



**UNIVERSITY OF PIRAEUS - DEPARTMENT OF INFORMATICS**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**MSc «Distributed Systems, Security  
and Emerging Information Technologies»**

ΠΜΣ «Κατανεμημένα Συστήματα, Ασφάλεια  
και Αναδυόμενες Τεχνολογίες Πληροφορίας»

**MSc Thesis**

Μεταπτυχιακή Διατριβή

<b>Thesis Title:</b> Τίτλος Διατριβής:	<b>Building threat profiles based on model asset inventorying and automated penetration testing.</b>  Σχεδίαση προφίλ επιθέσεων βασισμένη σε μοντελοποίηση αγαθών και σε αυτοματοποιημένο έλεγχο εισβολών δικτύου.
<b>Student's name-surname:</b> Όνοματεπώνυμο φοιτητή:	<b>Lazaridis Christos</b> Λαζαρίδης Χρήστος
<b>Father's name:</b> Πατρώνυμο:	<b>Ioannis</b> Ιωάννης
<b>Student's ID No:</b> Αριθμός Μητρώου:	ΜΠΚΣΑ20013
<b>Supervisor:</b> Επιβλέπων:	<b>Panagiotis Kotzanikolaou, Associate Professor</b> Παναγιώτης Κοτζανικολάου, Αναπληρωτής καθηγητής

November 2023/ Νοέμβριος 2023



**3-Member Examination Committee**

Τριμελής Εξεταστική Επιτροπή

**Panagiotis Kotzanikolaou**  
**Associate Professor**

Παναγιώτης Κοτζανικολάου  
Αναπληρωτής Καθηγητής

**Constantinos Patsakis**  
**Associate Professor**

Κωνσταντίνος Πατσάκης  
Αναπληρωτής Καθηγητής

**Michael Psarakis**  
**Associate Professor**

Μιχαήλ Ψαράκης  
Αναπληρωτής Καθηγητής

## Περίληψη

Η τεχνολογία επεκτείνεται με ταχείς ρυθμούς, με τα σύγχρονα συστήματα να προσθέτουν διαρκώς IT, OT, IoT, edge και διάφορα στοιχεία για να ανταποκρίνονται στις απαιτήσεις της αγοράς. Αυτή η ανάπτυξη επιτρέπει νέες λειτουργίες και ευκολίες, αλλά θέτει επίσης σημαντικές προκλήσεις στον τομέα της κυβερνοασφάλειας. Η προστασία γνωστών και άγνωστων περιουσιακών στοιχείων από πολυάριθμες απειλές αποτελεί πλέον κορυφαία προτεραιότητα για τους παγκόσμιους οργανισμούς.

Η παρούσα διατριβή προτείνει μια ολοκληρωμένη προσέγγιση για την αντιμετώπιση των ζητημάτων κυβερνοασφάλειας με τη χρήση προηγμένων τεχνικών και τεχνολογιών. Επικεντρώνεται κυρίως στη δημιουργία κυβερνοπεριοχών και στην εφαρμογή τεχνικών για την καταγραφή περιουσιακών στοιχείων από συσκευές Windows και Linux, οργανώνοντάς τα σε μορφή Common Platform Enumeration (CPE) v2.3. Αυτή η μέθοδος θέτει τις βάσεις για την καινοτόμο προσέγγιση της δημιουργίας προφίλ απειλών.

Για να ενισχύσουμε την άμυνα της κυβερνοασφάλειάς μας, ενσωματώνουμε ευρέως χρησιμοποιούμενες λύσεις διαχείρισης συμβάντων πληροφοριών ασφαλείας (SIEM), όπως το Sentinel και το Wazuh. Αυτές οι πλατφόρμες προσφέρουν ζωτικής σημασίας πληροφορίες για την κατάσταση ασφαλείας των καταχωρημένων περιουσιακών στοιχείων. Επιπλέον, χρησιμοποιούμε αυτοματοποιημένες λύσεις δοκιμών διείσδυσης για την επαλήθευση των απειλών, χρησιμοποιώντας το πλαίσιο τεχνικών ATT&CK για τον εντοπισμό ευπαθειών και μεθόδων επίθεσης.

Στις επόμενες ενότητες περιγράφεται λεπτομερώς η ολοκληρωμένη προσέγγιση, διερευνώντας το ιστορικό, τις σχετικές εργασίες και παρουσιάζοντας μια προσαρμοσμένη μέθοδο σχεδιασμένη για το διαρκώς μεταβαλλόμενο τοπίο της κυβερνοασφάλειας. Η διατριβή μας αποσκοπεί στην προστασία των σύγχρονων συστημάτων γεφυρώνοντας το χάσμα μεταξύ γνωστών και άγνωστων περιουσιακών στοιχείων, δημιουργώντας μια στέρεη βάση για την ανάπτυξη βασικών προφίλ απειλών που είναι ζωτικής σημασίας στη σημερινή εποχή της ταχείας τεχνολογικής ανάπτυξης.

## **Abstract**

Technology is expanding rapidly, with modern systems constantly adding IT, OT, IoT, edge, and various components to meet market demands. This growth enables new functionalities and convenience but also poses significant cybersecurity challenges. Protecting known and unknown assets from numerous threats is now a top priority for global organizations.

This thesis proposes a comprehensive approach to tackle cybersecurity issues by using advanced techniques and technologies. It primarily focuses on creating cyber ranges and employing techniques to inventory assets from Windows and Linux devices, organizing them into a Common Platform Enumeration (CPE) format v2.3. This method lays the groundwork for our innovative threat profiling approach.

To enhance our cybersecurity defense, we incorporate widely used Security Information Event Management (SIEM) solutions like Sentinel and Wazuh. These platforms offer vital insights into the security status of listed assets. Additionally, we utilize automated penetration testing solutions to verify threats, employing the ATT&CK technique framework to identify vulnerabilities and attack methods.

The following sections detail the comprehensive approach, exploring background information, related work, and introducing a customized method designed for the ever-changing cybersecurity landscape. Our thesis aims to protect modern systems by bridging the gap between known and unknown assets, creating a solid foundation for developing essential threat profiles crucial in the current era of rapid technological growth.

