security products, such as Cisco ASA firewalls and Cisco FirePower Threat Defense (FTD) systems, to provide a comprehensive and unified security solution. This integration enables organizations to leverage the full potential of Cisco's security capabilities and protect their networks and systems more effectively.
- Other than physical appliance, Cisco FirePower also offers virtual appliances, which can be deployed on supported hypervisor platforms, such as VMware ESXi, Microsoft Hyper-V, and KVM. The system requirements for virtual appliances depend on the desired performance and capacity, as well as the host system's resources. Usually, CPU: 4 vCPU (minimum), 8 vCPU (recommended) and Memory: 8 GB RAM (minimum), 16 GB RAM (recommended), Storage: 100 GB.

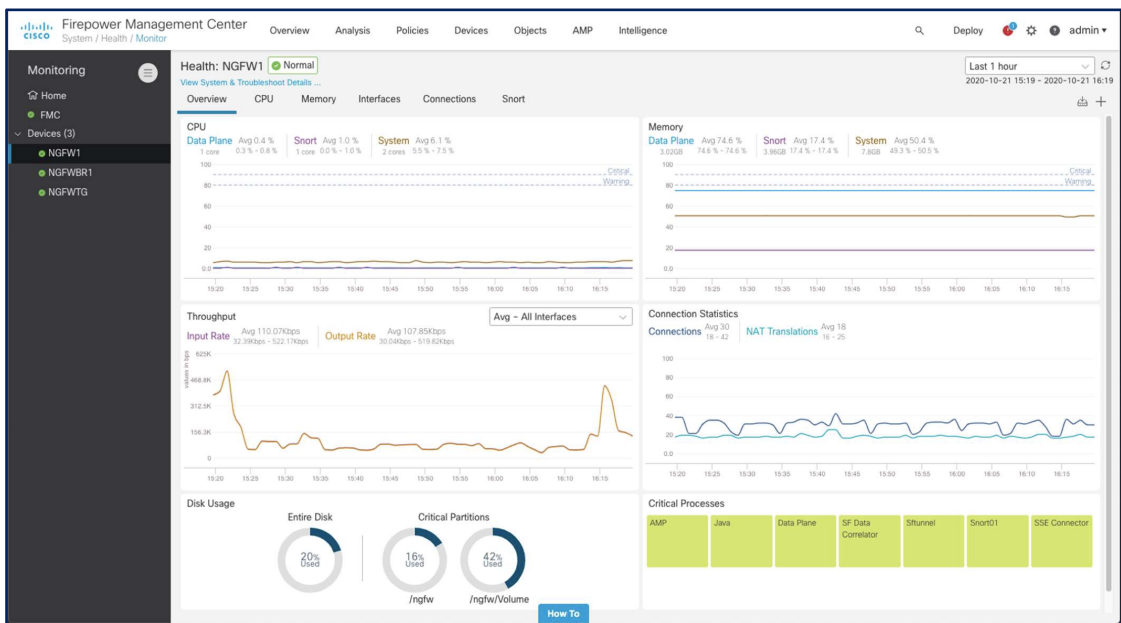


Figure 11.12. Cisco FirePower dashboard

Fortinet FortiGate FortiGuard offers an IPS solution with a variety of features to help organizations defend against various cyber attacks. These features focus on different aspects of network and system security, such as threat detection, prevention, analysis, and management.

Some specific features are:

- **Signature-based Detection:** FortiGuard IPS uses a comprehensive database of signatures to identify known threats, such as exploits, malware, and other malicious activities. This database is regularly updated by Fortinet’s FortiGuard Labs to provide effective protection against emerging threats.

- **Behavior-based Detection:** in addition to signature-based detection, FortiGuard IPS employs behavior-based analysis to detect unknown or zero-day threats by analyzing network traffic for suspicious patterns or activities.

Figure 11.13. FortiGuard logo



- **Real-time Prevention:** it can actively block or mitigate threats in real-time, preventing them from causing harm to the network and systems. This includes blocking network connections, dropping malicious packets, or quarantining infected systems.

- **Context-aware Analysis:** it can analyze network traffic and system activities in the context of the broader environment, providing more accurate and actionable insights. This includes understanding the relationships between users, devices, applications, and other network elements, which helps to reduce false positives and improve threat detection.

- **Application Control:** FortiGuard IPS can identify and control thousands of applications, allowing organizations to gain granular visibility and control over their network traffic. This helps prevent unauthorized applications from running on the network, reducing the attack surface and protecting against application-based attacks.
- **SSL/TLS Inspection:** it can decrypt SSL/TLS encrypted traffic, enabling it to inspect and analyze the content of encrypted communications for potential threats. This helps organizations detect and prevent cyber attacks that leverage encryption to evade detection.
- **Threat Intelligence:** it leverages threat intelligence from FortiGuard Labs, including information on known malicious IP addresses, domains, and URLs. This helps organizations protect against known threats and stay ahead of emerging cyber attacks.
- **Integrated Management:** FortiGuard IPS can be managed through a centralized management console, such as FortiManager, which provides a unified view of the network security and enables to configure, monitor, and analyze the IDPS deployment.
- **Integration with Fortinet Security Fabric:** FortiGuard IPS can be integrated with other Fortinet security products, such as FortiAnalyzer, FortiSandbox, and FortiWeb, to provide a comprehensive and unified security solution. This integration enables organizations to leverage the full potential of Fortinet's security capabilities and protect their networks and systems more effectively.

