CHAPTER 1
INTRODUCTION

In an era characterized by an ever-evolving digital landscape and an exponential increase in cyber threats, the need for robust and adaptive cybersecurity measures has become paramount. Cyberattacks pose significant risks to individuals, organizations, and nations alike, making it imperative to develop advanced systems capable of identifying, preventing, and responding to these threats effectively.

1.1 OVERVIEW

Intrusion Detection and Prevention Systems (IDPS) serve as the frontline defense against a multitude of cyber threats. These systems play a crucial role in safeguarding networks and information systems by monitoring activities and identifying unauthorized access, malicious behaviors, and potential vulnerabilities. However, the evolving nature of cyber threats demands a more sophisticated and agile approach to intrusion detection and prevention.

Virtualization technology has emerged as a powerful tool in enhancing the security posture of organizations. By isolating and compartmentalizing systems within virtual environments, vulnerabilities can be minimized, and potential threats contained. Leveraging virtualization in the context of intrusion detection and prevention not only improves security but also offers scalability and flexibility in deploying and managing security resources.

Threat intelligence, consisting of actionable information about potential threats, vulnerabilities, and attack patterns, is instrumental in enhancing the effectiveness of security systems. Real-time monitoring complements threat intelligence by providing continuous surveillance of network and system activities, allowing for the immediate detection of security incidents. The fusion of threat intelligence and real-time monitoring equips organizations with the ability to proactively defend against evolving cyber threats.

CHAPTER 2
LITERATURE SURVEY

The literature survey on Intrusion Detection and Prevention Systems (IDPS) reveals a rich and evolving landscape in the field of cybersecurity. The historical evolution of IDPS, dating back to the early origins of rule-based signature detection, showcases its pivotal role in safeguarding digital assets. Over time, IDPS has transitioned to behavior-based detection mechanisms, adapting to the growing complexity of cyber threats. This transformation led to the emergence of hybrid IDPS, which adeptly combines signature and behavior-based techniques, offering a more comprehensive approach to threat detection. Within the realm of IDPS, various types have evolved to cater to specific security needs, including Network-Based IDPS (NIDPS) for network-wide monitoring, Host-Based IDPS (HIDPS) for individual endpoints, Application-Based IDPS (AIDPS) for application-layer protection, Cloud-Based IDPS for cloud environments, and Wireless IDPS (WIDPS) for securing wireless networks. This literature review underscores the dynamic nature of IDPS, which has continually adapted to counter emerging threats, ultimately setting the stage for the development of a Virtualized Hybrid Intrusion Detection and Prevention System, as proposed in this project, with the integration of threat intelligence and real-time monitoring capabilities, thus aiming to address the evolving challenges in modern cybersecurity.

2.1 INTRUSION DETECTION AND PREVENTION SYSTEMS (IDPS)

Intrusion Detection and Prevention Systems (IDPS) are foundational components of modern cybersecurity infrastructure and have evolved significantly over the years, the literature review should encompass their historical development and the diverse types of IDPS.

2.1.1 HISTORICAL EVOLUTION OF IDPS

The origins of intrusion detection systems can be traced back to the early days of computer networking when the primary focus was on securing mainframes. The need to safeguard these systems led to the creation of rudimentary intrusion detection mechanisms. These early systems primarily relied on rule-based approaches, where predefined signatures or patterns were used to identify known attack attempts. They marked the inception of signature-based IDPS.

Transition to Behavior-Based Detection

As computing environments grew in complexity, attackers developed more sophisticated methods to evade signature-based detection. This necessitated the evolution of intrusion detection towards behavior-based approaches. In the late 20th century, research began to shift towards anomaly detection, where systems learned normal behavior patterns and alerted on deviations. This marked the emergence of behavior-based IDPS.

Emergence of Hybrid IDPS

The limitations of both signature and behavior-based approaches prompted the development of hybrid IDPS. Hybrid systems combine the strengths of both techniques, utilizing signature-based detection for known threats and behavior-based detection for identifying anomalies. The hybrid approach is particularly effective in addressing the challenges posed by zero-day exploits and advanced persistent threats (APTs).

2.1.2 VARIOUS TYPES OF IDPS

Network-based IDPS (NIDPS) monitors network traffic and analyzes it for signs of malicious activity. It operates at the network level, allowing it to detect attacks like port scanning, DDoS, and suspicious traffic patterns. NIDPS can be deployed at various points within a network, such as at the perimeter, within subnets, or on critical servers.

Host-based IDPS (HIDPS) focuses on individual hosts or endpoints within a network. It monitors activities on a host, including file system changes, logins, and system calls. HIDPS is particularly useful for detecting insider threats and attacks targeting specific hosts.

Application-based IDPS (AIDPS) is designed to protect specific applications or services. It focuses on the security of the application layer and can identify vulnerabilities or attacks targeting web applications, databases, and other critical services.

With the increasing adoption of cloud computing, cloud-based IDPS has gained prominence. These systems are tailored to protect virtualized environments and cloud services. They leverage the flexibility and scalability of the cloud to provide real-time threat detection and prevention.

Wireless IDPS (WIDPS) is specialized in monitoring and securing wireless networks. It can detect unauthorized access points, rogue devices, and attacks targeting Wi-Fi networks, making it crucial for organizations with wireless infrastructure.

2.2 VIRTUALIZATION IN CYBERSECURITY

Virtualization technologies have become indispensable tools in bolstering cybersecurity measures. This section explores the multifaceted role of virtualization in cybersecurity, examining various virtualization techniques and their applications.

2.2.1 VIRTUAL MACHINES (VMS) AND CONTAINERS

Understanding Virtual Machines (VMs)

Virtual Machines (VMs) are software-based emulations of physical computers. They enable the creation of multiple isolated environments, each with its own operating system (OS) and applications, running on a single physical host. This isolation provides a crucial security benefit by segmenting different workloads and applications, preventing vulnerabilities or breaches in one VM from affecting others. VMs are widely used in cybersecurity for tasks such as sandboxing potentially malicious files, running vulnerable software in controlled environments for analysis, and isolating critical security components.

Containers: Lightweight Virtualization

Containers represent a lightweight form of virtualization, offering efficient and scalable isolation without the overhead of full VMs. Unlike VMs, containers share the host OS kernel while maintaining separate user spaces, making them highly resource-efficient. Containers are particularly valuable in cybersecurity for packaging and deploying security services, such as intrusion detection systems (IDS) and firewalls, in a consistent and portable manner. They facilitate rapid application deployment and scaling, streamlining the deployment of security components across diverse environments.

Security Considerations with VMs and Containers

While VMs and containers enhance security, they introduce unique challenges. VM escape vulnerabilities and container breakout risks demand meticulous security configuration. Effective network segmentation, proper access controls, and continuous monitoring are essential for securing VM and container environments. Additionally, tools like Docker, Kubernetes, and container orchestration platforms have become pivotal for managing containerized security applications at scale.

2.2.2 MICROSEGMENTATION

Defining Microsegmentation

Microsegmentation is an advanced network security technique made possible through virtualization. It involves dividing a network into small, isolated segments, each with its own security policies and controls. Unlike traditional network security, where security policies are applied at the perimeter or between network zones, microsegmentation enforces policies at the granular level, down to individual workloads or applications. This approach reduces the attack surface and limits lateral movement within a network, thwarting attackers' attempts to move freely within compromised systems.

Use Cases and Benefits

Microsegmentation finds applications in securing modern data centers and cloud environments. It allows organizations to implement a Zero Trust security model, where trust is never assumed, and verification is continuous. By segmenting networks and applying access controls based on identity, device posture, and behavior, organizations can significantly enhance their security posture. Microsegmentation is particularly effective in thwarting insider threats and containing the spread of malware within a network.

Implementation and Challenges

Implementing micro segmentation requires careful planning and policy definition. Software-defined networking (SDN) and network virtualization technologies play a crucial role in enabling microsegmentation. Challenges include policy complexity, scalability, and ensuring that legitimate communication is not unduly restricted. Successful implementation demands a balance between security and operational efficiency.

Virtualization in cybersecurity, encompassing VMs, containers, and microsegmentation, revolutionizes the way organizations protect their digital assets. By isolating, compartmentalizing, and securing workloads, these virtualization techniques provide a robust foundation for safeguarding against an ever-evolving threat landscape. However, organizations must navigate security challenges specific to virtualized environments to realize the full benefits of these technologies.

2.3 THREAT INTELLIGENCE

Threat intelligence forms the cornerstone of proactive cybersecurity efforts, providing organizations with vital information and context to defend against a wide array of cyber threats. This section offers an in-depth exploration of threat intelligence, encompassing its fundamental concepts, types, integration strategies, and real-world applications.

2.3.1 THREAT INTELLIGENCE FEEDS

Defining Threat Intelligence Feeds

Threat intelligence feeds are repositories of actionable information about potential threats, vulnerabilities, and attack patterns. These feeds are typically aggregated from various sources, including open-source intelligence (OSINT), commercial threat feeds, government sources, and private security research. They provide organizations with timely and relevant data to enhance their cybersecurity posture.

Types of Threat Intelligence Feeds

Threat intelligence feeds come in different types, each offering distinct insights into the threat landscape:

- Indicator-Based Feeds: These feeds provide specific indicators of compromise (IoCs) such as IP addresses, domain names, or file hashes associated with known threats.
- Tactics, Techniques, and Procedures (TTPs) Feeds: TTP feeds focus on the methods and behaviors employed by threat actors, offering insights into their attack strategies.
- Strategic Intelligence Feeds: These feeds provide high-level insights into threat actor groups, their motivations, and geopolitical context, helping organizations anticipate targeted attacks.

2.3.2 THREAT INTELLIGENCE INTEGRATION

The Role of Threat Intelligence in Cybersecurity

Threat intelligence integration is a critical component of a robust cybersecurity strategy. It empowers organizations to identify and mitigate threats swiftly, enhancing their security posture. Threat intelligence aids in proactive threat detection, incident response, vulnerability management, and threat hunting.

Automated Integration

Automated integration of threat intelligence into security systems is a best practice for real-time threat detection and response. Security Information and Event Management (SIEM) platforms, Intrusion Detection and Prevention Systems (IDPS), and endpoint security solutions can ingest threat intelligence feeds to correlate incoming data with known threats. Automated responses, such as blocking malicious IP addresses or quarantining infected endpoints, can be triggered based on threat intelligence.

Challenges and Considerations

Effective threat intelligence integration poses challenges related to data quality, relevance, and privacy. Organizations must validate and contextualize threat intelligence to ensure its accuracy and applicability to their environment. Privacy concerns also arise when sharing threat intelligence with third parties. Striking a balance between sharing and safeguarding sensitive information is paramount.

2.3.3 APPLICATIONS

Threat Intelligence in Incident Response

Threat intelligence plays a pivotal role in incident response by providing context around detected threats. It helps incident response teams understand the severity of an incident, its origins, and potential impact. This information enables rapid containment and remediation.

Proactive Threat Hunting

Threat intelligence feeds can fuel proactive threat hunting initiatives. Security analysts use threat intelligence to proactively search for signs of advanced threats that may evade traditional security measures. This approach allows organizations to identify and neutralize threats before they cause damage.

Vulnerability Management

Threat intelligence aids in vulnerability management by highlighting vulnerabilities that are actively exploited in the wild. Organizations can prioritize patching or mitigating these vulnerabilities to reduce their exposure to known threats.