## Contents

Περίληψη .....	4
Abstract.....	5
<b>1 Introduction .....</b>	<b>10</b>
<b>1.1 Current state of IT/OT Infrastructure .....</b>	<b>10</b>
1.1.1 Cloud Infrastructure .....	10
1.1.2 On-Premises IT Infrastructure .....	12
1.1.3 IOT/OT Infrastructure.....	14
<b>1.2 Motivation .....</b>	<b>15</b>
1.2.1 Asset Inventory management.....	15
1.2.2 Network Asset Management.....	16
1.2.3 User Access and Privilege Management.....	18
<b>1.3 Contribution.....</b>	<b>19</b>
1.3.1 Cyber Range Development .....	19
1.3.2 Modular Asset Modeling Framework .....	19
1.3.3 Automated CPE Management Tool .....	20
1.3.4 Security Control Integration.....	20
<b>2 Related Work.....</b>	<b>21</b>
<b>2.1 Frameworks, Ontologies and Databases .....</b>	<b>21</b>
2.1.1 Common Platform Enumeration (CPE) .....	22
2.1.2 Common Weakness Enumeration (CWE) .....	22
2.1.3 Common Attack Pattern Enumeration and Classification (CAPEC) .....	22
2.1.4 Common Vulnerabilities and Exposures (CVE) .....	23
2.1.5 Common Vulnerability Scoring System (CVSS).....	23
2.1.6 Exploit Prediction Scoring System (EPSS).....	25
<b>2.2 Asset Inventory &amp; Vulnerability Management tools .....</b>	<b>26</b>
2.2.1 Asset Inventory Management tools.....	26
2.2.2 Network Asset Management tools .....	27
2.2.3 User Access and Privilege Management tools .....	28
2.2.4 Vulnerability Management tools.....	29
<b>2.3 Related research .....</b>	<b>30</b>
<b>3 Methodology – Architecture .....</b>	<b>34</b>
<b>3.1 Cyber Range Development .....</b>	<b>34</b>
3.1.1 Active Directory Environment with IIS & SQL Server .....	35
3.1.2 Firewall with IDS capabilities.....	35
3.1.3 `Windows and Linux Client.....	36
3.1.4 SIEM Technologies.....	36
<b>3.2 Modular Asset Modeling &amp; CPE Extraction .....</b>	<b>37</b>
3.2.1 Why Harvesting Assets in CPE Format is Crucial.....	37
3.2.2 Defining Device Layers .....	38
3.2.3 Enumerating Assets Within Each Layer .....	39
3.2.4 Creating CPEs from Extracted Data .....	40
<b>3.3 Security Control Integration .....</b>	<b>41</b>
3.3.1 Automated Penetration Testing Agents .....	41
3.3.2 Exploitable Threats in MITRE ATTACK Format .....	42
3.3.3 Building Custom Threat Profiles .....	43

3.3.4	Detection Engineering with Microsoft Sentinel.....	43
<b>4</b>	<b>Implementation.....</b>	<b>45</b>
<b>4.1</b>	<b>Cyber Range development.....</b>	<b>45</b>
4.1.1	Active Directory Environment with IIS & SQL Server.....	46
4.1.2	Firewall with IDS capabilities.....	49
4.1.3	Windows and Linux machine .....	59
4.1.4	SIEM Technologies.....	64
<b>4.2</b>	<b>Modular Asset Modeling &amp; CPE Extraction .....</b>	<b>79</b>
4.2.1	CPE Extraction – Linux Services.....	80
4.2.2	CPE Extraction – Linux Applications.....	82
4.2.3	CPE Extraction – Windows Applications.....	84
4.2.4	CPE Extraction – Windows Services.....	85
<b>4.3</b>	<b>Security Control Integration .....</b>	<b>87</b>
4.3.1	Automated Penetration Testing Agents .....	87
4.3.2	Exploitable Threats in MITRE Attack Format.....	92
4.3.3	Building Custom Threat Profiles .....	92
4.3.4	Detection Engineering with Sentinel .....	92
<b>5</b>	<b>Testing.....</b>	<b>95</b>
<b>6</b>	<b>Conclusions &amp; Future work.....</b>	<b>101</b>
<b>7</b>	<b>References.....</b>	<b>103</b>

## Table of Figures

Figure 1: Different responsibilities in IaaS, PaaS and SaaS.....	11
Figure 2: <a href="#">GOAD (Game of Active Directory) Environment</a> .....	46
Figure 3: The first 5 Virtual machines of the Lab environment.....	47
Figure 4: Network Adapter for VM interconnection.....	48
Figure 5: First LAN of Cyber Range's Network .....	49
Figure 6: <a href="#">PfSense installation media download</a> .....	50
Figure 7: Adding installation media to the newly created VM. ....	50
Figure 8: RAM resources of Firewall.....	50
Figure 9: CPU resources of Firewall .....	51
Figure 10: LAN network .....	51
Figure 11: OPT1 Network .....	51
Figure 12: Wan Network .....	52
Figure 13: PfSense menu.....	52
Figure 14: Interfaces of PfSense.....	53
Figure 15: Adapters of Machine in VirtualBox.....	53
Figure 16: Interface mapping .....	53
Figure 17: WAN IP configuration .....	53
Figure 18: Configuring IP addressing of LAN interface. ....	54
Figure 19: Configuring IP addressing of OPT1 interface.....	54
Figure 20: Final Configuration of PfSense.....	55
Figure 21: Allow all rule for OPT1 network segment.....	56
Figure 22: Suricata package .....	56
Figure 23: General Configuration of Suricata .....	57
Figure 24: Suricata rule feeds.....	58
Figure 25: Copy Suricata logs to Firewall logs .....	58
Figure 26: Update Suricata rule signatures.....	59
Figure 27: Enabled Suricata service in LAN interface.....	59
Figure 28: <a href="#">Ubuntu 20.04 LTS installation page</a> .....	60
Figure 29: RAM resources of Linux machine .....	60
Figure 30: CPU resources of Linux machine .....	61
Figure 31: Storage resources of Linux machine.....	61
Figure 32: Attaching installation media. ....	61
Figure 33: Network resource of Linux machine.....	62
Figure 34: <a href="#">Windows 10 Installation media Download page</a> .....	62
Figure 35: RAM resources for the Windows machine .....	63
Figure 36: CPU resources for the Windows machine .....	63
Figure 37: Storage resources for the Windows machine.....	63
Figure 38: Same Network as the GOAD environment.....	64
Figure 39: Sysmon download page .....	65
Figure 40: Sysmon configuration file. <a href="https://github.com/olafhartong/sysmon-modular">https://github.com/olafhartong/sysmon-modular</a> . .....	66
Figure 41: Sysmon Installation.....	66



Figure 42: Enhanced logging in Active Directory service provided by Microsoft <a href="https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations">https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations</a> .....	67
Figure 43: Additional logging for 4662 events .....	68
Figure 44: Preconfigured Wazuh server .....	69
Figure 45: Default resources of the preconfigured server .....	69
Figure 46: Network interface of Wazuh Server.....	70
Figure 47: Wazuh IP address obtained through DHCP.....	70
Figure 48: Wazuh Web Interface.....	71
Figure 49: Wazuh agent installation page .....	72
Figure 50: Wazuh Windows installation command .....	72
Figure 51: Wazuh agents .....	73
Figure 52: Azure Subscription.....	74
Figure 53: Resource Group .....	74
Figure 54: New Analytic Workspace .....	74
Figure 55: Add Sentinel to Log Analytics workspace.....	75
Figure 56: Sentinel Web interface .....	75
Figure 57: Content Hub .....	76
Figure 58: Connectors page.....	76
Figure 59: Security Events via Legacy Agent page .....	77
Figure 60: Microsoft Monitoring Agent installation .....	77
Figure 61: Heartbeat received.....	78
Figure 62: Additional logs configured.....	79
Figure 63: CPE extraction from Linux services. ....	81
Figure 64: CPE Extraction for Linux Applications .....	83
Figure 65: CPE Extraction - Windows Applications .....	85
Figure 66: CPE Extraction - Windows Services .....	87
Figure 67: Overview of Penetration Testing .....	88
Figure 68: Summary of Penetration Testing steps .....	89
Figure 69: Attributes of the attacks .....	90
Figure 70: Interesting fields extracted. ....	91
Figure 71: Successful attacks. ....	91
Figure 72: MITRE ATTACK Tactics & Techniques.....	92
Figure 73: Credential Compromise Scope Analysis May Detect NTDS .....	93
Figure 74: Analytic rules in Azure Sentinel. ....	94
Figure 75: NTDSUTIL Attack .....	95
Figure 76: NTDSUTIL command .....	95
Figure 77: Services Output: Active Directory Web Services.....	96
Figure 78: Services Output: Kerberos Key Distribution Center .....	96
Figure 79: Services Output: Active Directory Domain Services .....	97
Figure 80: Credential Compromise Scope Analysis May Detect NTDS .....	98
Figure 81: Mapping of the Analytic rule to the MITRE ATTACK Techniques.....	99
Figure 82: Analytic rule created. ....	99
Figure 83: Incident triggered. ....	100