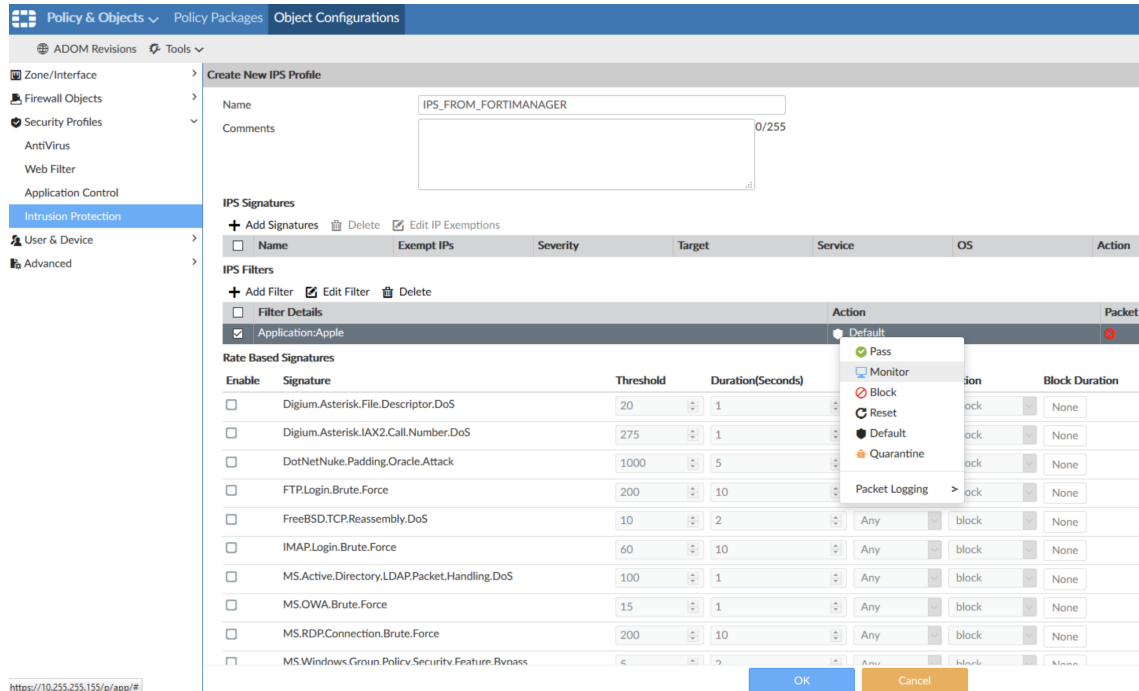


Figure 11.14. FortiGate dashboard

EasyIDS is an open-source project that aims to simplify the deployment and management of Snort, an intrusion detection system (IDS), by providing a pre-configured Linux distribution and web-based interface for managing Snort and its

related



Figure 11.15. EasyIDS logo

Snort alerts. The primary goal of EasyIDS is to make it easier for users, especially those with limited experience in Linux and network security, to set up and manage an IDS quickly.

With EasyIDS, users can:

- easily deploy Snort on a dedicated hardware or virtual machine;
- configure Snort using a web-based interface;
- manage Snort rules and rule sets;
- monitor and analyze Snort alerts using the BASE interface;
- generate reports and visualizations of network activity.

PMGraph (Passive Measurement Graphs), even if not a direct component of EasyIDS, it can be used in conjunction with EasyIDS or other intrusion detection systems to provide better insights into the network traffic patterns and potential security incidents.

PMGraph generates graphs based on data from various sources, such as NetFlow, sFlow, IPFIX, or pcap files. By analyzing and visualizing the data, PMGraph can help network administrators and security professionals identify trends, detect potential anomalies, and investigate security incidents more efficiently.

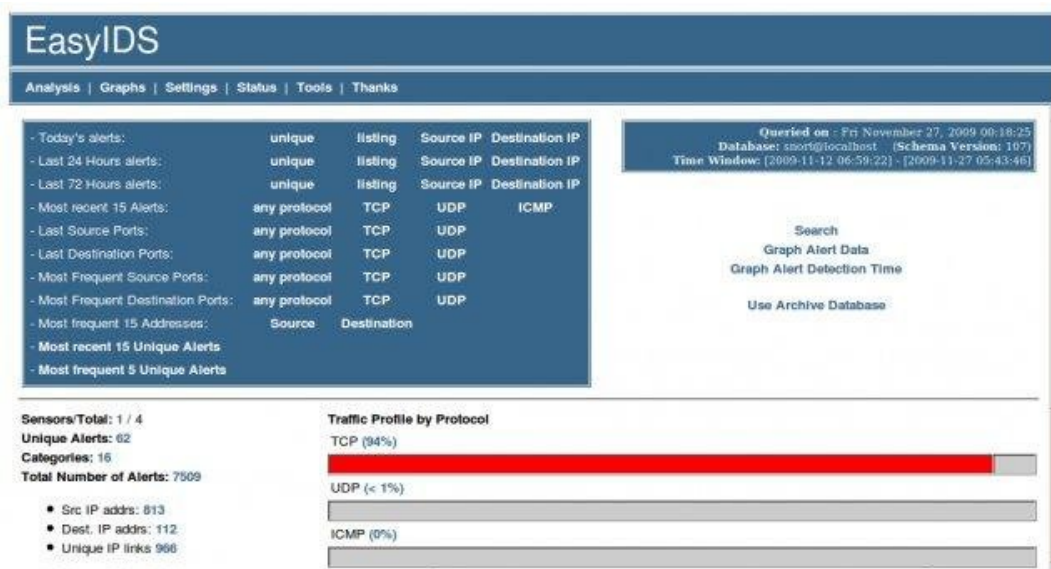


Figure 11.16. EasyIDS example dashboard

Untangle is an Intrusion Detection and Prevention System (IDPS) component within Untangle's NG Firewall, a Unified Threat Management (UTM) solution

designed for small and medium-sized businesses, educational institutions, and non-profit organizations.



Figure 11.17. Untangle logo

Untangle NG Firewall is a software-based solution that provides a range of integrated security applications, including firewall, VPN, content filtering, spam blocking, and IDPS. Untangle IDPS aims to detect and prevent network intrusions by monitoring network traffic for suspicious activity and known attack patterns. It utilizes the open-source Snort IDS/IPS engine, which is widely used and well-regarded in the industry. Snort is frequently updated with new rules and signatures to detect and

prevent the latest threats, ensuring that Untangle IDPS remains effective against emerging security risks.