Threat intelligence is not merely an information source; it is a strategic asset that empowers organizations to defend against the ever-evolving landscape of cyber threats. By leveraging various types of threat intelligence feeds and integrating them into their security infrastructure, organizations can proactively protect their digital assets, enhance incident response capabilities, and stay ahead of threat actors.

2.4 REAL-TIME MONITORING

Real-time monitoring is a cornerstone of proactive cybersecurity, enabling organizations to maintain continuous visibility into their network and system activities. This section delves deeply into real-time monitoring, covering its underlying technologies, benefits, applications, and challenges.

2.4.1 REAL-TIME MONITORING TECHNOLOGIES

Security Information and Event Management (SIEM) systems serve as central hubs for real-time monitoring. They collect and analyze security data from various sources, including logs, network traffic, and security events. SIEM platforms use correlation rules and machine learning algorithms to detect anomalies, threats, and vulnerabilities. Real-time alerts and notifications are generated when suspicious activities are identified, enabling immediate response.

Network Traffic Analysis (NTA) tools focus on monitoring network traffic patterns and behaviors. They employ deep packet inspection and behavioral analytics to identify anomalies indicative of attacks or compromised hosts. NTA tools excel in detecting network-based threats, including lateral movement and data exfiltration.

Endpoint Detection and Response (EDR) solutions provide real-time visibility into endpoints such as workstations and servers. They monitor processes, file changes, registry modifications, and network connections on endpoints. EDR solutions enable rapid detection of endpoint-related threats, including malware infections and unauthorized access.

2.4.2 BENEFITS OF REAL-TIME MONITORING

Reducing Dwell Time

Real-time monitoring significantly reduces dwell time—the duration between a security breach and its discovery. By promptly detecting and alerting on security incidents, organizations can mitigate the potential damage and limit the lateral movement of attackers within their networks.

Minimizing Data Breaches

Real-time monitoring helps minimize data breaches by identifying unauthorized access attempts and data exfiltration in their early stages. This proactive approach safeguards sensitive information and regulatory compliance.

Improving Incident Response Times

Rapid detection through real-time monitoring streamlines incident response efforts. Security teams can initiate containment and remediation actions swiftly, reducing the impact of security incidents and minimizing business disruption.

2.4.3 CHALLENGES OF REAL-TIME MONITORING

Data Volume and Noise

The volume of data generated in real-time can be overwhelming, leading to false positives and alert fatigue. Effective data filtering and normalization are essential to focus on genuine threats and reduce noise.

Advanced Threats

Real-time monitoring may struggle to detect advanced threats that employ sophisticated evasion techniques. Threat actors continuously evolve their tactics, making it challenging to identify their activities.

Privacy and Compliance

Real-time monitoring involves the collection and analysis of sensitive data, raising privacy and compliance concerns. Organizations must balance the need for security with data protection and regulatory requirements.

2.4.4 REAL-WORLD APPLICATIONS

Incident Response and Investigation

Real-time monitoring plays a pivotal role in incident response and forensic investigations. It provides forensic analysts with real-time data and contextual information to trace the origins and impact of security incidents.

Threat Hunting

Security teams use real-time monitoring data for proactive threat hunting. Threat hunters analyze real-time data to uncover hidden threats, emerging attack patterns, and signs of compromise that may evade automated detection.

Insider Threat Detection

Real-time monitoring helps organizations identify insider threats, including employees or partners with malicious intent. Suspicious user behaviors or data access patterns can trigger alerts, allowing for timely intervention. Real-time monitoring is a critical component of modern cybersecurity, offering continuous visibility and rapid response capabilities. Leveraging technologies like SIEM, NTA, and EDR, organizations can reduce dwell time, minimize data breaches, and improve incident response. However, addressing challenges related to data volume, advanced threats, and privacy is essential to harness the full potential of real-time monitoring in safeguarding digital assets.

CHAPTER 3
SYSTEM ENVIRONMENT

3.1 SYSTEM ARCHITECTURE

Designing the Hybrid Architecture

The first step in our methodology is the design of the hybrid architecture for our Virtualized Intrusion Detection and Prevention System. We will create a blueprint that combines the strengths of signature-based and behavior-based detection. This architecture will involve the deployment of virtual machines (VMs) or containers to compartmentalize security components within a controlled environment.

Integrating Threat Intelligence

We will integrate threat intelligence feeds from reputable sources into our architecture. This integration will require the development of data pipelines and mechanisms for real-time updates of threat intelligence information. The aim is to enrich our system with up-to-date knowledge about known threats, vulnerabilities, and attack patterns.

Real-Time Monitoring Component

A critical part of our architecture is the real-time monitoring component. We will deploy Security Information and Event Management (SIEM) systems and network traffic analysis (NTA) tools to continuously monitor network and system activities. This component will analyze incoming data, generate alerts, and correlate events to detect security incidents as they occur.

3.2 DATA SOURCES

Identifying Relevant Data Sources

In this phase, we will identify the data sources that our system will monitor. These sources may include system logs, network traffic logs, server logs, and application logs. We will also consider external data sources such as threat intelligence feeds and third-party logs.

Data Collection Mechanisms

To collect data from identified sources, we will implement data collection mechanisms. This may involve deploying agents on endpoints, configuring network taps, or utilizing log forwarding techniques. We will ensure that data collection is efficient, secure, and compliant with privacy regulations.

Preprocessing Data

Before analysis, collected data will undergo preprocessing. This step involves data cleaning, normalization, and transformation to ensure consistency and accuracy. Data preprocessing is critical for effective threat detection and incident analysis.

3.3 DATA COLLECTION AND PROCESSING

Behavior-based analysis is a key component of our methodology. We will develop algorithms and models to establish baselines for normal behavior within the network and system. Any deviations from these baselines will be flagged as potential security incidents, allowing us to detect previously unknown threats.

Signature-based detection will be integrated to identify known threats based on predefined patterns. We will update signature databases regularly to keep them current. This combination of behavior-based and signature-based detection enhances our system's ability to identify a wide range of threats.

Real-time monitoring and alerting will be implemented to provide immediate notifications of security incidents. Alerts will be categorized by severity and sent to designated personnel or response teams for further investigation and action.

3.4 EVALUATION AND VALIDATION

Testing Scenarios

To evaluate the effectiveness of our system, we will define a range of testing scenarios that simulate various cyber threats, including malware infections, phishing attacks, and network intrusions. These scenarios will test the system's detection and prevention capabilities.

Performance Metrics

We will establish performance metrics, such as false positive rates, false negative rates, detection time, and resource utilization. These metrics will allow us to quantitatively assess the system's performance and fine-tune its configuration.

Continuous Improvement

Continuous improvement is a crucial aspect of our methodology. We will use the results of our evaluation to refine and enhance the system. Regular updates to threat intelligence feeds and analysis algorithms will be part of our ongoing efforts to adapt to emerging threats.

CHAPTER 4
OVERVIEW

4.1 INTRODUCTION

The Virtualized Hybrid Intrusion Detection and Prevention System with Threat Intelligence and Real-Time Monitoring is a critical response to the ever-evolving cyber threat landscape, aiming to enhance cybersecurity measures by combining advanced technologies and strategies. This overview will outline the key components, objectives, scope, and the broader context within which our project operates.

4.2 VIEWS

Addressing the Evolving Threat Landscape

One of the primary objectives of our project is to address the ever-evolving threat landscape. Cyber threats continue to evolve in sophistication and scale, posing significant risks to organizations of all sizes and industries. Our project aims to provide an effective defense mechanism against these threats by leveraging a combination of signature-based and behavior-based detection techniques, threat intelligence integration, and real-time monitoring.

Enhancing Security Posture

We seek to enhance the security posture of organizations by providing a comprehensive and adaptable solution. By deploying a virtualized architecture, our system not only isolates and compartmentalizes security components but also offers scalability and flexibility in adapting to changing security needs. This enhancement goes beyond traditional security measures to ensure a robust defense against a wide array of cyber threats.

4.3 SCOPE

Comprehensive System Architecture

Our project encompasses the design and development of a comprehensive system architecture. This architecture integrates virtualization technologies, including virtual machines (VMs) or containers, with threat intelligence feeds and real-time monitoring components. The scope includes the creation of a blueprint for system deployment and configuration.

Hybrid Detection Mechanisms

Our project's scope extends to the implementation of hybrid detection mechanisms. We will combine signature-based and behavior-based techniques to maximize threat detection capabilities. The hybrid approach enables us to identify known threats based on predefined patterns while also detecting previously unknown threats through behavioral analysis.

Threat Intelligence Integration

Integration of threat intelligence feeds is a significant aspect of our project. We will connect our system to reputable threat intelligence sources to ensure that it remains updated with the latest threat information. The scope includes the development of mechanisms for real-time threat intelligence updates and correlation.

Real-Time Monitoring

Continuous real-time monitoring forms a critical component of our project. Our system will actively surveil network and system activities, generating alerts and notifications in real-time. The scope includes the implementation of Security Information and Event Management (SIEM) systems and network traffic analysis (NTA) tools for effective monitoring.

4.4 SIGNIFICANCE OF THE PROJECT

Cybersecurity Resilience

Our project's significance lies in its contribution to cybersecurity resilience. In an era where cyber threats are an ever-present and constantly evolving menace, organizations need robust, adaptive, and proactive defenses. Our project's multifaceted approach equips organizations with the tools to withstand and respond to the most advanced threats.

Proactive Threat Defense

The significance of our project also lies in its emphasis on proactive threat defense. By combining threat intelligence, real-time monitoring, and hybrid detection mechanisms, we empower organizations to detect and mitigate threats as they happen, reducing the potential impact and damage.

Scalability and Adaptability

Our project addresses the need for scalable and adaptable security solutions. As organizations grow and technology evolves, security measures must keep pace. The virtualized architecture of our system allows for easy scalability and adaptation to meet changing security demands. As we proceed with the implementation and evaluation of our system, the following chapters will provide detailed insights into each aspect of our project, from architecture design to performance evaluation and results.

CHAPTER 5
COMPARISON OF EXISTING AND PROPOSED SOLUTION

5.1 OVERVIEW OF EXISTING SYSTEM

In this chapter, we dissect the differences between the existing cybersecurity infrastructure and our proposed Virtualized Deployment of Intrusion Prevention Systems (VDIPS). The current solution, while functional, often grapples with scalability, agility, and real-time threat detection challenges. Our proposed VDIPS, leveraging Proxmox for virtualization and a suite of advanced tools like Wazuh, Suricata, Zeek, OWLH, and the ELK stack, aims to transform the organization's cybersecurity posture by addressing these limitations.

- Legacy Infrastructure

  The current cybersecurity infrastructure relies heavily on legacy hardware-based solutions. These traditional setups often lack the flexibility to adapt to rapidly changing threat landscapes and do not efficiently utilize available resources.

- Limited Scalability

  The existing system struggles to scale horizontally or vertically in response to changing workloads or business expansion. Adding new hardware or security appliances is a cumbersome and costly process.

- Latency in Threat Detection

  Real-time threat detection is a challenge with the current setup. The latency between threat occurrence and detection can be significant, leaving the organization vulnerable to sophisticated attacks.

- Resource Overhead

  Legacy intrusion detection and prevention systems consume significant hardware resources, resulting in high operational costs and inefficiencies.