# 1 Introduction

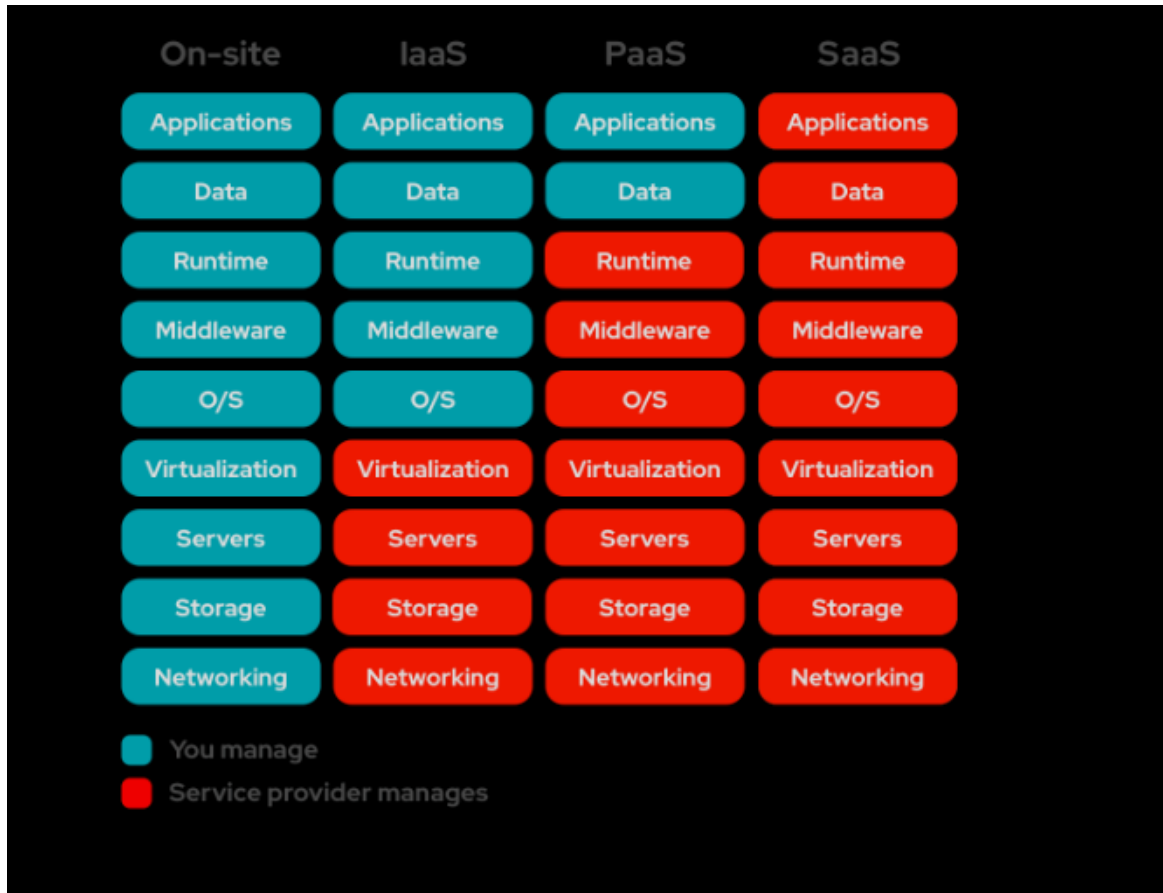
## 1.1 CURRENT STATE OF IT/OT INFRASTRUCTURE

In today's dynamic and interconnected digital landscape, the security of IT and Operational Technology (OT) systems has become paramount. The relentless expansion of technology has given rise to a diverse and complex IT ecosystem that encompasses Cloud Infrastructure, On-Premises IT Infrastructure, and the rapidly proliferating Internet of Things (IoT) and Operational Technology (OT) environments. Understanding the intricacies of these three key facets is essential in the pursuit of robust cybersecurity measures.

### 1.1.1 Cloud Infrastructure

Cloud environments have revolutionized the way organizations deploy and manage their IT infrastructures. They offer a wide range of technologies and services that provide flexibility, scalability, and cost-efficiency. Three prominent cloud service providers, Azure, AWS, and GCP, offer comprehensive solutions to meet the diverse needs of organizations.

Azure, Microsoft's cloud computing platform, provides a wide array of services, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). With Azure IaaS, organizations can provision virtual machines, storage, and networking resources to build their infrastructure. Azure PaaS offers managed services for application development, database management, and analytics. Additionally, Azure SaaS provides ready-to-use software applications, such as Microsoft Office 365 and Dynamics 365. These cloud services enable organizations to leverage external resources, reduce infrastructure costs, and scale their operations as needed.



*Figure 1: Different responsibilities in IaaS, PaaS and SaaS*

AWS, Amazon's cloud platform, offers a comprehensive suite of cloud services covering various domains. AWS provides IaaS solutions through Amazon Elastic Compute Cloud (EC2) for virtual machine provisioning and Amazon Simple Storage Service (S3) for scalable storage. AWS PaaS includes services like AWS Lambda for serverless computing, Amazon RDS for managed relational databases, and Amazon DynamoDB for NoSQL database management. As for AWS SaaS, popular examples include Amazon S3 for cloud storage and Amazon Connect for cloud-based contact centers. AWS's extensive range of services enables organizations to build and manage their IT infrastructure with high scalability and reliability.

GCP, Google's cloud platform, offers a rich set of services for cloud computing. GCP's IaaS offerings, such as Compute Engine and Cloud Storage, provide flexible virtual machine instances and scalable storage options. GCP's PaaS includes services like Google App Engine for application deployment, Cloud Spanner for distributed databases, and BigQuery for analytics. GCP also offers SaaS solutions like Google Workspace (formerly G Suite), which includes productivity tools like Gmail, Google Docs, and Google Drive. GCP's services are designed to enable organizations to build, deploy, and scale applications and services with ease.

While cloud environments bring numerous benefits, they also introduce challenges in asset management. One key difficulty is the dynamic nature of cloud infrastructures. Assets in the cloud can be provisioned, scaled, and deprovisioned on-demand, making it challenging to maintain an up-to-date inventory. Traditional asset management approaches that rely on manual tracking or periodic scans may struggle to keep pace with the rapid changes and scalability of cloud environments. Organizations need automated and real-time asset inventory techniques that can accurately capture and track assets as they are dynamically created, modified, and retired.

Another challenge is the diverse and complex nature of cloud technologies. Cloud environments often comprise a combination of virtual machines, containers, serverless functions, storage systems, and other services. Each of these components has its own unique properties, configurations, and dependencies, making it challenging to develop a unified asset management approach. Organizations must consider the heterogeneity of cloud technologies and develop asset modeling techniques that can capture the intricacies of these different components.