Key features of Untangle IDPS include:

- **Signature-based Detection:** Untangle IDPS uses Snort's extensive library of signatures to detect known attack patterns in network traffic.
 - **Automatic Updates:** the system regularly updates the rules and signatures to maintain effective protection against new threats.
 - **Custom Rules:** administrators can create custom rules to detect and respond to specific threats or patterns relevant to their environment.
 - **Alerts and Notifications:** Untangle IDPS can generate alerts and notifications for detected intrusions, allowing security teams to respond quickly to potential incidents.
 - **Reporting:** the solution provides detailed reporting on detected intrusions, enabling security teams to analyze incidents and improve their defense strategies.
- Untangle NG Firewall provides a comprehensive suite of integrated security applications designed to protect organizations against various types of cyberattacks. Some of the key features and applications included in Untangle NG Firewall are:
- **Firewall:** Untangle's stateful firewall offers network segmentation and access control, filtering traffic based on IP addresses, ports, and protocols, to protect the internal network from unauthorized access.
 - **Intrusion Detection and Prevention System (IDPS):** as discussed earlier, Untangle's IDPS monitors network traffic for suspicious activity and known attack patterns, helping to detect and prevent intrusions.
 - **Virtual Private Network (VPN):** Untangle provides both site-to-site and remote-access VPN capabilities, enabling secure connections for remote users and offices.
 - **Web Filter:** the web filter application allows administrators to enforce acceptable use policies, block access to malicious websites, and filter content based on categories, helping to protect users from phishing attacks, malware, and inappropriate content.
 - **Spam Blocker:** Untangle's spam blocker application filters incoming email to reduce the volume of unsolicited messages and protect users from email-borne threats such as phishing and malware.
 - **Virus Blocker:** the virus blocker application scans files and email attachments for known malware signatures, helping to detect and block the spread of viruses, worms, and other forms of malware.

- **Phish Blocker:** this application specifically targets phishing attacks by detecting and blocking access to known phishing websites and email messages designed to steal sensitive information.
- **Application Control:** Untangle's application control feature allows administrators to identify and manage the use of specific applications on the network, helping to prevent the abuse of potentially risky or bandwidth-consuming applications.
- **Bandwidth Control:** this feature enables the management and prioritization of network traffic to ensure critical applications have the necessary resources while limiting non-essential or high-bandwidth usage.
- **SSL Inspector:** Untangle's SSL Inspector decrypts and inspects encrypted HTTPS traffic to detect and block hidden threats, such as malware or data exfiltration, concealed within encrypted communications.
- **Advanced Threat Protection (ATP):** the ATP application (available in the premium package) provides an additional layer of protection by detecting and blocking zero-day threats, ransomware, and other advanced cyberattacks using cloud-based sandboxing and threat intelligence.

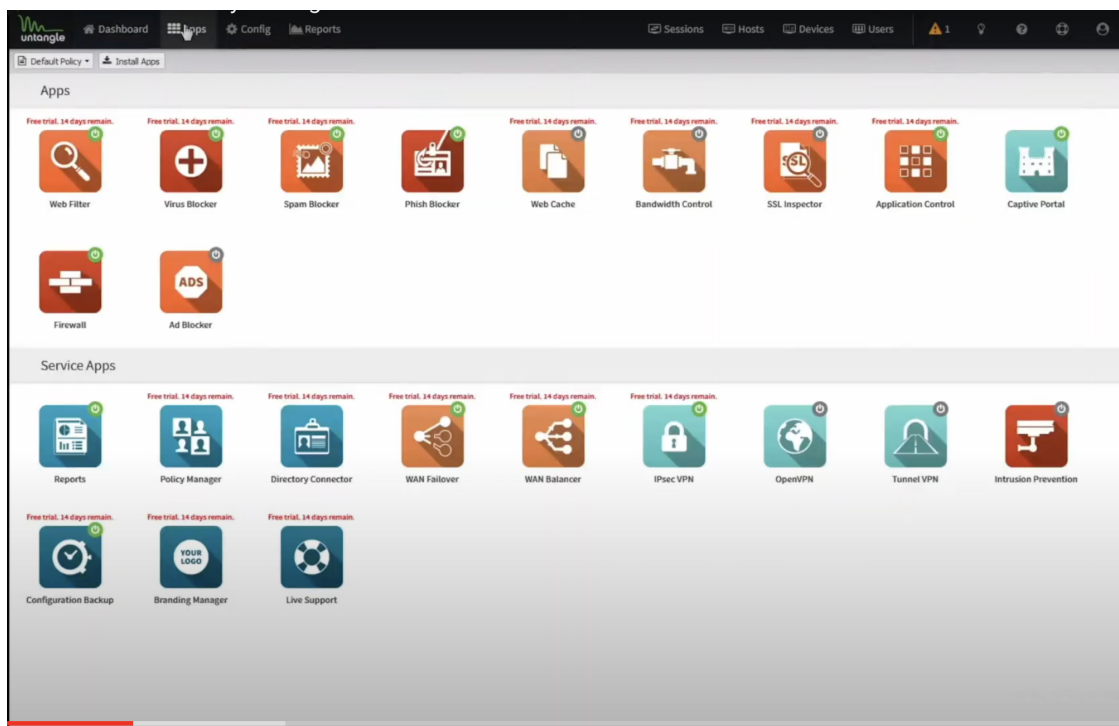


Figure 11.18. Untangle dashboard

AIDE is an open-source software that helps detect unauthorized changes to crucial system files by monitoring file integrity. It is commonly used on Unix-based systems such as Linux, FreeBSD, and macOS.

AIDE works by creating a database of file attributes, including file permissions, ownership, modification time, and other properties. This database is then used as a baseline to compare against future scans of the system, allowing AIDE to detect any unauthorized changes to the monitored files.



Figure 11.19. AIDE logo

Some key features of AIDE include:

- **Flexible Configuration:** it allows to define which files and directories to monitor and what attributes to check for each file, giving a granular control over own intrusion detection strategy. Checksum Algorithms: AIDE supports various checksum algorithms, such as MD5, SHA1, SHA256, and others, to create a unique fingerprint for each monitored file. This helps ensure that any unauthorized changes to the files are detected.

- **Compression and Encryption:** it can compress and encrypt its database to save storage space and protect the database from unauthorized access.

- **Reporting:** AIDE provides detailed reports of any detected changes, including information about the affected files, the type of change, and the date and time when the change was detected.

- **Integration with System Loggers:** AIDE can be configured to send alerts to syslog or other system loggers when it detects unauthorized changes, allowing the integration of AIDE with an existing security information and event management (SIEM) systems.