5.2 DISADVANTAGES OF THE EXISTING SYSTEM

- Hardware Limitations : Reliance on physical hardware constrains the system's ability to scale rapidly in response to increased workloads or evolving threats. Hardware maintenance and upgrades are resource-intensive and often involve significant downtime.

- Detection Latency: Real-time threat detection is hampered by the latency between threat occurrences and their identification. This delay exposes the organization to advanced threats that can exploit vulnerabilities during this gap.

- Manual Threat Updates: Threat intelligence updates are performed manually and periodically, leaving the system vulnerable to emerging threats until updates are applied. Manual updates require dedicated personnel and can result in human error, leading to missed vulnerabilities.

- Resource Inefficiency: Hardware-based solutions tend to overprovision resources to handle peak loads, resulting in underutilization during regular operations. This inefficiency leads to higher operational costs, increased energy consumption, and a larger environmental footprint.

- Fragmented Log Management: The existing setup lacks a centralized log management and analysis platform, resulting in fragmented log data across various devices. Fragmentation makes it challenging to correlate events and conduct comprehensive threat analysis.

5.3 OVERVIEW OF PROPOSED SYSTEM

- Virtualized Infrastructure with Proxmox

  Our proposed VDIPS leverages Proxmox for virtualization, marking a paradigm shift from traditional hardware-based setups. Proxmox allows for dynamic allocation of resources, enabling on-the-fly scalability and resource optimization.

- Integrated Threat Intelligence

  One of the primary differentiators is the integration of Threat Intelligence. Wazuh, Suricata, and Zeek are deployed within virtualized environments to provide signature-based and behavior-based threat detection. These components are updated with real-time threat intelligence feeds, ensuring up-to-the-minute threat awareness.

- Real-Time Monitoring and Analysis

  VDIPS incorporates OWLH for real-time monitoring and analysis. This enhances the system's ability to detect and respond to threats as they happen. The ELK stack (Elasticsearch, Logstash, Kibana) complements OWLH by providing centralized log management, analysis, and visualization.

- Agility and Scalability

  By virtualizing security components, VDIPS offers unmatched agility and scalability. New security instances can be spun up effortlessly to accommodate increased workloads or adapt to emerging threats, all within the Proxmox virtualization environment.

5.4 ADVANTAGES OF THE PROPOSED VDIPS SOLUTION

- Real-Time Scalability : VDIPS, hosted on Proxmox, offers real-time scalability by allowing the dynamic allocation of virtual resources. This enables the system to respond instantly to increased workloads or emerging threats without the need for lengthy procurement cycles.

- Instant Threat Detection : VDIPS excels in real-time threat detection, virtually eliminating detection latency. It detects and responds to threats as they happen, reducing the window of vulnerability to near-zero.

- Continuous Threat Intelligence : VDIPS integrates real-time threat intelligence feeds, ensuring that the system remains up-to-date with the latest threat information at all times. This continuous threat intelligence integration empowers the system to identify and mitigate emerging threats instantly.

- Resource Optimization : VDIPS optimized resource utilization through Proxmox's virtualization capabilities. Resources are allocated dynamically, minimizing operational costs and energy consumption. This efficiency aligns with modern sustainability goals, reducing the environmental impact of cybersecurity operations.

- Centralized Log Management and Analysis : VDIPS employs the ELK stack (Elasticsearch, Logstash, Kibana) for centralized log management, normalization, and visualization. This centralized approach offers a unified view of security events and activities, simplifying incident detection, investigation, and analysis.

5.5 COMPARISON

- Flexibility and Adaptability
    - Existing Solution

      The legacy infrastructure lacks the flexibility to adapt swiftly to emerging threats or changing business requirements. Adding new security appliances or updating software often entails a lengthy procurement and deployment process. This rigidity can result in vulnerability gaps and increased response times during incidents.

    - Proposed VDIPS Solution

      VDIPS, built on Proxmox, offers unparalleled flexibility and adaptability. The virtualized environment allows for the rapid provisioning of security components, enabling quick response to emerging threats or sudden spikes in traffic. This dynamic nature aligns seamlessly with the organization's evolving cybersecurity needs and reduces the window of vulnerability.

- Resource Optimization
    - Existing Solution

      Hardware-based solutions typically involve overprovisioning of resources to meet peak demands, resulting in underutilization during regular operations. This inefficiency leads to higher operational costs and a considerable environmental footprint.

    - Proposed VDIPS Solution

      VDIPS leverages Proxmox's resource allocation and optimization capabilities. Virtual machines or containers can be dynamically adjusted to match current workloads, ensuring resource utilization efficiency. This optimization minimizes costs, both in terms of hardware procurement and energy consumption, aligning with modern sustainability goals.

- Centralized Log Management and Visualization
  - Existing Solution

    The existing setup often lacks a centralized log management and analysis platform. Log data is distributed across various devices, making correlation and analysis cumbersome. This fragmentation hampers the ability to detect and respond to threats comprehensively.

  - Proposed VDIPS Solution

    VDIPS integrates the ELK stack (Elasticsearch, Logstash, Kibana) for centralized log management and visualization. This unification streamlines log collection, normalization, and analysis. Security teams gain a unified view of network and system activities, facilitating rapid incident detection and investigation. Kibana's intuitive dashboards offer visual insights into security events and trends.

- Real-Time Scalability
  - Existing Solution

    The existing cybersecurity infrastructure struggles to scale in real-time, often requiring the acquisition and installation of physical appliances or servers when facing increased workloads or sudden traffic spikes. This results in operational delays, increased capital expenditure, and prolonged vulnerability periods.

    - Proposed VDIPS Solution

      VDIPS introduces an era of real-time scalability. Proxmox's virtualization capabilities allow for the instant provisioning of additional virtual machines or containers to meet surging demands. Whether it's accommodating seasonal traffic variations or responding to a sudden increase in security event logs, VDIPS can dynamically allocate resources, reducing response time and minimizing exposure to threats.

- Threat Intelligence Integration
    - Existing Solution

      The current cybersecurity infrastructure often relies on periodic manual updates of threat intelligence databases, leaving organizations exposed to new threats until these updates are performed. This delay compromises the ability to detect and respond to the latest attack vectors effectively.

    - Proposed VDIPS Solution

      VDIPS integrates real-time threat intelligence seamlessly. Wazuh, Suricata, and Zeek, operating within virtualized environments, constantly receive and process threat intelligence feeds from reliable sources. This means that the system is always armed with the most current threat information, empowering it to identify and mitigate emerging threats instantly. The gap between threat discovery and response is practically non-existent, a stark contrast to the existing solution's reliance on periodic updates.

- Cost Efficiency
    - Existing Solution

      Legacy hardware-based cybersecurity solutions often come with high capital and operational costs. These expenses include the purchase of physical hardware, maintenance, power consumption, and cooling infrastructure. Over time, these costs can become prohibitive.

    - Proposed VDIPS Solution

      VDIPS transforms cost efficiency. The virtualized architecture hosted on Proxmox reduces capital expenditures by eliminating the need for dedicated physical appliances. Additionally, the dynamic allocation of resources minimizes operational costs by optimizing resource usage. As a result, VDIPS presents a cost-effective alternative that aligns with budget constraints while delivering superior security capabilities.

CHAPTER 6
SYSTEM DESIGN

6.1 OBJECTIVES

1. Infrastructure Setup
    - Objective: Establish the foundation for VDIPS.
    - Proxmox Virtualization: Set up Proxmox as the virtualization platform to create and manage virtual machines (VMs) and containers. Configure network interfaces and storage resources.

2. Security Component Deployment
    - Objective: Deploy essential security components within the virtualized environment.
    - Wazuh Integration: Install Wazuh agents on monitored endpoints to collect security data. Configure the Wazuh manager to process and analyze incoming data. Set up Elasticsearch for data storage and querying.
    - Suricata Implementation: Deploy Suricata sensors at strategic points within the network infrastructure to monitor traffic. Configure rule-based detection and integrate Suricata alerts with Wazuh for comprehensive threat detection.
    - Zeek Deployment: Implement Zeek for passive network analysis. Configure protocol analyzers to identify and extract relevant information from network traffic.
    - OWLH with ELK Stack: Set up OWLH as the real-time monitoring and analysis component. Integrate OWLH with the ELK stack for centralized log management, normalization, and visualization.

3. Threat Intelligence Integration

- Objective: Enhance threat awareness through real-time threat intelligence.
- Threat Intelligence Feeds: Integrate real-time threat intelligence feeds into Wazuh, Suricata, and Zeek. Ensure continuous updates for known threats, vulnerabilities, and attack patterns.

4. Real-Time Monitoring and Analysis
   - Objective: Enable real-time threat detection and response.
   - Continuous Monitoring: Activate OWLH to monitor network traffic, security events, and logs in real-time.
   - Alerting and Incident Response: Configure Wazuh and Suricata to generate alerts for detected threats. Implement incident response procedures to mitigate security incidents swiftly.

5. Resource Optimization and Scalability
   - Objective: Ensure efficient resource utilization and scalability.
   - Resource Allocation: Dynamically allocate virtual resources within Proxmox to optimize resource usage and minimize operational costs.
   - Scalability: Implement mechanisms to scale VDIPS in real-time, accommodating changes in workloads and emerging threats effortlessly.

6. Centralized Log Management and Visualization
   - Objective: Streamline log management and facilitate comprehensive threat analysis.

- ○ ELK Stack Utilization: Leverage the ELK stack (Elasticsearch, Logstash, Kibana) for centralized log storage, normalization, and visualization.
- ○ Visual Insights: Utilize Kibana's intuitive dashboards to gain visual insights into security events, trends, and network activity.

7. Testing and Evaluation
    - ○ Objective: Validate the functionality and effectiveness of VDIPS.
    - ○ Testing Scenarios: Conduct controlled testing scenarios to evaluate VDIPS's ability to detect known and emerging threats in real-time.
    - ○ Performance Evaluation: Measure the performance of the virtualized environment, including resource usage and scalability.

8. Documentation and Training
    - ○ Objective: Create comprehensive documentation and provide training for administrators.
    - ○ Documentation: Document the VDIPS setup, configuration, and maintenance procedures for future reference.
    - ○ Administrator Training: Train administrators and security personnel on VDIPS operation, monitoring, and incident response.

9. Deployment in Production Environment
    - ○ Objective: Transition VDIPS into the production environment.
    - ○ Production Deployment: Migrate the VDIPS solution into the organization's production environment while ensuring minimal disruption to ongoing operations.

10. Ongoing Monitoring and Maintenance
    - ○ Objective: Ensure the continued effectiveness and security of VDIPS.
    - ○ Monitoring and Updates: Continuously monitor VDIPS for security events, apply updates, and incorporate the latest threat intelligence feeds.
    - ○ Incident Response: Implement ongoing incident response procedures to address security incidents as they occur.

6.2 PROXMOX VIRTUALIZATION

System Architecture

Proxmox Virtualization serves as the foundation for the Virtualized Deployment of Intrusion Prevention Systems (VDIPS). Proxmox provides a hypervisor-based virtualization platform that combines two virtualization technologies: KVM (Kernel-based Virtual Machine) for full virtualization and LXC (Linux Containers) for lightweight container virtualization. This architecture allows for the creation and management of virtual machines (VMs) and containers within a single integrated environment.

Technical Explanation

- Proxmox leverages KVM to run VMs, providing near-native performance for guest operating systems. This is essential for hosting security components like Wazuh, Suricata, Zeek, and OWLH.
- LXC-based containers offer a lightweight alternative for certain security tools, enhancing resource efficiency and enabling rapid deployment of security components.