Additionally, multi-cloud and hybrid cloud environments further complicate asset management. Many organizations leverage multiple cloud providers or maintain a combination of cloud and on-premises infrastructure. Managing assets across these diverse environments requires a holistic approach that can seamlessly integrate and synchronize asset information. Organizations must address the challenges of asset management across multiple clouds, ensure consistent asset tracking, and reconcile differences in asset management practices.

To overcome these challenges, organizations can leverage various tools and techniques provided by cloud service providers and third-party vendors. Cloud-native tools, such as Azure Resource Manager, AWS CloudFormation, and GCP Deployment Manager, enable organizations to define and manage cloud infrastructure as code. These tools can be integrated with asset management systems to automatically capture asset information as part of the infrastructure provisioning process. Additionally, cloud asset management platforms and configuration management tools, such as Azure Arc, AWS Config, and GCP Cloud Asset Inventory, provide comprehensive asset tracking and configuration management capabilities across cloud environments.

### **1.1.2 On-Premises IT Infrastructure**

In addition to cloud environments, on-premises IT infrastructures remain prevalent in many organizations. These infrastructures consist of physical servers, networking equipment, storage systems, and other hardware components located within an organization's premises. On-premises environments often host critical systems and services, such as Active Directory, databases, and internal applications, which form the backbone of an organization's operations.

One significant component of on-premises environments is Active Directory (AD), a directory service provided by Microsoft. AD is commonly used for centralized user management, authentication, and authorization within an organization's IT infrastructure. It facilitates the management of user accounts, security groups, and access

policies. AD also enables the integration of various services, such as email systems, file servers, and internal applications. However, the complexity of AD environments poses challenges for asset management.

Asset management in on-premises environments faces difficulties due to the decentralized nature of infrastructure components and the diverse range of technologies involved. On-premises environments often comprise a mix of physical and virtual servers, network switches, routers, firewalls, and storage devices from different vendors. Each of these components may have specific configuration requirements, management interfaces, and monitoring capabilities, making asset management a complex and resource-intensive task.

One particular challenge is the lack of standardized asset identification and tracking mechanisms in on-premises environments. Unlike cloud environments that offer built-in asset tracking features, on-premises environments typically require organizations to implement custom solutions or rely on third-party tools for asset management. This lack of standardization makes it difficult to maintain a unified and up-to-date inventory of assets. Organizations often resort to manual inventory checks, periodic scanning, or legacy systems that may not provide real-time visibility into the state of assets.

Another challenge is the diverse array of technologies and legacy systems that coexist in on-premises environments. These environments may incorporate a mix of Windows, Linux, and Unix-based systems, each with its own management tools and methodologies. Managing assets across these heterogeneous systems requires organizations to develop strategies that can handle the complexities and nuances of each technology. Furthermore, legacy systems may lack modern asset management capabilities, making it harder to integrate them into centralized asset management processes.

Additionally, on-premises environments often lack the scalability and elasticity offered by cloud environments. The capacity planning and resource allocation in on-premises infrastructures require careful consideration of hardware limitations and future growth projections. As organizations scale their operations or introduce new services, it becomes essential to accurately track and manage the associated assets. Failure to do so can result in overprovisioning, underutilization of resources, or inadequate capacity, leading to increased costs and inefficiencies.

To address these challenges, organizations can implement various asset management practices and technologies. Automated discovery and inventory tools can help scan the network and collect information about connected devices, their configurations, and software installed. Configuration management databases (CMDBs) provide a central repository for storing asset-related data, including hardware specifications, software versions, and relationships between assets. Additionally, organizations can implement network monitoring solutions that provide real-time visibility into the status and performance of on-premises assets.

### 1.1.3 IOT/OT Infrastructure

The proliferation of Internet of Things (IoT) and Operational Technology (OT) devices has brought about a new wave of challenges for asset management. IoT devices are typically small, connected devices equipped with sensors and actuators that collect and transmit data over the internet. They are utilized in various domains, including smart homes, industrial automation, healthcare, and transportation. On the other hand, OT devices are used in industrial settings to monitor and control physical processes such as manufacturing, energy production, and infrastructure management.

IoT/OT environments pose unique challenges for asset management due to the characteristics of the devices and the specific protocols and communication patterns they employ. Unlike traditional IT assets, IoT/OT devices often have resource-constrained hardware, limited processing power, and restricted memory and storage capacities. Additionally, they may operate on diverse communication protocols such as Bluetooth, Zigbee, Z-Wave, or industrial protocols like Modbus, OPC, and Profibus.

Asset discovery and identification in IoT/OT environments can be challenging due to the large number of devices, their mobility, and the dynamic nature of network topologies. Devices may join or leave the network frequently, making it difficult to maintain an accurate and up-to-date inventory. Furthermore, IoT/OT devices often lack standard identification mechanisms, such as unique identifiers or well-defined metadata, making it harder to track and manage them effectively.

Another challenge lies in the heterogeneity of IoT/OT devices and their proprietary software stacks. Each device may have its own firmware, operating system, and management interfaces, making it difficult to standardize asset management processes across the entire ecosystem. Furthermore, device manufacturers often provide limited visibility into the inner workings of their devices, making it challenging to extract detailed information about their configurations, vulnerabilities, or software versions.

Security is another critical aspect of asset management in IoT/OT environments. Due to the nature of IoT/OT devices being connected to physical processes or controlling critical infrastructure, the implications of compromised devices can be severe. IoT/OT environments are vulnerable to various attacks, including device tampering, unauthorized access, data breaches, and denial-of-service attacks. Managing the security posture of IoT/OT assets, including applying firmware updates, implementing access controls, and monitoring device behavior, becomes crucial for maintaining a secure and resilient infrastructure.

To address these challenges, organizations can employ specialized IoT/OT asset management solutions. These solutions provide features such as device discovery, inventory management, remote configuration, firmware updates, and security monitoring for IoT/OT assets. They may utilize protocols like MQTT, CoAP, or OPC UA for communication with IoT/OT devices and integrate with centralized asset management systems or security information and event management (SIEM) platforms.

Asset management in these diverse technology environments, including cloud, on-premises, and IT/IoT, presents numerous challenges that need to be addressed. The dynamic nature of these environments, with constant additions and modifications of services, systems, and devices, poses difficulties in maintaining accurate asset

inventories. The sheer scale and complexity of these infrastructures further complicate the asset management process, requiring comprehensive solutions to track, monitor, and manage assets effectively.

To address these challenges, organizations can leverage specialized asset management tools and methodologies that provide features such as discovery, identification, tracking, and monitoring of assets across cloud, on-premises, and IT/IoT environments. These solutions should integrate with existing systems and frameworks, such as Mitre's CPE catalog and Digital Artifact Ontology, and leverage scripting languages and automation techniques like PowerShell, CMD, and Bash to extract relevant asset information. By developing comprehensive asset models and implementing efficient inventory management processes, organizations can gain better visibility and control over their assets, enabling effective risk assessment, compliance, and security measures.

## **1.2 MOTIVATION**

In an attempt to enable the enumeration of assets MITRE and the National Institute of Technology (NIST) have created an open-source asset enumeration catalogue called Common Platform Enumeration (CPE), which contains roughly a million entries of known hardware, operating systems, and application components.