Samhain, as a host-based intrusion detection system (HIDS), offers several features to protect individual systems against cyber attacks by monitoring file integrity, log files, and detecting unauthorized access or changes.



Figure 11.20. Samhain logo

These features can be valuable for identifying security breaches, such as malware infections, unauthorized access, or configuration tampering. Here are some specific features of Samhain against cyber attacks:

- **File Integrity Monitoring:** Samhain monitors specified files and directories for unauthorized changes, including modifications, deletions, or creation of new files. This helps uncover potential security breaches, such as malware modifying system files or an attacker tampering with configuration files.

- **Log File Monitoring:** it can monitor system log files for signs of intrusion attempts or other malicious activities. This real-time monitoring provides an additional layer of security by detecting and alerting to potential security incidents as they occur.

- **Centralized Management:** it supports centralized management, allowing to control and receive alerts from multiple Samhain clients across the network. This feature enables to maintain a consistent security posture across the infrastructure and respond more effectively to cyber attacks.

- **Stealth Mode:** it can operate in stealth mode, making it difficult for attackers to detect its presence on the system. This feature can help protect the intrusion detection system from being bypassed or disabled by an attacker.

- **Tamper Resistance:** Samhain includes measures to prevent tampering with its configuration files, log files, and database. This ensures the integrity of the data and helps protect against attempts to bypass detection or manipulate the system.

- **Alerting and Reporting:** it provides detailed alerts and reports when it detects potential security incidents, allowing to take appropriate action to remediate the issue. This information can help to identify the nature of the attack and respond accordingly.

Samhain is compatible with Unix-based systems, including Linux, FreeBSD, and macOS.

Samhain is not resource-intensive, and the memory requirements depend on the size of file system and the number of files monitored.

Chapter 12

Firewall and IDS/IPS

A firewall is a network security device that monitors and controls incoming and outgoing network traffic based on predetermined security rules. Its primary purpose is to establish a barrier between trusted internal networks (such as corporate or home networks) and untrusted external networks (such as the Internet), thereby preventing unauthorized access and protecting the internal network from potential threats.

Firewalls can be implemented as hardware, software, or a combination of both. They work by filtering traffic according to specific criteria, such as IP addresses, ports, protocols, or application-level content. When network traffic meets the defined security rules, it is allowed to pass through the firewall. Conversely, if the traffic does not meet the criteria, it is blocked or denied.

A firewall and an Intrusion Detection and Prevention System (IDPS) are two distinct security technologies that can be used together to provide a more comprehensive and layered approach to network security. They focus on different aspects of network protection.

FWs use rules to allow or deny traffic based on criteria like IP addresses, ports, or protocols, while Intrusion Detection and Prevention System (IDPS) combines the capabilities of an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) to provide both detection and prevention functions in a single system. The logical connection between a firewall and an IDPS is in their complementary roles in protecting the network.

While a firewall provides the first line of defense by controlling traffic based on predefined rules, an IDPS offers more in-depth analysis and real-time threat detection and prevention.

In a typical network security setup, a perimeter firewall is placed at the company network edges, filtering traffic between the internal network and external networks such as the Internet.

By working together, a firewall and an IDPS provide a more robust security posture.

The firewall prevents unauthorized access and filters traffic based on specific rules, reducing the attack surface. The IDPS detects and prevents threats and intrusions that may have bypassed the firewall or originated from within the internal network. If an IDPS cannot stop a threat coming from the network such as Network Service Worm or Denial of Service, it can instead reconfigure the Firewall to be able to

block them.

In addition, the IDPS can directly analyze firewall logs to be able to get a better view of what is happening in the external network or take information about impending threats.

These are principles from which Security-in-Depth starts and it is based on: a strategy referred to a cybersecurity approach that uses multiple layers of security for holistic protection.

A layered defense helps security organizations in reducing vulnerabilities, containing threats, and mitigating cyber risks. With a Defense-In-Depth approach, if one layer of defense fails or a bad actor breaches into it, an issue might be contained by the next layer or next layers of defense.

Chapter 13

VEREFOO

VEREFOO stands for Verified Refinement and Optimized Orchestration and it is a concrete solution for automatically managing security configuration on firewalls. It is a perfect solution especially for large networks that are hardly to configure, considering misconfigurations probability, so vulnerabilities, would increase with network complexity.

The aim of VEREFOO is to automatically orchestrates and configures network security functions in virtualized networks. This is achieved through the definition of a proper allocation scheme and then computing the optimal configuration, consecutively.

It is a project presented in the 2019 at the 4th IEEE International Conference on Computing, Communications and Security in Rome (Italy)¹.

It follows a policy-based security automation, starting from defined security policies and a network topology that assures an automatic definition of the optimal allocation scheme for Network Security Functions NFSs, in a given network, and their optimal configuration.

Currently, VEREFOO works for packet filter firewalls but it is ready for other security functions as well.

The first inputs are an allocation graph where it is described the network topology (endpoints, load balancers, etc.) and the allocation places where VEREFOO will decide to place a firewall². [fig. 13.1-13.2]

The other input is a Network Security Requirement where are listed records categorized by action (allow/deny), source and destination IP, source and destination port, and transport level protocols used. [fig. 13.3]

Those help to define topology and requirements to send over VEREFOO.

¹D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Towards a fully automated and optimized network security functions orchestration", 4th International Conference on Computing, Communications and Security (ICCCS), pp. 1-7, 2019

²D. Bringhenti, G. Marchetto, R. Sisto and F. Valenza, "A novel approach for security function graph configuration and deployment", IEEE 7th International Conference on Network Softwarization (NetSoft), pp. 457-463, 2021