- Proxmox's web-based management interface simplifies VM and container management, making it accessible for administrators with varying levels of expertise.
- The ability to allocate resources dynamically ensures that VDIPS can scale instantly to meet the demands of changing workloads and emerging threats.

6.3 WAZUH

System Architecture

Wazuh is an open-source security monitoring platform that plays a crucial role in VDIPS. Its architecture consists of agents installed on monitored endpoints, a manager for log analysis, and an Elasticsearch database for storing and querying security data.

Technical Explanation

- Wazuh agents are deployed on endpoints (servers, workstations, etc.) to collect and forward security-related data, including logs and configuration information, to a central manager.
- The manager, in conjunction with the Elasticsearch database, processes and analyzes the incoming data in real-time. It applies predefined rules and decodes logs, identifying security incidents and anomalies.
- Wazuh offers alerting capabilities, providing immediate notifications for detected threats. These alerts can be integrated with other components like Suricata for enhanced threat response.

6.4 SURICATA

System Architecture

Suricata is a high-performance Network IDS, IPS, and Network Security Monitoring (NSM) engine. Its architecture revolves around the deployment of sensors that monitor network traffic for signs of intrusions or malicious activities.

Technical Explanation

- Suricata sensors are strategically placed within the network infrastructure to monitor incoming and outgoing traffic.
- Suricata's rule-based detection engine examines network packets in real-time, identifying patterns that match known attack signatures or suspicious behaviors.
- Upon detection, Suricata generates alerts and logs, providing detailed information about the detected threats. These alerts can be forwarded to a central log management system for analysis.

6.5 ZEEK

System Architecture

Zeek, formerly known as Bro, is a powerful network analysis framework used for traffic analysis, protocol detection, and network forensics. Its architecture involves passive network monitoring and analysis.

Technical Explanation:

- Zeek passively analyzes network traffic by sniffing packets on network segments or interfaces, capturing data without interfering with network operations.

- It employs a scripting language to define protocol analyzers, enabling the identification of network protocols and extracting relevant information from network traffic.
- Zeek generates logs that contain detailed network activity information, providing visibility into network communications, DNS resolutions, HTTP transactions, and more.

6.6 OWLH WITH ELK STACK

System Architecture

OWLH (OpenWareLab Hardware) is a cybersecurity platform designed for security information and event management (SIEM) and network traffic analysis. It is often complemented by the ELK stack (Elasticsearch, Logstash, Kibana) for log management and visualization.

Technical Explanation

- OWLH serves as the real-time monitoring and analysis component in VDIPS. It captures and processes network traffic data and security events.
- The ELK stack provides a centralized log management solution. Elasticsearch stores logs, Logstash normalizes and forwards them, and Kibana offers visualization and analysis capabilities.
- Together, OWLH and the ELK stack create a unified platform for storing, analyzing, and visualizing security logs and network traffic data. This centralized approach simplifies incident detection, investigation, and reporting.

6.7 SYSTEM ARCHITECTURE

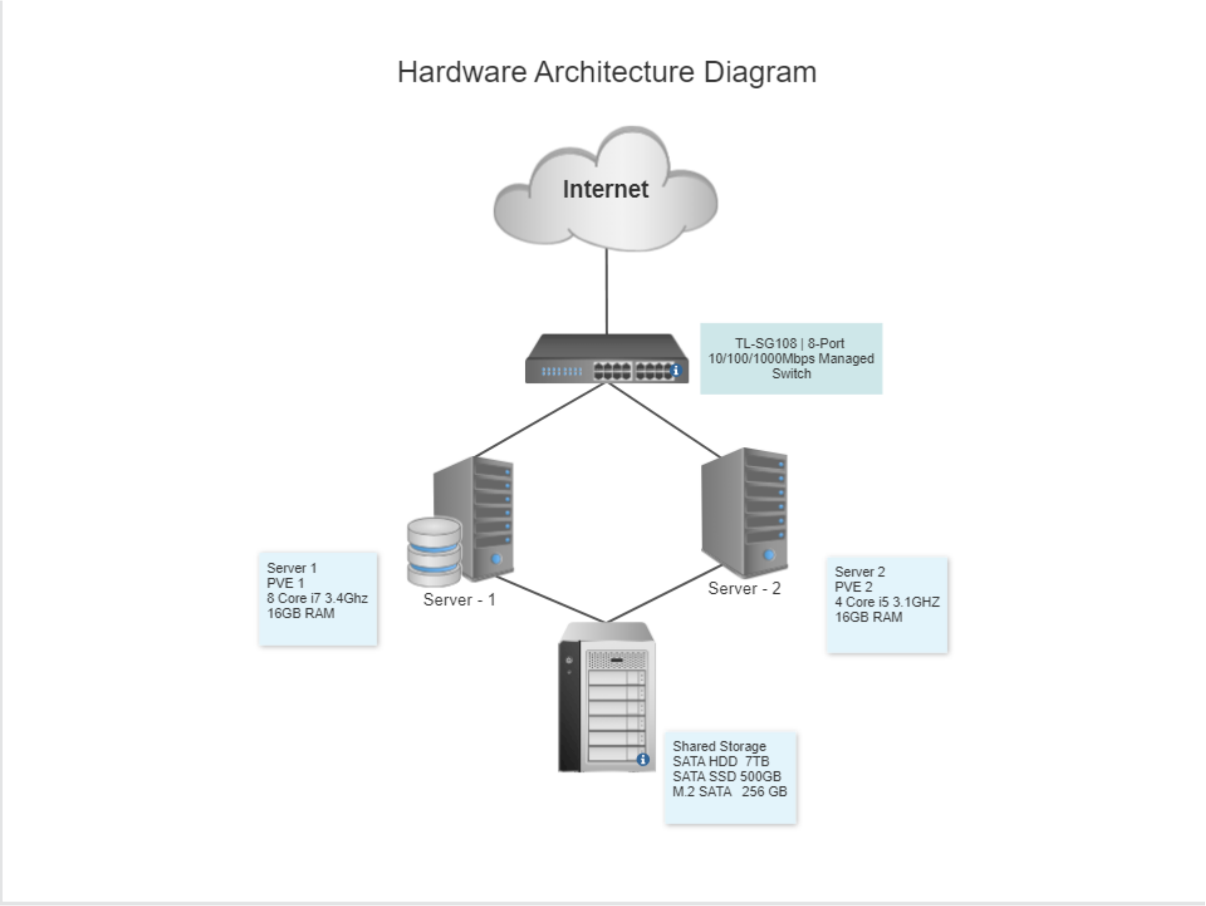


Figure 6.1: The Diagrammatic Representation of the Physical Infrastructure of the System