While such asset modeling techniques cover the needs of typical IT systems, they usually contain marginal, or no information related to CPS. For example, a lot of information related to specialized components is missing, such as background services to operating systems, custom operating systems such as the firmware of SCADA devices, and widely used home applications such as known video games.

Another challenge identified in this field is the grouping of assets into composite systems such as devices and networks and the identification of their applicable internal and external interaction surface.

Finally, existing asset modeling techniques for CPS do not consider the different views, in the context of the applicable vectors of communication.

In today's interconnected digital landscape, effective asset management is crucial for maintaining the security and integrity of IT systems. Organizations face the challenge of safeguarding sensitive data, preventing potential threats, and ensuring compliance. To address these challenges, it is imperative to explore various aspects of asset inventory management.

### **1.2.1 Asset Inventory management**

Asset Inventory Management is a systematic method for identifying, cataloging, and monitoring both digital and physical assets within an organization's IT environment. It serves as the foundation for a robust cybersecurity strategy.

**Automated discovery and inventory** solutions employ scanning and monitoring techniques to identify and record assets within the IT infrastructure. These tools utilize various protocols to gather asset information such as IP addresses, MAC addresses, device types, and installed software. They offer real-time visibility into assets and can

integrate with network management systems and asset repositories for centralized tracking and management.

**Configuration Management Databases (CMDB)** store detailed information about configuration items (CIs) in an organization's IT infrastructure, including hardware and software, along with their relationships. They support tracking asset attributes, dependencies, ownership, and version control. CMDBs help in understanding the relationships between assets, facilitating change management, incident resolution, and compliance tracking.

**Asset tagging and labeling** involve physically labeling assets with unique identifiers or barcodes that link to asset information in a database. This method aids in asset identification, tracking, and inventory management, primarily for physical assets in on-premises environments and IoT devices.

**Asset lifecycle management** involves tracking assets from procurement to retirement. It includes recording asset details such as purchase date, warranty information, maintenance history, and disposal processes. Managing assets throughout their lifecycle helps optimize utilization, maintain compliance with licensing agreements, and plan for replacements or upgrades.

**Software Asset Management (SAM)** focuses on managing software assets, including tracking licenses, usage, compliance, and entitlements. SAM solutions provide visibility into software installations, help manage license agreements, and ensure compliance with vendor requirements, reducing costs and mitigating legal and security risks.

**Cloud asset management** solutions are tailored to track and manage assets in cloud environments, providing visibility into virtual machines, storage, and services across multiple cloud providers. They monitor resource utilization, control costs, enforce security policies, and ensure compliance with cloud-specific regulations.

Each of these asset management methods offers different approaches suitable for diverse technology environments. Organizations can choose and combine these methods based on their specific requirements, infrastructure complexity, and regulatory compliance needs. Implementing these asset management solutions improves visibility, control, and optimization of assets, leading to enhanced security, compliance, and operational efficiency.

### **1.2.2 Network Asset Management**

In the realm of asset inventory management, Network Asset Management takes center stage as a pivotal component. This facet involves the systematic management and control of an organization's network resources, encompassing a spectrum of solution methods:



**Network monitoring and alerting solutions** form the vanguard of Network Asset Management. These tools employ continuous surveillance of network traffic, devices, and performance metrics. Their mission is to unearth anomalies, troubleshoot issues, and ensure optimal network performance. By providing real-time visibility into network health, bandwidth utilization, device availability, and performance metrics, they serve as the vigilant guardians of network integrity. These tools also generate alerts and notifications for network events, facilitating timely responses and proactive network management.

Centralization is the hallmark of **Network Configuration Management** solutions. They assist organizations in managing and maintaining the configurations of network devices, including routers, switches, and firewalls. By centralizing network device configurations, enforcing configuration standards, and automating configuration deployment and changes, these tools ensure consistency and compliance. In doing so, they significantly reduce network downtime, enhance security, and streamline network administration tasks, thereby harmonizing the network's operational symphony.

**Network Performance Optimization** solutions dedicate their efforts to enhancing network efficiency. Employing techniques like traffic shaping, load balancing, Quality of Service (QoS) policies, and bandwidth management, they optimize network utilization and prioritize critical traffic. Their mission is to maximize network throughput, minimize latency, and guarantee a seamless user experience. These tools act as the virtuoso conductors, orchestrating the network's performance to deliver a symphony of efficiency.

**Network Security Management** solutions stand as the sentinels guarding the network's citadel against unauthorized access, threats, and vulnerabilities. Their arsenal comprises firewall management, intrusion detection and prevention systems (IDPS), virtual private networks (VPNs), and robust authentication mechanisms. These tools provide centralized visibility and control over network security policies, enabling the identification of security incidents and facilitating timely responses and mitigation. In essence, they fortify the network's ramparts, ensuring its security.

**Network Configuration Auditing and Compliance** solutions play a pivotal role in ensuring regulatory harmony within the network environment. They analyze network configurations, compare them against predefined standards, and generate reports spotlighting configuration deviations or vulnerabilities. In doing so, they assist organizations in maintaining a secure and compliant network infrastructure. These tools act as the meticulous auditors of the network's adherence to industry regulations and security best practices.

Finally, **Software-Defined Networking (SDN)** emerges as a paradigm shift in Network Asset Management. SDN decouples the control plane from the data plane, ushering in a new era of centralized network management and programmability. SDN solutions provide a software-based control layer that empowers administrators to dynamically configure and manage network resources, policies, and services. With its enhanced agility, scalability, and flexibility, SDN reshapes the way organizations manage their network infrastructure, acting as the network's maestro.

These diverse solution methods empower organizations with an array of tools to effectively manage their network assets. Depending on their unique requirements, organizations can adopt a combination of these methods to monitor, configure, optimize, secure, and ensure compliance within their networks. By implementing robust Network Asset Management solutions, organizations can enhance network performance, security, and overall operational efficiency.

### 1.2.3 User Access and Privilege Management

Within the realm of asset inventory management, User Access and Privilege Management play a pivotal role, ensuring that the keys to the digital fortress remain in capable hands. This facet encompasses a spectrum of solution methods:

**Role-Based Access Control (RBAC)** stands as a widely embraced methodology for managing user access and privileges. It operates on the premise of assigning users to specific roles based on their responsibilities within an organization. Each role is intricately associated with a defined set of permissions, outlining the actions and resources accessible to users within that role. RBAC simplifies access management by granting permissions at the role level, thus eliminating the need for individual user-level permissions.

**Identity and Access Management (IAM)** systems offer a comprehensive framework for managing user identities, authentication, and access to resources. These systems encompass a suite of features, including user provisioning, single sign-on (SSO), multi-factor authentication (MFA), and password management. IAM solutions streamline user onboarding and offboarding, enforce access policies, and ensure secure user authentication across a multitude of systems and applications.

**Privileged Access Management (PAM)** directs its focus toward managing and securing privileged accounts and access rights within an organization. Privileged accounts wield elevated permissions and control over critical systems and data. PAM solutions enforce stringent controls, such as password vaulting, session monitoring, and just-in-time access provisioning, effectively mitigating the risks associated with unauthorized access and potential misuse of privileged credentials.