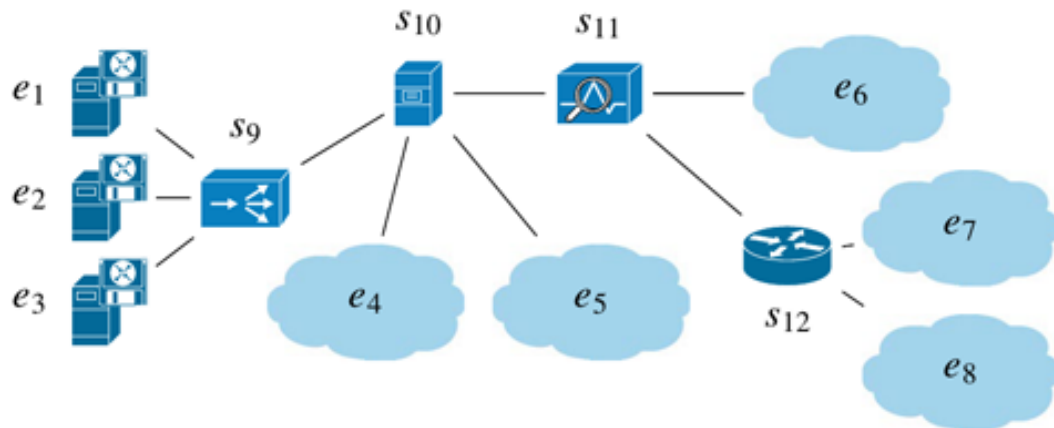


Figure 13.1. Allocation Graph input

Identifier	IP address	Function type / role
e_1	130.10.0.1	HTTP web server
e_2	130.10.0.2	HTTP web server
e_3	130.10.0.3	HTTP web server
e_4	40.40.41.*	IT office of Company A
e_5	40.40.42.*	Business office of Company A
e_6	88.80.84.*	Company B
e_7	192.168.1.*	IT office of Company C
e_8	192.168.2.*	Business office of Company C
s_9	130.10.0.4	Load balancer
s_{10}	33.33.33.2	Web cache
s_{11}	33.33.33.3	Traffic monitor
s_{12}	220.124.30.1	NAT

Figure 13.2. Service Graph functions input

The two outputs are a Firewall Allocation Scheme with firewalls assigned to every allocation place, and an optimized configuration for each one, setting also a small set of filtering rules. VEREFOO will decide the best number of firewalls and their

Action	IPSrc	IPDst	pSrc	pDst	tProto
Allow	192.168.1.*	192.168.2.*	*	*	*
Allow	192.168.2.*	192.168.1.*	*	*	*
Allow	192.168.1.*	130.10.0.*	*	80	TCP
Deny	192.168.1.*	130.10.0.*	*	≠80	TCP
Deny	192.168.1.*	130.10.0.*	*	*	UDP
Deny	192.168.2.*	130.10.0.*	*	*	*
Allow	130.10.0.*	192.168.1.*	*	*	*
Allow	40.40.41.*	130.10.0.*	*	80	TCP
Deny	40.40.41.*	130.10.0.*	*	≠80	TCP
Deny	40.40.41.*	130.10.0.*	*	*	UDP
Deny	40.40.42.*	130.10.0.*	*	*	*
Allow	130.10.0.*	40.40.41.*	*	*	*
Allow	40.40.42.*	40.40.41.*	*	*	*
Deny	40.40.41.*	40.40.42.*	*	*	*
Allow	88.80.84.*	40.40.*.*	*	*	*
Deny	88.80.84.*	130.10.0.*	*	*	*

Figure 13.3. Network Security Requirements input

position within the network, defining also appropriate security policies³.

So, on one hand, it establishes the optimal Firewall Allocation Scheme, composed of the minimum number of firewall instances to be placed in the input Allocation Places. On the other hand, for each allocated instance, it computes the optimal configuration, composed of a default action and the smallest set of filtering rules⁴. [fig. 13.4-13.5] Allocation graph is written in a XML file, where is defined how network nodes are interconnected and their configuration, but also the file is filled with network security requirements.

The usage of VEREFOO is simplified by a functional graphical user interface through which a network topology can be designed with a drag-n-drop feature, and specifying network security requirements without having any knowledge in managing XML files.

After have run the Java Spring Boot, related to the Spring framework, it is possible to interact with VEREFOO in order to generate the FW allocation schema and the configuration for all FW instances (VEREFOO exposes a set of REST APIs)

³D. Bringhenti, F. Valenza, "Optimizing distributed firewall reconfiguration transients", Computer Networks, 2022

⁴D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, "Automation for network security configuration: state of the art and research trends", ACM Computing, 2023

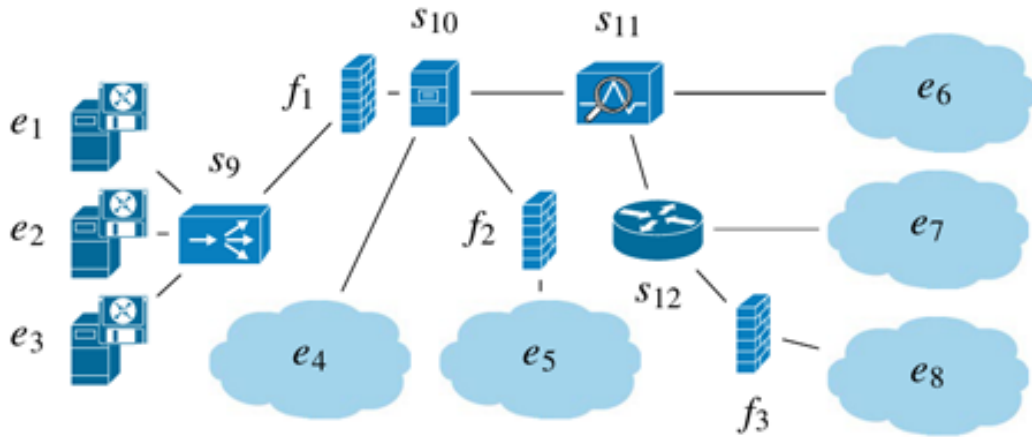


Figure 13.4. Firewall Allocation Scheme output

Firewall fw ₁						
#	Action	IPSrc	IPDst	pSrc	pDst	tProto
1	Allow	220.124.30.1	130.10.0.4	*	80	TCP
2	Allow	40.40.41.*	130.10.0.4	*	80	TCP
3	Allow	130.10.0.4	*.*.*.*	*	*	*
D	Deny	*.*.*.*	*.*.*.*	*	*	*

Firewall fw ₂						
#	Action	IPSrc	IPDst	pSrc	pDst	tProto
1	Allow	40.40.42.*	40.40.41.*	*	*	*
2	Allow	88.80.84.*	40.40.42.*	*	*	*
D	Deny	*.*.*.*	*.*.*.*	*	*	*

Firewall fw ₃						
#	Action	IPSrc	IPDst	pSrc	pDst	tProto
1	Allow	*.*.*.*	192.168.*.*	*	*	*
D	Deny	*.*.*.*	*.*.*.*	*	*	*

Figure 13.5. Firewall rules output

also drawing and showing the topology with the minimum number of firewalls the software computed, while taking into consideration specified requirements.