6.8 VIRTUAL IDS/IPS COMPONENTS

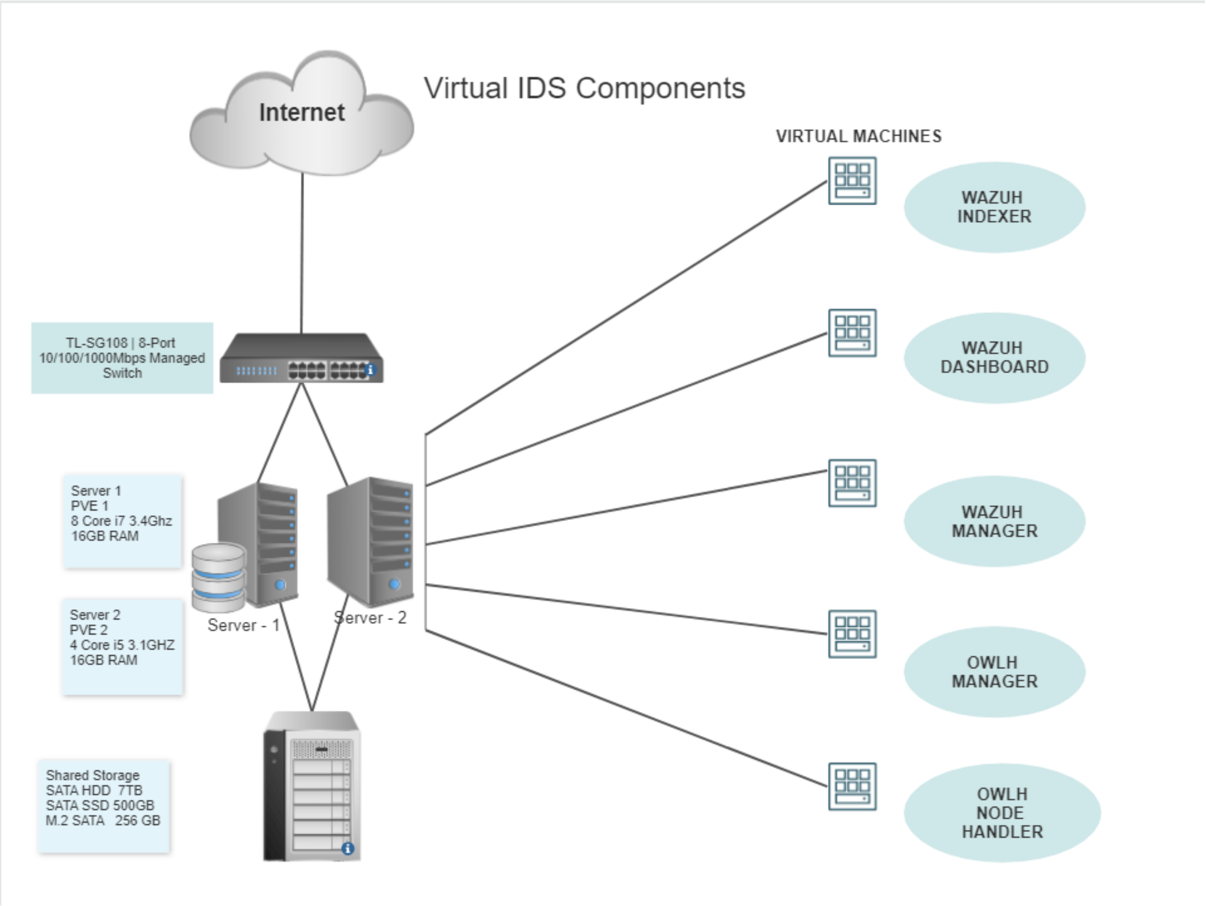


Figure 6.2: The Various Virtual Components

6.9 SERVICE CONNECTORS

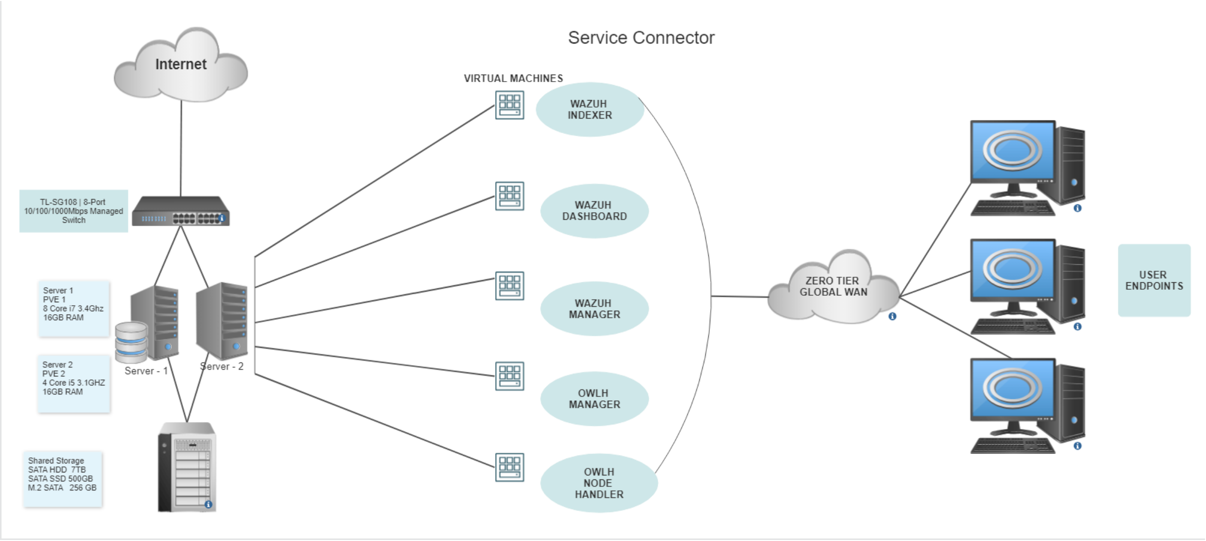


Figure 6.3: The Virtual Components connected via VPC

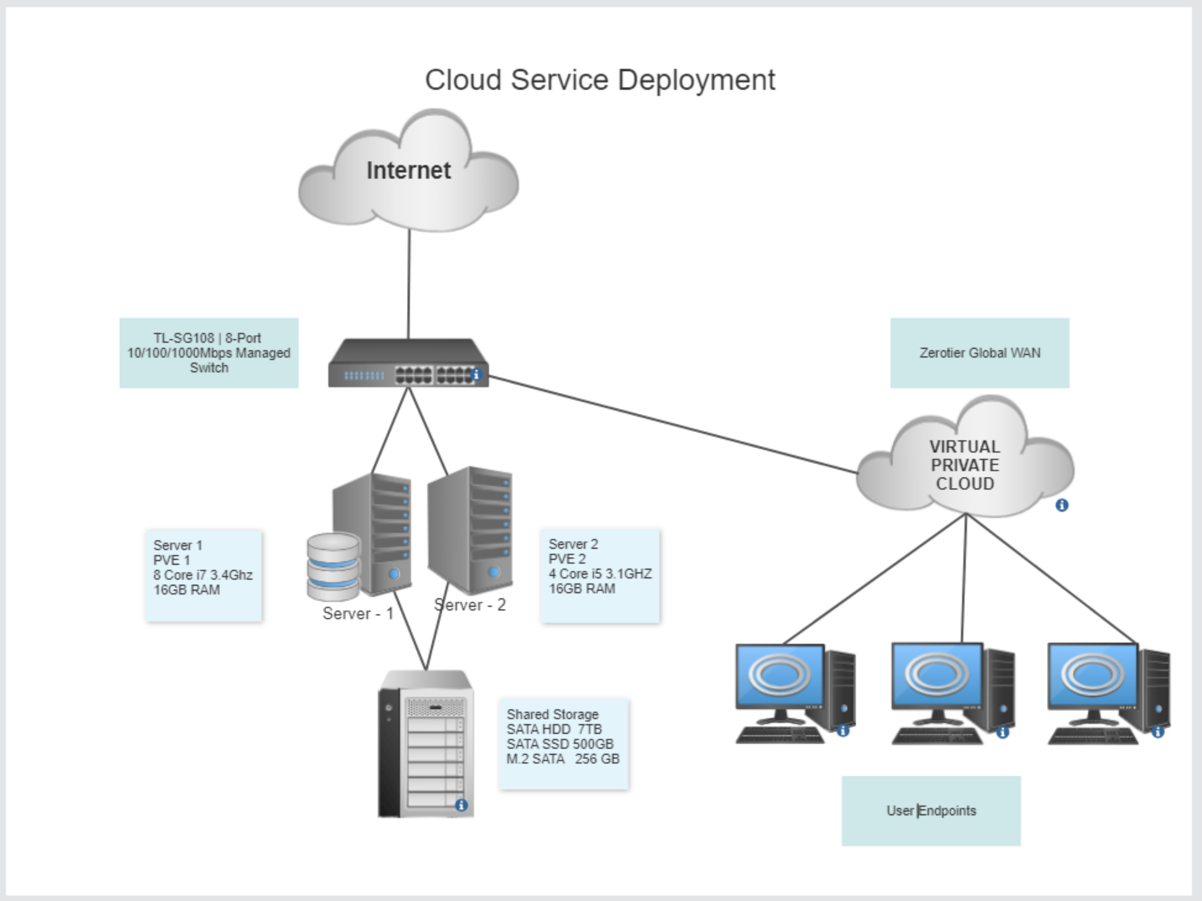


Figure 6.4: VPC for Endpoint traffic collection and management facilitated from remote location

CHAPTER 7
OVERVIEW OF TECHNIQUES

7.1 INFRASTRUCTURE SETUP AND CONTAINERIZATION - PHASE 1

7.1.1 PROXMOX INSTALLATION AND CONFIGURATION

In the initial phase, we lay the foundation for the Virtualized Hybrid Intrusion Detection and Prevention System (VDIPS). The first two weeks are dedicated to the installation and configuration of Proxmox, our virtualization platform.

- Objective: Set up Proxmox to create and manage virtual machines (VMs) and containers.
- Tasks:
  - Install Proxmox on the designated host system.
  - Configure network interfaces and storage resources.
  - Verify hardware compatibility and virtualization support.

7.1.2 LXC CONTAINERIZATION FOR SECURITY MODULES

With Proxmox in place, the next step is to create Linux Containers (LXC) to house the security modules: Wazuh, Zeek, Suricata, and OWLH.

- Objective: Prepare the infrastructure by creating the necessary containers for upcoming security module installations.
- Tasks:
  - Create individual LXC containers for Wazuh, Zeek, Suricata, and OWLH.
  - Assign resources and networking parameters for each container.
  - Validate connectivity between containers and the Proxmox host.

7.2 INSTALLATION AND IMPLEMENTATION - PHASE 2

7.2.1 INSTALLATION OF WAZUH

In the second phase, we delve into the installation and implementation of each security module. We start with Wazuh.

- Objective: Deploy Wazuh as the foundational security monitoring platform.
- Tasks:
  - Install Wazuh manager and agents within the designated LXC containers.
  - Configure manager-agent communication and perform initial setup.
  - Set up Elasticsearch as the data store for Wazuh.

7.2.2 IMPLEMENTATION OF SURICATA

Next in line is the implementation of Suricata, our high-performance Network IDS, IPS, and NSM engine.

- Objective: Deploy Suricata sensors for network traffic analysis.
- Tasks:
  - Install and configure Suricata sensors within their respective LXC containers.
  - Define and fine-tune detection rules for known threats.
  - Integrate Suricata alerts with the Wazuh manager for centralized threat monitoring.

7.2.3 INTEGRATION OF ZEEK

Following Suricata, we proceed with the implementation of Zeek, the network analysis framework.

- Objective: Deploy Zeek for passive network monitoring and analysis.
- Tasks:
  - Install Zeek within its designated LXC container.
  - Define protocol analyzers and configurations for network traffic analysis.
  - Set up log forwarding to the central log management system.

7.2.4 REAL-TIME MONITORING WITH OWLH

The final component of the implementation phase is OWLH, our real-time monitoring and analysis platform, complemented by the ELK stack (Elasticsearch, Logstash, Kibana).

- Objective: Activate OWLH for real-time monitoring and visualization.
- Tasks:
  - Configure OWLH to capture and process network traffic data and security events.
  - Set up the ELK stack for centralized log management, normalization, and visualization.
  - Validate the integration between OWLH and the ELK stack.

7.3 INTEGRATION OF ZEROTIER VPC FOR REMOTE DEVICE PROTECTION - PHASE 2

As part of the VDIPS ecosystem, the integration of ZeroTier VPC provides a secure and efficient means to connect and protect remote devices. ZeroTier acts as a Virtual Private Cloud, extending the reach of VDIPS to remote environments. This section outlines the objectives and tasks for incorporating ZeroTier VPC into the VDIPS infrastructure.

1. Establish a secure and seamless connection between remote devices and the VDIPS network, ensuring remote device protection using the proposed system.

2. Enable real-time monitoring and threat detection for remote devices, enhancing overall cybersecurity coverage.

7.3.1 ZEROTIER INSTALLATION AND CONFIGURATION

- Objective: Set up ZeroTier on the VDIPS infrastructure and remote devices.
- Tasks:
  - Install ZeroTier on the Proxmox host system.
  - Create a ZeroTier network and obtain network ID.
  - Install ZeroTier client software on remote devices.
  - Join remote devices to the ZeroTier network using the network ID.

7.3.2 INTEGRATION WITH OWLH AND ELK STACK

- Objective: Ensure remote device logs and network traffic data are integrated into the OWLH and ELK Stack for centralized monitoring and analysis.

- Tasks:
  - Configure OWLH to capture and process data from ZeroTier-connected devices.
  - Set up log forwarding from ZeroTier to the ELK stack.
  - Verify data flow and integration between remote devices, ZeroTier, OWLH, and the ELK stack.

7.3.3 REAL-TIME THREAT DETECTION AND ALERTING

- Objective: Implement real-time threat detection and alerting for remote devices using Wazuh and Suricata.
- Tasks:
  - Install and configure Wazuh agents on remote devices connected through ZeroTier.
  - Set up Suricata sensors for network traffic analysis of ZeroTier traffic.
  - Define and fine-tune detection rules for remote devices.
  - Integrate remote device alerts with the central Wazuh manager.

7.3.4 SCALABILITY AND ACCESS CONTROL

- Objective: Ensure scalability and access control within the ZeroTier network.
- Tasks:
  - Define access policies and permissions for remote devices.
  - Implement scalability measures to accommodate additional remote devices.
  - Test access control and scalability with new devices.

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

The Virtualized Hybrid Intrusion Detection and Prevention System (VDIPS) project, powered by an amalgamation of cutting-edge technologies and proactive strategies, stands as a formidable sentinel in the face of relentless and evolving cyber threats.

VDIPS derives its strength from a meticulously crafted system architecture, seamlessly integrating virtualization technologies, including virtual machines (VMs) and containers, within the Proxmox framework. This hybrid model empowers the system with dual threat detection mechanisms – signature-based and behavior-based, and augments it with real-time threat intelligence sourced from reputable channels. The vigilant eyes and ears of VDIPS lie in its real-time monitoring, driven by Security Information and Event Management (SIEM) systems and network traffic analysis (NTA) tools. This vigilant sentinel scrutinizes system and network activities, promptly alerting to anomalies, thus reducing dwell time and mitigating potential damage.

One of VDIPS's significant achievements lies in its scalability and adaptability. The virtualized architecture seamlessly adapts to organizational growth and technological advancements, ensuring that cybersecurity remains agile and future-ready.

Looking ahead, the VDIPS project presents ample opportunities for further refinement and expansion such as:

1. Advanced Machine Learning Integration : Incorporating machine learning algorithms can enhance VDIPS's anomaly detection capabilities. The system can learn from historical data and adapt its threat detection models in real-time, staying ahead of evolving attack vectors.