**User Behavior Analytics (UBA)** solutions harness the power of machine learning and behavioral analytics to identify anomalous user behavior and potential security threats. By meticulously analyzing user activities, UBA systems establish baseline behavior patterns and identify deviations indicative of compromised accounts, insider threats, or unauthorized access attempts. UBA enhances user access management by providing continuous monitoring and early detection of suspicious activities.

The **Least Privilege Principle** advocates for granting users the minimal level of access required to perform their job functions. By adhering to this doctrine, organizations minimize the risk of inadvertent or deliberate privilege misuse and mitigate the potential impact of security incidents. Implementation of the Least Privilege Principle necessitates

regular access reviews, fine-grained permission controls, and ongoing monitoring of user access rights.

**Identity Governance and Administration (IGA)** solutions offer organizations a streamlined approach to managing user identities, access rights, and entitlements. These solutions automate user lifecycle processes, encompassing onboarding, role transitions, and offboarding, while concurrently ensuring compliance with regulatory mandates. IGA furnishes centralized visibility and control over user access, simplifies access certification, and elevates overall governance standards concerning user identities and privileges.

These diverse solution methods empower organizations to effectively manage user access and privileges. Depending on their unique requirements and organizational structure, entities can adopt a combination of these methods to bolster security, minimize the risk of unauthorized access, ensure regulatory compliance, and maintain a robust grip on their systems and data.

### **1.3 CONTRIBUTION**

In the context of cybersecurity for IT systems, our research introduces a comprehensive methodology aimed at automating asset modeling within the domain of cyber systems. This innovative approach is fortified with a robust implementation and meticulous validation, offering noteworthy contributions that can be summarized as follows:

#### **1.3.1 Cyber Range Development**

The creation of a dedicated Cyber Range constitutes a pivotal component of our methodology. This Cyber Range serves as a controlled environment specifically designed to subject our asset modeling approach to rigorous testing and evaluation. By incorporating both Windows and Linux systems within this testing ecosystem, we ensure a comprehensive assessment of our methodology's adaptability across heterogeneous IT environments. This multipronged approach mirrors the real-world IT landscape, allowing us to gauge the robustness and versatility of our asset modeling framework.

The Cyber Range also serves as an invaluable source of data. It allows us to gather extensive data on asset behavior, interactions, vulnerabilities, and responses to cyber threats. This data, collected in a controlled and structured manner, forms the basis for our thorough validation process. Through data analysis, we can identify patterns, anomalies, and areas for improvement, ultimately refining our asset modeling methodology to better align with the dynamic threat landscape.

#### **1.3.2 Modular Asset Modeling Framework**

Our methodology hinges on a modular asset modeling approach, which accounts for multifaceted dimensions encompassing cyber and physical devices, as well as user and network interactions. By embracing a modular framework, we ensure a comprehensive representation of assets, thus enhancing the effectiveness of our cybersecurity strategy.

At the core of our framework lies a component-based approach to asset modeling. Rather than treating assets as monolithic entities, we break them down into granular components, considering various facets such as hardware, software, data, and user attributes. This granular representation enables a more precise understanding of the assets, facilitating targeted security measures.

The framework comprises interconnected modules, each dedicated to a specific aspect of asset modeling. These modules include but are not limited to:

**Device Modeling:** This module focuses on the physical and digital attributes of devices, encompassing hardware specifications, software configurations, and firmware versions.

**User Behavior Modeling:** Understanding user interactions and behavior is vital. This module tracks user activities, privileges, and access patterns, contributing to a more holistic view of asset utilization.

**Network Topology Modeling:** To account for network-centric threats, our framework includes a module for modeling network topologies, ensuring that asset interactions within the network are comprehensively analyzed.

### **1.3.3 Automated CPE Management Tool**

In support of our methodology, we have developed a specialized tool that streamlines the process of Common Platform Enumeration (CPE) harvesting and production. This tool proves invaluable in cases where existing CPEs are inadequate. It automates these crucial steps, facilitating the efficient integration of assets into our cybersecurity framework.

In situations where predefined CPEs are unavailable or inadequate, the tool facilitates dynamic CPE production. This involves the generation of CPEs based on the tool's analysis of asset characteristics, including software versions, hardware specifications, and configuration settings. This adaptive approach ensures that assets are accurately represented, even when dealing with non-standard or custom components.

Beyond CPE collection and generation, the tool plays a pivotal role in connecting CPEs to potential threats and existing security controls. It correlates CPE data with known vulnerabilities, exploits, and threat intelligence feeds, allowing security professionals to assess the risk associated with each asset. Moreover, it identifies the security controls in place and their effectiveness in mitigating identified risks.

### **1.3.4 Security Control Integration**

A pivotal aspect of our research involves the seamless connection of CPEs to potential threats, culminating in the identification and assessment of existing security controls. This critical step ensures that our cybersecurity strategy not only models' assets but also provides actionable insights into safeguarding them against emerging threats.

## 2 Related Work

### 2.1 FRAMEWORKS, ONTOLOGIES AND DATABASES

The control, maintenance, and security of information systems have grown increasingly complex due to the expansion of existing systems with new technologies like IoT devices and distributed systems. This complexity has created more opportunities for malicious entities to launch attacks as the number of devices and software exposed in networks continues to expand. Identifying and assessing security weaknesses and connecting them to potential threats and attacks has become a challenging and time-consuming task. Therefore, there's a need to develop methodologies that enable organizations to maintain accurate inventories of their information systems, ensuring their effective operation and security. Such inventories offer opportunities for organization and control while providing a basis for risk assessment by linking asset identifiers to libraries that record vulnerabilities.

To establish a robust and reliable security ontology, it's crucial to leverage a plethora of catalogs and models maintained by organizations like NIST, MITRE, and FIRST. Notable among these are the Common Platform Enumeration (CPE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), and the Common Vulnerabilities and Exposures (CVE) database. These resources collectively provide a structured naming scheme for hardware, software, and

operating systems, categorize software and hardware weaknesses, outline common attack patterns, and list known vulnerabilities, respectively.

### **2.1.1 Common Platform Enumeration (CPE)**

The Common Platform Enumeration (CPE) [1] plays a pivotal role in asset inventorying for cybersecurity within IT systems. It serves as a standardized method for describing and identifying classes of software, operating systems, and hardware. The CPE schema enables the enumeration of components within an enterprise network, serving as a source of information for enforcing and verifying management policies related to vulnerability management and configuration.

CPE categorizes assets into three distinct pillars: Hardware Platform, Operating System Platform, and Application Environment. Each pillar encapsulates critical attributes, including Part (identifying software applications, operating systems, or hardware), Vendor (identifying the manufacturer or creator of the product), Product (naming the product), and Version (specifying the specific version of the product).

One of the key advantages of CPE is its uniform naming scheme, which allows for the consistent and predictable creation of identifiers for new platforms. This uniformity fosters interoperability and synchronization among automated tools by enabling them to use these identifiers. Additionally, CPE entries are meticulously linked to corresponding entries in the CVE and CWE lists, creating a robust ecosystem for vulnerability management and threat mitigation.

### **2.1.2 Common Weakness Enumeration (CWE)**

Within the realm of cybersecurity, the Common Weakness Enumeration (CWE) [2] is an indispensable system for categorizing software and hardware weaknesses. These weaknesses can encompass defects or flaws in various aspects, including software and hardware design, architecture, code, or implementation. When left unaddressed, these weaknesses render components vulnerable to exploitation by malicious entities.