The software is set to define filtering rules into every FW in order to be the minimum possible (all information are represented in the xml file received from the VEREFOO REST API, including FW allocation schema and an explicit declaration if a specific requirements has been satisfied or not).

XML file received cannot be used directly because it is a not standardized configuration file. For this reason, a translation to low level configuration file (types available are iptables, bpf and ovs) needs to be performed still via a POST request. Once got the FAS (FW Allocation Schema), the virtual environment needs to be created including some additional information in the network topology (allocation graph sent as input to VEREFOO included just some necessary information) like presence of L2 switches and all interfaces IP addresses with all the intermediate network functions in the network.

The result is the building of a virtual environment where dockers are created for a single node composing the input network topology⁵.

A check for a correctness of FW configuration is done via commandline sending test packages specifying source IP and port, but also number of packets to send, to check if only proper FWs will let packets pass. The last test is on iptables within FWs in the network.

Via a graphical user interface a manual topology can be drawn sending to VEREFOO so a new FW Allocation Scheme will be recomputed (an XML schema can be imported and sent to VEREFOO)^{6 7}.

⁵D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated optimal firewall orchestration and configuration in virtualized networks" in NOMS - IEEE/IFIP Network Operations and Management Symposium, IEEE, pp. 1–7., 2020

⁶D. Bringhenti, R. Sisto, F. Valenza, "A novel abstraction for security configuration in virtual networks", Computer Networks, 2023

⁷D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Automated Firewall Configuration in Virtual Networks", IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1559-1576, 2023

Chapter 14

MiddleVerefoo Project

The scope of work of this Thesis is to extend the functionality currently available on VEREFOO by creating an interface with IDPS. The newly developed software, is named MiddleVerefoo.

It is intended as a kind of plug-in, an external module that remains listening on files (defined through a customizable path) or on ports chosen by the administrator. These fields are configurable through the configuration file.

The software has been written in Java to take advantage of the full potential of this language. The version used is the 1.8, the same version of VEREFOO. This is to achieve maximum compatibility between the two systems and in expectation of incorporation into VEREFOO.

Only native Java libraries were used in the implementation, so that consequences resulting from copyright and any cyber risk related to use of third-party code can be considered totally excluded.

MiddleVerefoo development

It started from paragraph "IDPS Comparison" of this Thesis where all the most suitable IDPS solutions, were explained in their features and highlights.

The objective was to select the most suitable IDS and IPS products to be selected as a source to be interfaced with MF.

The software was created around the selection made, which assumed, clearly, a key importance in writing the code.

All the security products considered are very valuable. Each of them brings with it, obviously, pros and cons that were seriously considered when choosing the most convenient IDS and IPS for the purpose.

The evaluation parameters were:

- open-source solution;
- reliable and stable system;
- simplicity of configuration;
- comprehensive documentation;
- good support community;
- flexible and suitable for different operating systems.

Pros and cons of the products are easily presented in the following overview:

Product	IDS	IPS	Host-Based	Network-Based	Multi-OS
Snort	Y	Y	N	Y	Y
Suricata	Y	Y	N	Y	Y
Zeek (formerly Bro)	Y	N	N	Y	Y
OSSEC	Y	N	Y	N	Y
EasyIDS	Y	N	N	Y	N
Untangle	Y	Y	N	Y	N
AIDE	Y	N	Y	N	Y
Samhain	Y	Y	Y	N	Y

Figure 14.1. IDPS comparison overview

For IPS, after an initial skimming, Snort and Suricata came to the podium. All solutions that are not explicitly dedicated to the function of IPS, which were too computationally overwhelming, were excluded. Were also excluded IDPS dedicated operating systems, which are out of scope.

Suricata and Snort are both very popular open-source.

Suricata was developed in 2009 as an alternative to Snort and is known for its speed and flexibility. It has advanced features such as parallel processing, decoding, and protocol analysis.

It is considered one of the fastest intrusion detection systems and can handle large amounts of network traffic, which are not, however, among the project requirements. Snort is very reliable (it is "signed" by Cisco Systems, after the purchase of Sourcefire in 2013), and has been in use since 1998. It is known for its ability to detect and prevent attacks in real time and is highly customizable due to its wide range of rules. It also supports creating custom outputs and writing scripts to analyze network traffic.

It has been used in a variety of contexts, including government and military networks, and has proven to be a reliable and time-tested system.

For the project, a system with a wide range of customizable rules that is robust, reliable, and proven over time is the best choice.

Snort is the selected IPS solution.

As for the IDS, OSSEC was chosen.

The rationale, in addition to its extreme ease of configuration, is clearly presented in the overview below:

Product	HOST-BASED									
	Active Response	Signature-based	Anomaly-based	Buffer overflow	System call	File check integrity	File access attempt	Rookit		
OSSEC	Y	Y	Y	Y	Y	Y	Y	Y		
AIDE	N	N	N	N	N	Y	N	N		
Samhain	Y	Y	N	Y	Y	Y	Y	Y		

Figure 14.2. Host-based IDS comparison overview

As can be clearly observed, the domain of OSSEC is exclusively host-related, not network-based.

This deficiency is not a problem at all and it does not result in any functionality deficit on the software.

The motivation is in the choice made previously for the IPS and directed specifically to fill this shortage: SNORT turns out to be an excellent NIDS, whenever the need occurs.

The reason for having a NIDS was explained earlier in this Thesis, in the section "Differences between IDS and IPS."

Among the most crucial determinants is the scope of application of these technologies: in the Operation Technology (OT) field, the IPS solution, in case of false positives, would block some lawful traffic, causing an interruption of service.

OSSEC and Snort products with versions 3.6.0 and 2.9.20, respectively, were used in the study.