2. Automated Incident Response : Integrating automated incident response mechanisms can enable VDIPS to not only detect but also autonomously mitigate certain threats, reducing response times and human intervention.

3. Enhanced User Interface and Reporting : Improving the user interface and reporting capabilities, possibly through the utilization of advanced visualization tools, can provide security teams with more intuitive and actionable insights.

4. Cloud Integration : Extending VDIPS to seamlessly integrate with cloud environments can further enhance its scalability and adaptability, allowing it to protect assets across various infrastructures.

5. Threat Intelligence Enrichment : Enhancing the integration of real-time threat intelligence by diversifying sources and employing threat feeds with machine-readable formats can augment VDIPS's threat awareness.

As the cybersecurity landscape continues to evolve, VDIPS remains poised to evolve with it. With ongoing innovation and a commitment to proactive cybersecurity strategies, VDIPS will stand as a steadfast guardian, equipping organizations to navigate the digital future with confidence and resilience.

APPENDIX

A1 - SOURCE CODE

Install Procedure of proxmox in bare metal servers:

Download Proxmox VE ISO:

1. Download the Proxmox VE ISO image from the official website. You can use a web browser or command-line tools like wget to download the ISO. Replace <ISO_URL> with the actual download link:

   ○ **wget <ISO_URL>**

Create a Bootable USB Drive:

2. You can create a bootable USB drive using tools like dd. Make sure to replace <USB_DEVICE> with the path to your USB device (e.g., /dev/sdX). Be extremely cautious when using dd, as it can overwrite data if misused:

```
sudo dd if=proxmox-ve.iso of=/dev/<USB_DEVICE> bs=4M
status=progress
```

Boot from the USB Drive:

3. Insert the bootable USB drive into the server, then boot from it. You may need to adjust the BIOS/UEFI settings to prioritize booting from the USB drive.

Proxmox VE Installation:

4. Once the server boots from the USB drive, the Proxmox VE installer will start. Follow these steps:

a. Select "Install Proxmox VE" and press Enter.
b. Choose the target disk for installation. You can use the command below to list available disks:
   - **lsblk**
c. Then select the appropriate disk (e.g., /dev/sda) for the installation.
d. Confirm the installation by typing "YES" (all uppercase) when prompted.
e. Choose your country, timezone, and keyboard layout.
f. Set a strong password for the root user.
g. Configure the network settings, including the IP address, gateway, and DNS.
h. Select whether to use the Proxmox VE subscription repository. If you don't have a subscription, you can choose "No Subscription."
i. Review the installation summary and confirm the installation.

Completing the Installation:

5. The installation will take some time to complete. Once finished, remove the USB drive and reboot the server. Proxmox VE should now be installed on the bare-metal server.

Accessing the Web Interface:

6. After rebooting, you can access the Proxmox VE web interface by opening a web browser and entering the server's IP address with port 8006 (https://<Server_IP>:8006/). Log in using the root username and the password you set during installation.

Installation of the VPC in Endpoint devices :

1. Install cli

   a.
   ```
   $ curl -s 'https://raw.githubusercontent.com/zerotier/ZeroTierOne/master/doc/contact%40zerotier.com.gpg' | gpg --import && \ if z=$(curl -s 'https://install.zerotier.com/' | gpg); then echo "$z" | sudo bash; fi
   ```

2. Start service

   b.
   ```
   $ sudo service zerotier-one restart
   ```

3. Join the network

   c.
   ```
   $ sudo zerotier-one.zerotier-cli join <network id>
   ```

Creating hardware-specific LXC containers in Proxmox:

1. Log In to the Proxmox Web Interface:
    a. Open a web browser and access the Proxmox web interface by entering the server's IP address with port 8006 (https://<Server_IP>:8006/). Log in using your Proxmox credentials.

2. Create a New LXC Container:
    b. In the Proxmox web interface, navigate to the desired storage location (e.g., "local") and click "Content."
    c. Click the "Create CT" (Container) button.
    d. Fill in the required information, such as the container name (e.g., "my-container"), root password, and networking settings.
    e. Under "OS," choose the Linux distribution and version you want to install in the container. You can specify the storage location for the container's root file system.
    f. Click "Create" to create the LXC container.

3. Customize Hardware Resources:
    - ○ By default, Proxmox assigns default hardware resources to the container. To customize these resources, follow these steps:
        a. Select the newly created container in the Proxmox web interface.
        b. Go to the "Hardware" tab.
        c. Click "Add" to add hardware resources to the container. You can add the following hardware resources:

This is the Allocation of resources.

I. CPU: Allocate specific CPU cores or set CPU limits.

```
# Assign specific CPU cores to the container
pct set <CTID> -cpus 1,2

# Set CPU limits (e.g., 50%)
pct set <CTID> -cpuunits 512
```

II. Memory (RAM): Set the amount of RAM for the container.

```
pct set <CTID> -memory 4G
```

III. Network: Configure network settings for the container.

```
pct set <CTID> -net0 name=eth0,bridge=vmbr0,ip=<IP_ADDRESS>/24,gw=<GATEWAY_IP>
```

IV. Storage: Attach additional storage devices or directories to the container.

```
pct set <CTID> -mp0 /host/path,mp=/container/path
```

V. Start the LXC Container:

After configuring hardware resources and settings, start the LXC container:

```
pct start <CTID>
```

VI. Access and Configure the LXC Container:

To access and configure the container, you can use the Proxmox web console or an SSH client, depending on the distribution you installed. You can log in with the root username and the password you specified during the container creation process.

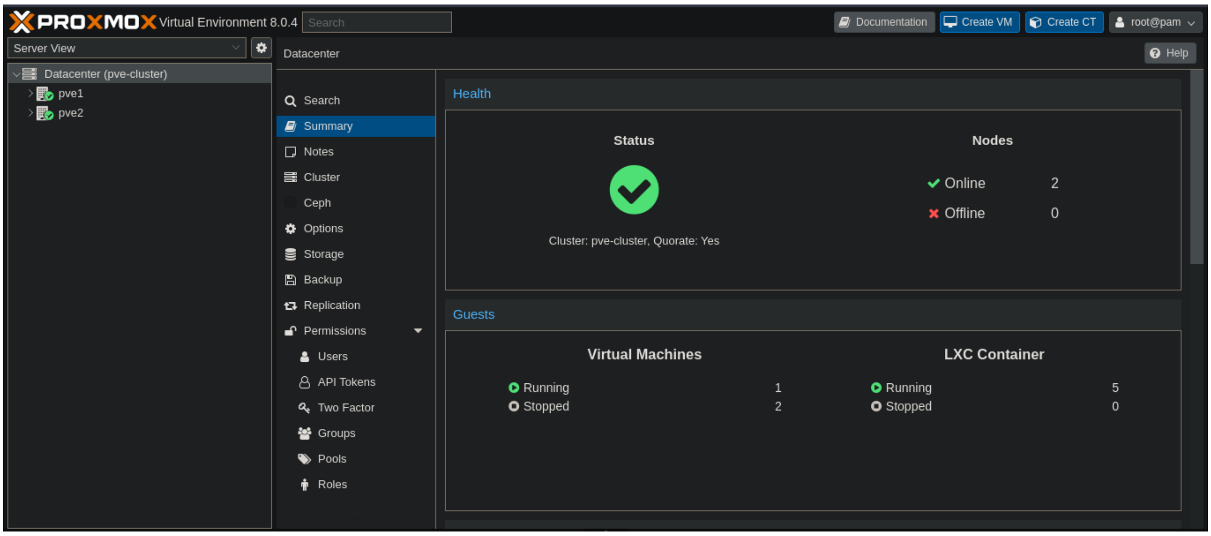
VII. Install and Customize Software:
Inside the container, you can install and customize the software and services you need for your specific use case.

VIII. Save Changes:
After making customizations to the LXC container, it's essential to save any changes you've made. If you want to create a template from this container for future use, you can use the "Convert to Template" option in the Proxmox web interface.

APPENDIX

A2 – SCREENSHOTS

PROXMOX DASHBOARDS



Health

Status

Cluster: pve-cluster, Quorate: Yes

Nodes

✓ Online	2
✗ Offline	0

Guests

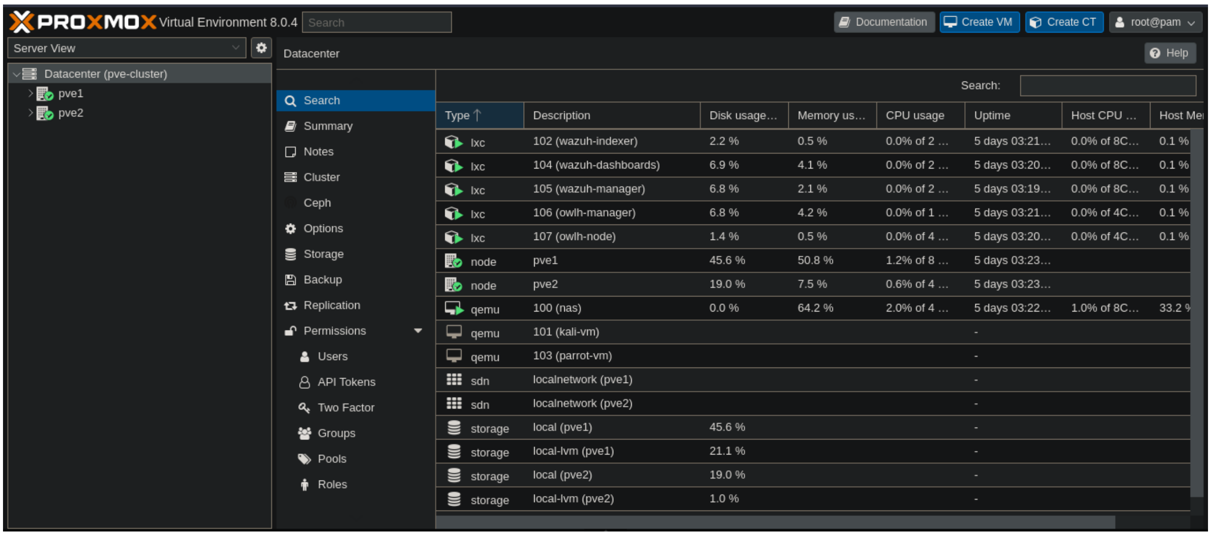
Virtual Machines

Running	1
Stopped	2

LXC Container

Running	5
Stopped	0

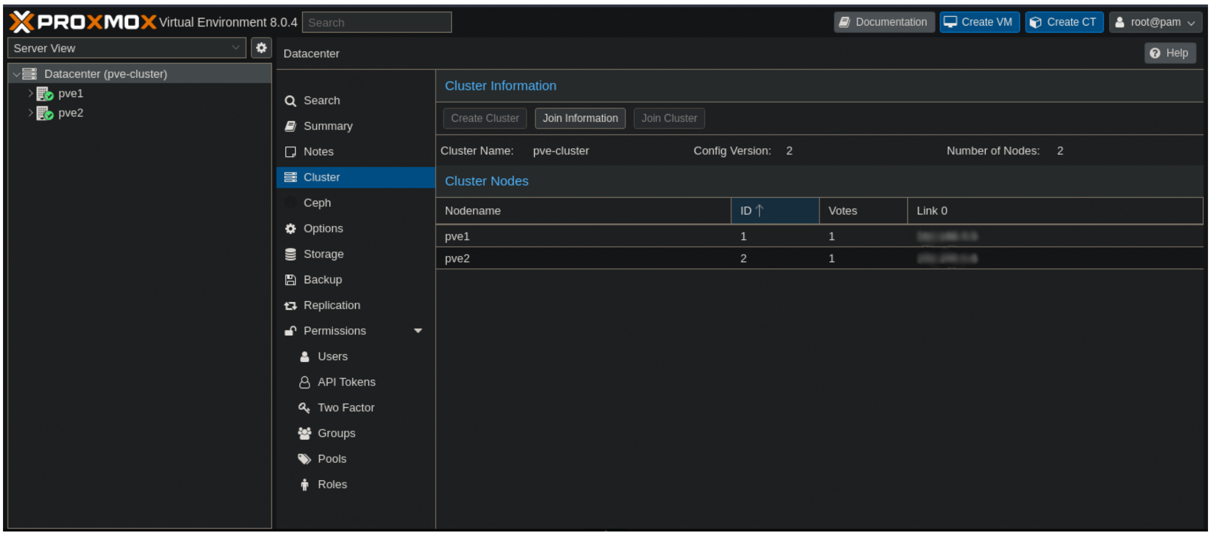
Figure A2.1 : The Summary page of proxmox dashboard