The CWE catalog serves as a widely embraced resource in the realm of system security. It systematically categorizes and describes software and hardware weaknesses, providing invaluable insights to security professionals and developers. This categorization facilitates the understanding of vulnerabilities and aids in the development of effective mitigation strategies.

CWE's comprehensive framework allows for the identification and classification of vulnerabilities, making it an essential component of the cybersecurity landscape. By leveraging CWE, organizations can proactively address and remediate vulnerabilities, bolstering their overall security posture.

### **2.1.3 Common Attack Pattern Enumeration and Classification (CAPEC)**

Understanding how malicious entities launch attacks against systems is paramount for timely and effective response. The Common Attack Pattern Enumeration and

Classification (CAPEC) [3] catalog serves as a valuable resource in this regard. It provides security analysts with a publicly available repository of common attack patterns, offering insights into how attackers exploit weaknesses in components.

CAPEC's attack patterns are descriptive of common characteristics and methods employed by attackers to exploit known vulnerabilities in a system. Each attack pattern offers knowledge about the planning and execution of specific attack aspects, along with guidance on addressing vulnerabilities and preventing attacks.

#### **2.1.4 Common Vulnerabilities and Exposures (CVE)**

The Common Vulnerabilities and Exposures (CVE) [4] [5] list represents a comprehensive compilation of records, each assigned a unique identification number following the format CVE-YYYY-NNNN. These records encompass descriptions of vulnerabilities associated with technology systems as defined by the Common Platform Enumeration (CPE) database. A CVE entry provides a detailed account, including a description of the vulnerability, at least one public reference to the vulnerability, the Common Vulnerability Scoring System (CVSS) risk score, references to exploit code, technical analysis, and methods to mitigate the threat.

CVE entries serve as a foundational element of the cybersecurity landscape, facilitating the sharing of security information among various tools and platforms. Organizations benefit from the free accessibility of CVE entries, which empower them to search for vulnerabilities within their systems and services. Entries are contributed by organizations known as CVE Number Authorities (CNAs), authorized to assign CVE IDs to vulnerabilities disclosed to software and hardware vendors by researchers.

#### **2.1.5 Common Vulnerability Scoring System (CVSS)**

The Common Vulnerability Scoring System (CVSS) [6] [7] is instrumental in quantifying the severity of vulnerabilities, providing a score that categorizes threats into different risk levels. This score serves as a valuable tool for organizations to prioritize their efforts in addressing vulnerabilities effectively. The CVSS score falls into the following categories: None, Low, Medium, High, and Critical, each corresponding to a specific risk level.

The CVSS score is derived from the values assigned to various CVSS metrics, including Base, Temporal, and Environmental metrics. These metrics collectively offer a comprehensive assessment of the vulnerability's attributes, potential impact, and dynamic aspects. The resulting CVSS score provides a concise representation of the vulnerability's overall risk.

#### ***Base Metrics***

The Base Metrics category within the Common Vulnerability Scoring System (CVSS) encompasses several critical factors that contribute to the overall assessment of a vulnerability's severity. These metrics provide detailed insights into the characteristics of a vulnerability that remain constant over time and across various environments.

Attack Vector (AV) is a key metric that reflects the framework through which a vulnerability can be exploited. It categorizes attack vectors into different levels, including Network (N), Adjacent (A), Local (L), and Physical (P). The choice of attack vector helps security professionals understand the accessibility and proximity required for an attacker to exploit the vulnerability.

Attack Complexity (AC) describes the level of complexity involved in successfully exploiting a vulnerability. It is classified into Low (L), indicating straightforward exploitation, and High (H), signifying a more intricate and resource-intensive attack process.

Privileges Required (PR) highlights the level of access or privileges an attacker needs to exploit a vulnerability. It distinguishes between None (N), Low (L), and High (H) privileges, offering insights into the attacker's requirements to compromise the target.

User Interaction (UI) assesses whether user interaction is necessary for an attacker to complete the exploit. It differentiates between None (N), signifying no user interaction, and Required (R), indicating that the victim user must perform specific actions for the exploit to succeed.

Scope (S) pertains to the vulnerability's scope, determining whether an exploited vulnerability affects resources beyond its perimeter. It can be categorized as Unchanged (U) or Changed (C), reflecting whether the exploitation extends beyond the vulnerable element.

Confidentiality (C), Integrity (I), and Availability (A) metrics gauge the impact of the vulnerability on these respective aspects. They classify the impact as High (H), Low (L), or None (N), providing valuable insights into the potential consequences of a successful attack.

### ***Temporal Metrics***

Temporal Metrics in the Common Vulnerability Scoring System (CVSS) focus on aspects that can change over time and influence the severity of a vulnerability. These metrics adjust the vulnerability's base score based on evolving factors that impact its exploitability and the availability of mitigations.

Exploit Code Maturity (E) assesses the likelihood of an active exploit being available for a vulnerability. It can take values such as Not Defined (X), High (H), Functional (F), Proof-of-Concept (P), or Unproven (U), offering insights into the maturity of exploit code.

Remediation Level (RL) reflects the current state of correction for a vulnerability. It helps in prioritization by considering whether solutions are available. Possible values include Not Defined (X), Unavailable (U), Workaround (W), Temporary Fix (T), and Official Fix (O).

Report Confidence (RC) gauges the confidence in the reporting of a vulnerability. It provides an indication of the reliability of known technical details and can be Confirmed (C), Reasonable (R), or Unknown (U), among other values.



These temporal metrics enhance the CVSS score by considering factors that change over time, ensuring that the score remains dynamic and reflective of the evolving threat landscape.

### ***Environmental Metrics***

The Environmental Metrics category within the Common Vulnerability Scoring System (CVSS) is pivotal in shaping the vulnerability rating based on the specific conditions and context within an organization's environment. These metrics are particularly relevant when environmental information regarding network elements is available.

Environmental Metrics acknowledge that vulnerabilities may not be universally exploitable and can be contingent on certain conditions or factors within an organization's environment. For instance, a vulnerability may exist, but its exploitability in an organization's specific environment can vary based on various factors.

Factors considered in this category include network access, security controls, configuration, and segmentation. These factors help in determining the potential impact of a vulnerability within the organization's unique setup.

Based on the values assigned to the aforementioned CVSS metrics, a score is calculated, categorizing the threat into five levels: None, Low, Medium, High, and Critical. Each level corresponds to a specific range of scores, allowing for a clear classification of the threat's severity.

The CVSS score serves as a valuable tool for prioritizing vulnerability remediation efforts. It aids in identifying the most critical vulnerabilities that require immediate attention. Additionally, the CVSS score is accompanied by a vector text representation, providing a detailed breakdown of the individual metric values.

While CVSS is a robust framework, its full potential is realized when combined with data-driven threat intelligence sources like the Exploit Prediction Scoring System (EPSS). These sources offer predictive insights into the likelihood of a vulnerability being exploited, enhancing the prioritization of vulnerability management efforts.

#### **2.1.6 Exploit Prediction Scoring System (EPSS)**

The Exploit Prediction Scoring System (EPSS) [8] is a data-driven initiative aimed at estimating the probability of a software vulnerability being exploited "in the wild." EPSS is designed to be simple yet flexible, providing accurate estimates of the likelihood of vulnerability exploitation. Its primary goal is to assist security analysts in prioritizing vulnerability remediation.