The app's operation is tied to events captured by OSSEC and Snort: as soon as the OSSEC IDS or Snort IPS detects any attack in progress, the event log will be sent to the software, which will parse it and produce a security policy written according to the VEREFOO XML schema. This security policy will become the responsibility of the system administrator who will send it to the Firewalls in scope. Automatic forwarding has been deliberately left out to leave the administrator with the decision-making responsibility for managing the policies output to a specific FW type (internal or perimeter).

MiddleVerefoo components

In the "/src/config" file, the administrator can set the software to remain listening on specific ports or if the IDPS is expected to write the event logs to a specific file, defined similarly.

The IDPS product will have to be configured, accordingly, to send logs to the ports on which MV (MiddleVerefoo) is listening or by writing to the files set in the configuration file "ids.conf" and "ips.conf".

Due to a limitation of Java's native WatchService library, and to ensure proper handling of events, the files that IDPSs will modify must be saved in two different folders (OSSEC and Snort, for example).

In general, this library continuously monitors contents of a folder by generating an event in case a file is created (ENTRY_CREATE), modified (ENTRY_MODIFY), deleted, moved, or renamed (ENTRY_DELETE); or for any event that does not fall into the previous cases (OVERFLOW).

In MV, monitoring is then performed via the ENTRY_MODIFY function.

Once the program has been started, accordingly to configuration files mentioned before, the **watchRunner** class launches fileWatcher or tcp/udp socketListner that continuously listening on the file or on the ports to which the IDPS is configured. Additionally, the full path of the expected output XML file, must be passed as an argument.

With each new security event recorded by the IDPS, the **logParser** classes will be involved, which will manipulate the raw logs, making them usable to the application.

The disposable format for OSSEC is .json.

For Snort, has been chosen to tune MiddleVerefoo so that it accepts the "snort fast"

format, which includes all the necessary parameters to create a security policy fit for the purpose.

In Snort, raw log "fast" and "full" refer to two different levels of log detail that can be generated by the intrusion detection systems.

- **Raw log "Fast"**: is a simpler and more compact log format than the raw log "full." This type of log is designed to provide only essential information about a detected event, minimizing the workload of the logging system. "Fast" logs are particularly useful in high-traffic environments or when it is necessary to maintain a small volume of logs such as application logs.

- **Raw log "Full"**: is a more detailed log format that includes more complete information about the detected event. This type of log provides a more in-depth analysis of the detected network activity, including details of the packets involved in the event. "Full" logs may contain information such as IP header, TCP/UDP header, packet data and other relevant protocol information. However, because these logs are more detailed, they may generate a larger volume of data and require more resources for storage and analysis.

The use of "fast" logs is congenial to the output of security policies, the purpose of the software, but also to the storage capacity and resource usage that will be required to run MiddleVerefoo.

Examples of log formats, correctly understood by the software, are as follows:

- **OSSEC**

```
{ "rule":1000,
  "level":1,
  "comment":"This is a comment",
  "sidid":1111,
  "cve":"cve-1001-1000",
  "action":"drop",
  "srcip":"10.1.1.1",
  "srcport":"1000",
  "srcuser":"root",
  "dstip":"10.2.2.2",
  "dstport":"2000",
  "dstuser":"root",
  "location":"/var/log/auth.log",
  "full_log":"This is the full log message",
  "file":{
    "path":"/etc/",
    "md5_before":"","",
    "md5_after":"","",
    "sha1_before":"","",
    "sha1_after":"","",
    "owner_before":"root",
    "owner_after":"nobody",
    "gowner_before":"root",
    "gowner_after":"nodody",
    "perm_before":660,
```

```
"perm_after":777,  
}  
}
```

- **Snort_short**

```
07/14-2023:12:34:56.789012 [**] [1:2013456:3] MALWARE-OTHER Suspicious  
outbound connection detected [**] [Classification: A Network Trojan was de-  
tected] [Priority: 1] TCP 192.168.1.100:12345 -> 203.0.113.1:80
```

In the latter log, the information are not categorized explicitly. The meaning is explained below:

- the date and time 07/14-2023:12:34:56.789012 indicate when the event was detected;
- the header in square brackets [] indicates a warning message or relevant information about the event;
- the number in parentheses [1:2013456:3] represents the signature of the detected event;
- the description of the event is provided after the signature;
- the source and destination IP addresses and ports (SRC IP:SRC PORT → DST IP:DST PORT) indicate the network information about the event.

The class **xmlCreator** refers to the official VEREFOO XML schema found at <https://raw.githubusercontent.com/netgroup-polito/verefoo/master/xsd/nfvSchema.xsd>.

This class will names the output file consequently to the full path entered previously. The event epoch time will be added automatically to the output filename, so different security policies can be easily distinguished.

Different security events by IDPS side, will be translated into different security policies as output. Inside them there will be written clearly source and destination IPs and ports, but also the Internet protocol identified (UDP, TCP, OTHER or Any if no value set into the field "lv4proto").

Another important feature written within this software is the class **VFLogger**.

Apart from debugging, logging is also important from multiple other perspectives, like event tracing, request tracing, security, and Business intelligence can also be directly fueled by data produced by logs. With the help of logs, information about what is happening in the application can be easily get with a record of errors and unusual circumstances.

Also for the logging function, no third-party library was used although it would allow advanced and more convenient functions. It was decided to use Java's native library `java.util.logging`.

The package provides the following logging levels in descending order of gravity:

- SEVERE(HIGHEST LEVEL)
- WARNING
- INFO
- CONFIG
- FINE
- FINER
- FINEST(LOWEST LEVEL)

Apart from this, the above package also provides two additional levels ALL and OFF used for logging all messages and disabling logging respectively.

The logging configuration file will be found at the path "config/logger.properties". Inside it the administrator can define the file path, the maximum number of records but also if new log event will be append inside the same file.

The two used handlers in the MiddleVerefoo java.util.logging package are as follows:

- **FileHandler**, that writes the log message a file;
- **ConsoleHandler**, that writes the log message to the console.

As concerns, however, formatting and convert data in a log event, as default in the logger configuration file, the SimpleFormatter is set. It generates text messages with basic information.

Chapter 15

Conclusions

A firewall is a network security device designed to separate two networks having different levels of trustness. It monitors and controls incoming and outgoing network traffic based on predetermined security rules.