Search:

Type ↑	Description	Disk usage…	Memory us…	CPU usage	Uptime	Host CPU …	Host Me
lxc	102 (wazuh-indexer)	2.2 %	0.5 %	0.0% of 2 …	5 days 03:21…	0.0% of 8C…	0.1 %
lxc	104 (wazuh-dashboards)	6.9 %	4.1 %	0.0% of 2 …	5 days 03:20…	0.0% of 8C…	0.1 %
lxc	105 (wazuh-manager)	6.8 %	2.1 %	0.0% of 2 …	5 days 03:19…	0.0% of 8C…	0.1 %
lxc	106 (owlh-manager)	6.8 %	4.2 %	0.0% of 1 …	5 days 03:21…	0.0% of 4C…	0.1 %
lxc	107 (owlh-node)	1.4 %	0.5 %	0.0% of 4 …	5 days 03:20…	0.0% of 4C…	0.1 %
node	pve1	45.6 %	50.8 %	1.2% of 8 …	5 days 03:23…		
node	pve2	19.0 %	7.5 %	0.6% of 4 …	5 days 03:23…		
qemu	100 (nas)	0.0 %	64.2 %	2.0% of 4 …	5 days 03:22…	1.0% of 8C…	33.2 %
qemu	101 (kali-vm)				-		
qemu	103 (parrot-vm)				-		
sdn	localnetwork (pve1)				-		
sdn	localnetwork (pve2)				-		
storage	local (pve1)	45.6 %			-		
storage	local-lvm (pve1)	21.1 %			-		
storage	local (pve2)	19.0 %			-		
storage	local-lvm (pve2)	1.0 %			-		

Figure A2.2 : The Search page of dashboard which shows the list off all LXC Containers



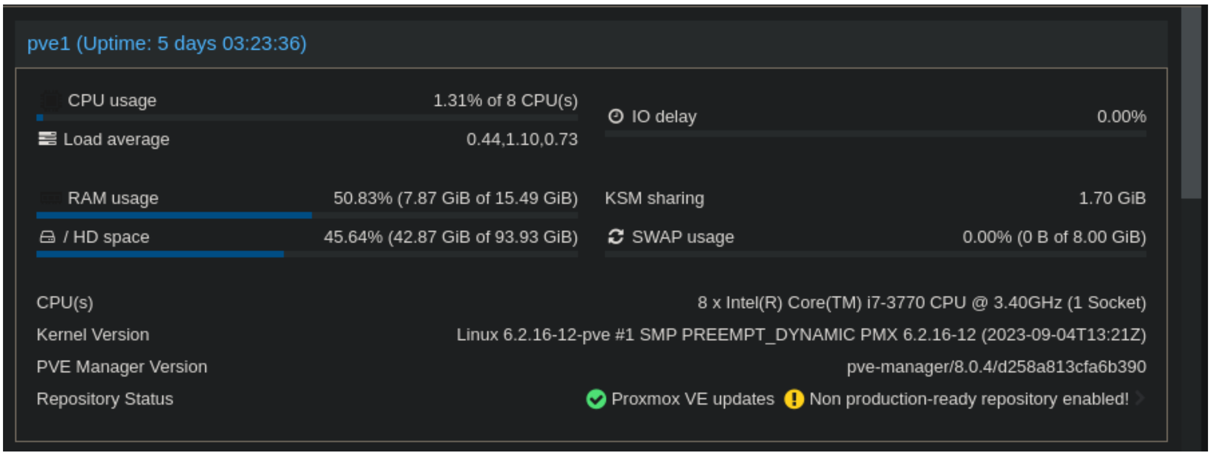
Cluster Information

Cluster Name: pve-cluster Config Version: 2 Number of Nodes: 2

Cluster Nodes

Nodename	ID ↑	Votes	Link 0
pve1	1	1	[redacted]
pve2	2	1	[redacted]

Figure A2.3 : The Node Quorum of both the proxmox hosts

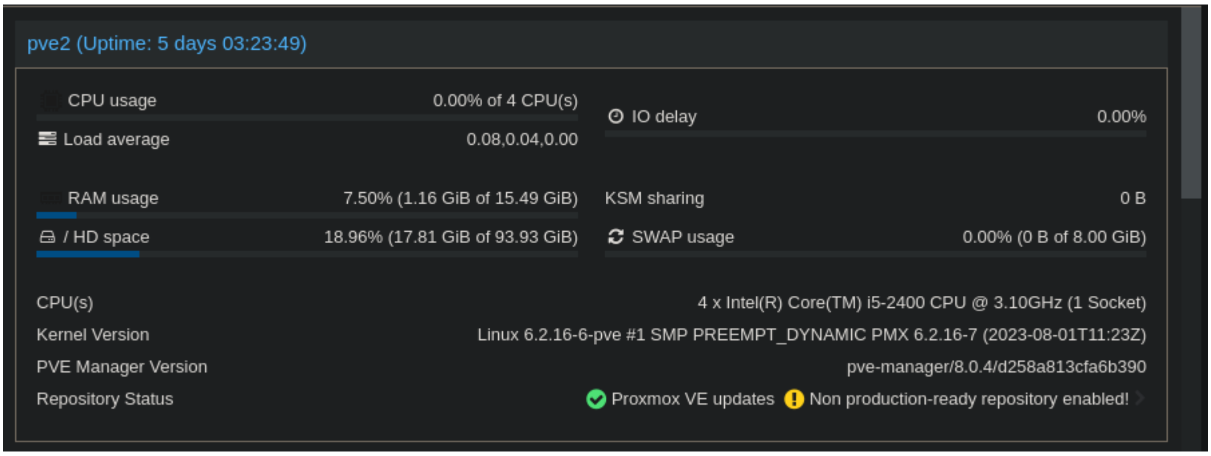


pve1 (Uptime: 5 days 03:23:36)

CPU usage	1.31% of 8 CPU(s)	IO delay	0.00%
Load average	0.44,1.10,0.73		
RAM usage	50.83% (7.87 GiB of 15.49 GiB)	KSM sharing	1.70 GiB
/ HD space	45.64% (42.87 GiB of 93.93 GiB)	SWAP usage	0.00% (0 B of 8.00 GiB)

CPU(s)	8 x Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz (1 Socket)
Kernel Version	Linux 6.2.16-12-pve #1 SMP PREEMPT_DYNAMIC PMX 6.2.16-12 (2023-09-04T13:21Z)
PVE Manager Version	pve-manager/8.0.4/d258a813cfa6b390
Repository Status	✓ Proxmox VE updates ! Non production-ready repository enabled!

Figure A2.4 : The Hardware details of Proxmox Host 1



pve2 (Uptime: 5 days 03:23:49)

CPU usage 0.00% of 4 CPU(s)

Load average 0.08,0.04,0.00

IO delay 0.00%

RAM usage 7.50% (1.16 GiB of 15.49 GiB)

/ HD space 18.96% (17.81 GiB of 93.93 GiB)

KSM sharing 0 B

SWAP usage 0.00% (0 B of 8.00 GiB)

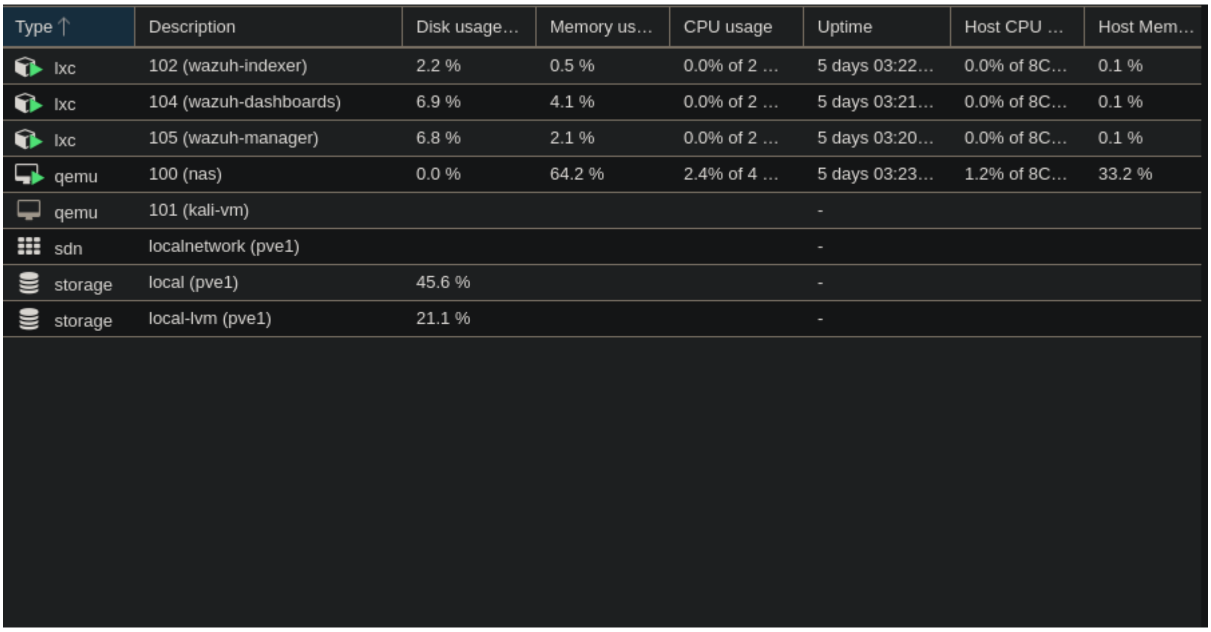
CPU(s) 4 x Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz (1 Socket)

Kernel Version Linux 6.2.16-6-pve #1 SMP PREEMPT_DYNAMIC PMX 6.2.16-7 (2023-08-01T11:23Z)