Intrusion Detection and Prevention System (IDPS) combines the capabilities of an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) to provide both detection and prevention of some cyber security events in a system.

A firewall provides the first line of defense by controlling traffic based on predefined rules, an IDPS offers more in-depth analysis and real-time threat detection and prevention.

Verified Refinement and Optimized Orchestration software (VEREFOO) was designed for managing security configurations on firewalls automatically.

It is a perfect solution especially for large networks that are the hardest to configure where the problem of vulnerabilities management and misconfigurations are always around the corner.

VEREFOO has worked so far only with packet filter firewalls allowing configuration updates between them: its aim is to automatically orchestrates and configures network security functions in virtualized networks. This is achieved through the definition of a proper allocation scheme and then compute the optimal configuration, consecutively.

The research work done with this Thesis has led to an important evolution about the functionality of VEREFOO by designing an interface with the Intrusion Detection System (OSSEC) and Intrusion Protection System (Snort) products. These two security solutions have been carefully chosen to achieve an excellent level of strength, reliability, and functionality.

The aim of the software is to transform all security events detected by those technologies into security policies carrying the same official VEREFOO XML schema. The security policies created can be managed by the system administrator, who will decide on their distribution to specific firewall categories (whether they are internal or perimeter FWs). The attack detected by the IDPS in the vicinity of a certain FW will be used to update blocklists on all other FWs in scope, resulting in a corporate network that is always up-to-date in terms of protection from attacks already suffered or even in progress. The benefits of MiddleVerefoo are now clear. Initially the software has been designed as an external module, running continuously, waiting for updated operations on files or guarded ports operated by IDPSs.

It is not excluded that in the future it may be integrated directly within VEREFOO. This is exactly why it is written in Java, which is also known for its flexibility to be used on different platforms, in version 1.8, which is the same version with which VEREFOO was written.

Future developments of VEREFOO will steer it to integrate perfectly with all other security devices as well. Web Application Firewall (WAF) or Endpoint Detection and Response (EDR), for example, will write new logs to produce updated security policies, then distribute them to all FWs in the scope network.

This will ensure expanded levels of protection and making the corporate perimeter increasingly more secure from the various cyberattacks.

Bibliography

- [1] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated optimal firewall orchestration and configuration in virtualized networks" in NOMS - IEEE/IFIP Network Operations and Management Symposium, IEEE, pp. 1–7., 2020
- [2] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, "Automation for network security configuration: state of the art and research trends", ACM Computing, 2023
- [3] D. Bringhenti, R. Sisto, F. Valenza, "A novel abstraction for security configuration in virtual networks", Computer Networks, 2023
- [4] D. Bringhenti, G. Marchetto, R. Sisto and F. Valenza, "A novel approach for security function graph configuration and deployment", IEEE 7th International Conference on Network Softwarization (NetSoft), pp. 457-463, 2021
- [5] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Automated Firewall Configuration in Virtual Networks", IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1559-1576, 2023
- [6] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Introducing programmability and automation in the synthesis of virtual firewall rules", 6th IEEE Conference on Network Softwarization (NetSoft), pp. 473-478, 2020
- [7] D. Bringhenti, F. Valenza, "Optimizing distributed firewall reconfiguration transients", Computer Networks, 2022
- [8] D. Bringhenti and F. Valenza, "A Twofold Model for VNF Embedding and Time-Sensitive Network Flow Scheduling," in IEEE Access, vol. 10, pp. 44384-44399, 2022
- [9] D. Bringhenti, G. Marchetto, R. Sisto, S. Spinoso, F. Valenza and J. Yusupov, "Improving the Formal Verification of Reachability Policies in Virtualized Networks" in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 713-728, 2021
- [10] G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "A Framework for Verification-Oriented User-Friendly Network Function Modeling," in IEEE

- Access, vol. 7, pp. 99349-99359, 2019
- [11] G. Marchetto, R. Sisto, F. Valenza, J. Yusupov and A. Ksentini, "A Formal Approach to Verify Connectivity and Optimize VNF Placement in Industrial Networks" in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1515-1525, 2021
- [12] D. Bringhenti, F. Valenza and C. Basile, "Toward Cybersecurity Personalization in Smart Homes" in *IEEE Security Privacy*, vol. 20, no. 1, pp. 45-53, 2022
- [13] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza and J. Yusupov, "Towards a fully automated and optimized network security functions orchestration", 4th International Conference on Computing, Communications and Security (ICCCS), pp. 1-7, 2019
- [14] Scarfone, K., Mell, P., Romanosky, S., "Guide to Intrusion Detection and Prevention Systems (IDPS)", (NIST Special Publication 800-94), National Institute of Standards and Technology, 2020
- [15] Northcutt, S., Novak, J., "Network Intrusion Detection: An Analyst's Handbook (3rd ed.)", New Riders, 2020
- [16] Roesch, M., "Snort Intrusion Detection and Prevention Toolkit", Syngress, 2007
- [17] Stamp, M., "Information Security: Principles and Practice (2nd ed.)", Wiley, 2011
- [18] Cheswick, W. R., Bellovin, S. M., Rubin, A. D., "Firewalls and Internet Security: Repelling the Wily Hacker (2nd ed.)", Addison-Wesley Professional, 2003
- [19] Cole, E., Krutz, R. L., Conley, J., "Network Security Bible (2nd ed.)", Wiley, 2013
- [20] Skoudis, E., Liston, T., "Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses (2nd ed.)", Prentice Hall, 2014
- [21] Cole, E., Jacobs, A., "Advanced Persistent Threat: Understanding the Danger and How to Protect Your Organization", Syngress, 2015
- [22] Kumar, S., "Intrusion Detection Systems", CRC Press, 2014
- [23] Zou, C. C., Gong, W., "Security Monitoring: Proven Methods for Incident Detection on Enterprise Networks", Syngress, 2011
- [24] Scambray, J., Shema, M., "Hacking Exposed: Network Security Secrets and Solutions (6th ed.)", McGraw-Hill Education, 2009

- [25] Vacca, J. R., "Computer and Information Security Handbook (2nd ed.)", Morgan Kaufmann, 2013.