PVE Manager Version pve-manager/8.0.4/d258a813cfa6b390

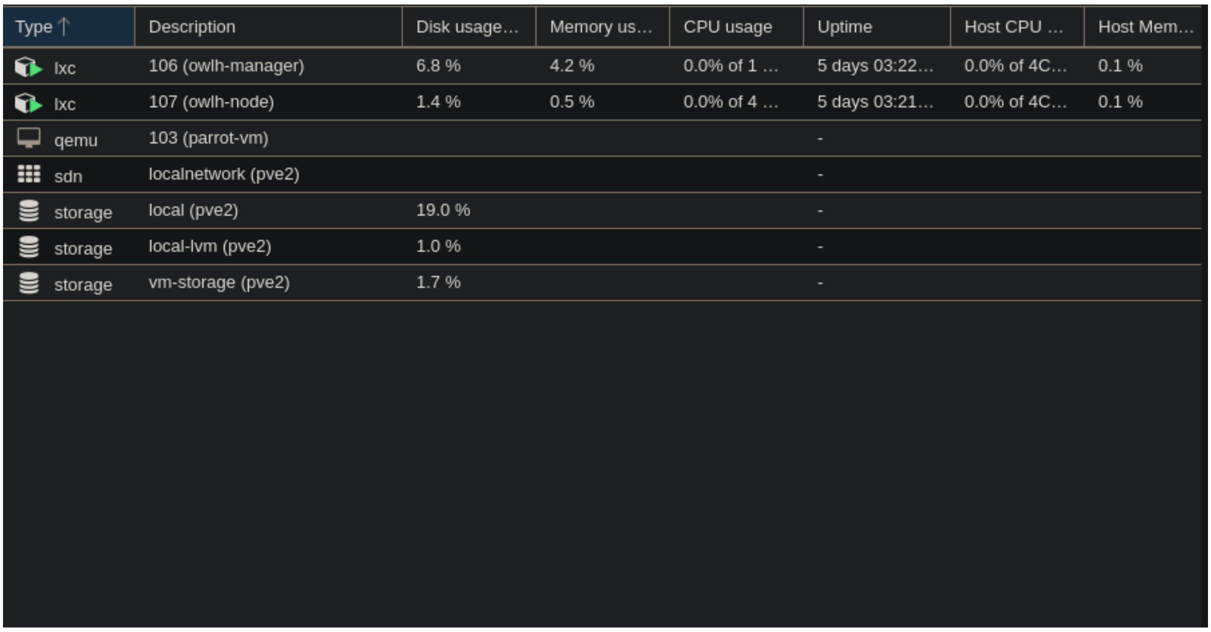
Repository Status ✓ Proxmox VE updates ! Non production-ready repository enabled!

Figure A2.5 : The Hardware details of Proxmox Host 2



Type ↑	Description	Disk usage…	Memory us…	CPU usage	Uptime	Host CPU …	Host Mem…
lxc	102 (wazuh-indexer)	2.2 %	0.5 %	0.0% of 2 …	5 days 03:22…	0.0% of 8C…	0.1 %
lxc	104 (wazuh-dashboards)	6.9 %	4.1 %	0.0% of 2 …	5 days 03:21…	0.0% of 8C…	0.1 %
lxc	105 (wazuh-manager)	6.8 %	2.1 %	0.0% of 2 …	5 days 03:20…	0.0% of 8C…	0.1 %
qemu	100 (nas)	0.0 %	64.2 %	2.4% of 4 …	5 days 03:23…	1.2% of 8C…	33.2 %
qemu	101 (kali-vm)				-		
sdn	localnetwork (pve1)				-		
storage	local (pve1)	45.6 %			-		
storage	local-lvm (pve1)	21.1 %			-		

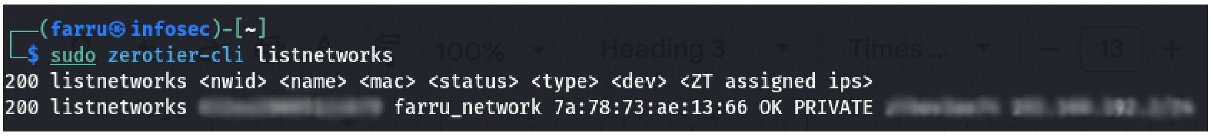
Figure A2.6 : List of LXC Containers hosted in Proxmox Host 1



Type ↑	Description	Disk usage…	Memory us…	CPU usage	Uptime	Host CPU …	Host Mem…
lxc	106 (owlh-manager)	6.8 %	4.2 %	0.0% of 1 …	5 days 03:22…	0.0% of 4C…	0.1 %
lxc	107 (owlh-node)	1.4 %	0.5 %	0.0% of 4 …	5 days 03:21…	0.0% of 4C…	0.1 %
qemu	103 (parrot-vm)				-		
sdn	localnetwork (pve2)				-		
storage	local (pve2)	19.0 %			-		
storage	local-lvm (pve2)	1.0 %			-		
storage	vm-storage (pve2)	1.7 %			-		

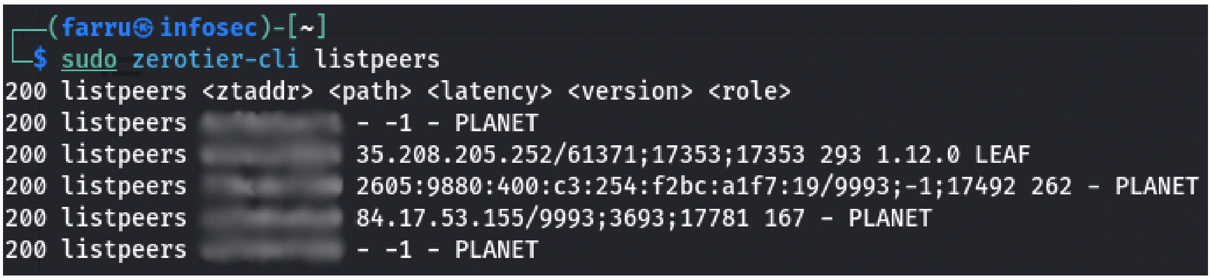
Figure A2.7 : List of LXC Containers hosted in Proxmox Host 2

ZEROTIER VPC



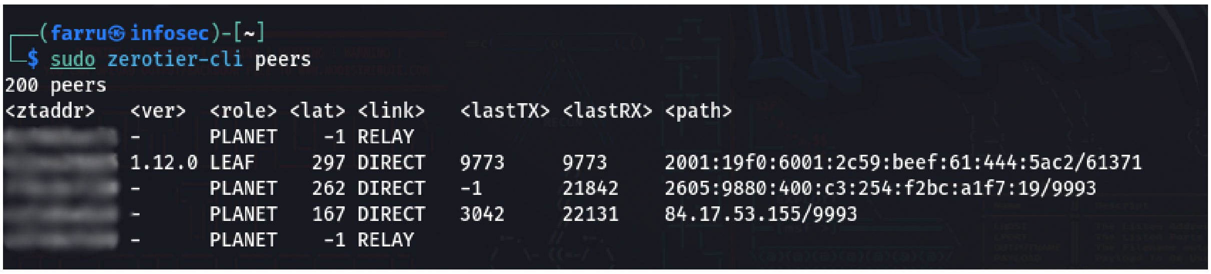
```
┌──(farru㉿infosec)-[~]
└─$ sudo zerotier-cli listnetworks
200 listnetworks <nwid> <name> <mac> <status> <type> <dev> <ZT assigned ips>
200 listnetworks [redacted] farru_network 7a:78:73:ae:13:66 OK PRIVATE [redacted] [redacted]
```

Figure A2.8: The endpoint connected to zerotier VPC Network



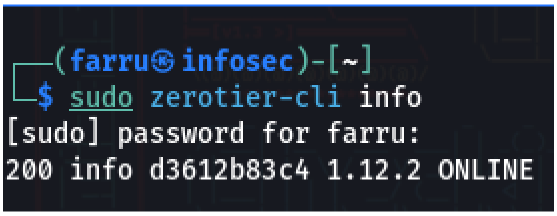
```
┌──(farru㉿infosec)-[~]
└─$ sudo zerotier-cli listpeers
200 listpeers <ztaddr> <path> <latency> <version> <role>
200 listpeers [redacted] - -1 - PLANET
200 listpeers [redacted] 35.208.205.252/61371;17353;17353 293 1.12.0 LEAF
200 listpeers [redacted] 2605:9880:400:c3:254:f2bc:a1f7:19/9993;-1;17492 262 - PLANET
200 listpeers [redacted] 84.17.53.155/9993;3693;17781 167 - PLANET
200 listpeers [redacted] - -1 - PLANET
```

Figure A2.9 : The listed peers and the devices address with network CIDR



```
┌──(farru㉿infosec)-[~]
└─$ sudo zerotier-cli peers
200 peers
<ztaddr>   <ver>  <role> <lat> <link>   <lastTX> <lastRX> <path>
[redacted] -      PLANET    -1 RELAY
[redacted] 1.12.0 LEAF     297 DIRECT   9773     9773     2001:19f0:6001:2c59:beef:61:444:5ac2/61371
[redacted] -      PLANET   262 DIRECT   -1       21842    2605:9880:400:c3:254:f2bc:a1f7:19/9993
[redacted] -      PLANET   167 DIRECT   3042     22131    84.17.53.155/9993
[redacted] -      PLANET    -1 RELAY
```

Figure A2.10 : The list of static peers



```
┌──(farru㉿infosec)-[~]
└─$ sudo zerotier-cli info
[sudo] password for farru:
200 info d3612b83c4 1.12.2 ONLINE
```

Figure A2.11 : Connection status VPC

REFERENCES

1. J. Li, "Network Intrusion Detection Algorithm and Simulation of Complex System in Internet Environment," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 520-523, doi: 10.1109/ICIRCA54612.2022.9985720.

2. A. S. Ayanboye, J. E. Efiong, G. E. Alilu, A. I. Oyebade and B. O. Akinyemi, "An Assessment of Security Techniques for Denial of Service Attack in Virtualized Environments," 2022 5th Information Technology for Education and Development (ITED), Abuja, Nigeria, 2022, pp. 1-7, doi: 10.1109/ITED56637.2022.10051209.

3. M. Shamseddine, A. Al-Dulaimy, W. Itani, T. Nolte and A. V. Papadopoulos, "Nodeguard: A Virtualized Introspection Security Approach for the Modern Cloud Data Center," 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Taormina, Italy, 2022, pp. 790-797, doi: 10.1109/CCGrid54584.2022.00093.

4. A. A. E. Boukebous, M. I. Fettache, G. Bendiab and S. Shiaeles, "A Comparative Analysis of Snort 3 and Suricata," 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), London, United Kingdom, 2023, pp. 1-6, doi: 10.1109/GlobConET56651.2023.10150141.

5. J. -H. Lee, Y. S. Kim, J. H. Kim and I. K. Kim, "Toward the SIEM architecture for cloud-based security services," 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 2017, pp. 398-399, doi: 10.1109/CNS.2017.8228696.

6. K. Wong, C. Dillabaugh, N. Seddigh and B. Nandy, "Enhancing Suricata intrusion detection system for cyber security in SCADA networks," 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 2017, pp. 1-5, doi: 10.1109/CCECE.2017.7946818.

7. V. Mavroeidis and S. Bromander, "Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence," 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 2017, pp. 91-98, doi: 10.1109/EISIC.2017.20.

8. “Adding a Comprehensive Wazuh SIEM and Network Intrusion Detection System (NIDS) to the Proxmox Lab,” *0xBEN*, Jan. 17, 2022. [Online]. Available: https://benheater.com/proxmox-lab-wazuh-siem-and-nids/

9. “The Zeek Network Security Monitor,” *Zeek*. [Online]. Available: https://zeek.org/

10. “v0.18.x,” *v0.18.x*. [Online]. Available: https://www.owlh.net/