EPSS leverages machine learning techniques in its process for calculating this predictive score. Although the underlying algorithms may be intricate, the objective is to provide a simplified representation of the process. EPSS considers a wide range of factors, including vulnerability characteristics, historical exploitation trends, and other relevant data sources.

By distilling complex data inputs into a predictive score, EPSS equips security professionals with a valuable tool for prioritizing their vulnerability management efforts.

## **2.2 ASSET INVENTORY & VULNERABILITY MANAGEMENT TOOLS**

Asset Inventory and Vulnerability Management Tools is a critical component of modern cybersecurity strategies. These tools play a pivotal role in identifying, cataloging, and securing digital and physical assets within an organization's IT infrastructure. Our exploration will uncover the diverse range of tools available and their efficacy in mitigating vulnerabilities, ultimately contributing to a more robust cybersecurity posture.

### **2.2.1 Asset Inventory Management tools**

Asset Inventory Management is a systematic approach for identifying, cataloging, and monitoring both digital and physical assets within an organization's IT environment. It's a fundamental component of a robust cybersecurity strategy.

**Qualys AssetView** is an automated tool that scans and monitors assets in real-time, identifying details such as IP addresses, MAC addresses, device types, and installed software. It ensures comprehensive visibility into an organization's assets and integrates seamlessly with network management systems.

**ServiceNow CMDB** is a centralized database that stores detailed information about an organization's hardware and software assets, including their relationships, dependencies, ownership, and version control. It simplifies change management, incident resolution, and compliance tracking.

The **RFID Asset Tracking System** involves physically labeling assets with unique identifiers or barcodes, simplifying asset identification, tracking, and inventory management. It's especially useful for managing physical assets in on-premises environments and IoT devices.

**SolarWinds Web Help Desk** assists in tracking assets from procurement to retirement by recording purchase dates, warranty information, maintenance history, and disposal processes. It optimizes asset utilization, ensures compliance with licensing agreements, and facilitates strategic asset planning.

**Snow License Manager** focuses on managing software assets, tracking licenses, usage, compliance, and entitlements. It provides visibility into software installations, helps manage license agreements, and ensures compliance with vendor requirements, reducing costs and mitigating legal and security risks.

**CloudCheckr** is tailored for tracking and managing assets in cloud environments. It provides visibility into virtual machines, storage, and services across multiple cloud providers, monitors resource utilization, controls costs, enforces security policies, and ensures compliance with cloud-specific regulations.

Each of these asset management methods offers different approaches suitable for various technology environments. Organizations can choose and combine these methods based on their specific requirements, infrastructure complexity, and regulatory

compliance needs. Implementing these asset management solutions enhances visibility, control, and optimization of assets, leading to improved security, compliance, and operational efficiency.

### **2.2.2 Network Asset Management tools**

In the realm of IT infrastructure management and cybersecurity, organizations rely on a suite of specialized solutions to ensure the efficiency, security, and compliance of their network assets. These solutions play a vital role in safeguarding digital territories and optimizing network performance.

**Network Configuration Management (NCM)** solutions are the backbone of efficient network administration. They provide automation and oversight of network devices, including routers and switches. NCM tools offer features like automated backups, version control, and change management to maintain consistent, secure, and compliant configurations. For instance, **SolarWinds Network Configuration Manager** simplifies the management of network configurations across diverse devices. It automates routine tasks, tracks configuration changes, and enforces compliance with defined standards. This ensures network stability and reduces the risk of errors, contributing to reliable network operations.

**Network Performance Optimization (NPO)** solutions enhance network efficiency, reliability, and speed. These tools are designed to streamline network traffic, allocate resources effectively, and minimize latency. They ensure that the network operates smoothly, meeting performance requirements and delivering an optimal user experience. Consider **Riverbed SteelHead**, an NPO solution that employs data compression, deduplication, and traffic prioritization techniques. It optimizes network performance by reducing bandwidth usage, accelerating data transfer, and improving application response times. This results in enhanced productivity and a seamless user experience.

**Network Security Management** solutions are critical for safeguarding network assets from unauthorized access, cyberattacks, and vulnerabilities. They encompass technologies like firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and network access control (NAC) to protect sensitive data and maintain compliance with security policies. For instance, **Cisco Firepower** combines firewall capabilities with advanced security features, ensuring robust protection for network infrastructure. It detects and mitigates threats, enforces security policies, and provides comprehensive network security.

**Network Configuration Auditing and Compliance** solutions scrutinize network configurations against industry standards, security policies, and regulatory requirements. They identify deviations or vulnerabilities and help organizations maintain consistency, security, and compliance. **Tenable Nessus**, a prominent example, scans network devices and configurations, highlighting areas where compliance standards or recommended settings are not met. Network administrators can then take corrective actions to align configurations with established standards, ensuring a secure and compliant network environment.

**Software-Defined Networking (SDN)** solutions revolutionize network management by separating the control plane from the data plane. SDN allows for centralized control and dynamic configuration of network resources, enhancing flexibility and programmability. **Cisco Application Centric Infrastructure (ACI)** exemplifies SDN in action. It simplifies network provisioning and management by enabling administrators to define network policies in software. These policies can be adjusted on-the-fly to adapt to changing network requirements, ensuring an agile and responsive network infrastructure.

In conclusion, these network management and security solutions form the bedrock of modern IT infrastructure management and cybersecurity. They empower organizations to maintain efficient, secure, and compliant network environments, providing the agility needed to navigate the digital frontier effectively.

### **2.2.3 User Access and Privilege Management tools**

In the world of cybersecurity, User Access and Privilege Management solutions are pivotal. They provide the means to oversee and control how users access network resources, ensuring security and adherence to organizational guidelines.

**Identity and Access Management (IAM)** solutions serve as the custodians of digital identity. They oversee user identities, roles, and permissions within an organization's network. IAM tools determine who can access specific resources based on predefined roles and guarantee that users have appropriate access levels. A prime example of IAM technology is **Microsoft Azure Active Directory**. It streamlines user identity management and access control across Microsoft 365 and Azure services. This simplifies the process of provisioning users, enforces multifactor authentication, and offers detailed access management.

**Privileged Access Management (PAM)** solutions are dedicated to safeguarding crucial systems and data by meticulously controlling and monitoring privileged user access. Privileged users, such as administrators and IT staff, possess elevated access privileges, and PAM tools are instrumental in securing and auditing their actions. **CyberArk** stands as an esteemed PAM solution, providing secure storage and administration of privileged account credentials. It imposes strict access controls, session recording, and password rotation policies. This guarantees that only authorized individuals access sensitive systems and that their activities are tracked and logged.

**Single Sign-On (SSO)** solutions streamline user access by enabling users to log in once and gain entry to numerous applications and resources without repeatedly entering their credentials. This boosts user productivity and mitigates security risks linked to password management. **OneLogin** serves as an SSO solution that seamlessly integrates with diverse applications and services. It offers a secure portal where users can access all authorized resources using a single set of credentials. This simplifies access management and reduces the complexity of password management.

**Role-Based Access Control (RBAC)** solutions allocate access permissions based on user roles and responsibilities. Users are granted access solely to resources and data pertinent to their specific roles, diminishing the chances of unauthorized access. For instance, **AWS Identity and Access Management (IAM)** empowers organizations to

define roles and policies for AWS resources. It guarantees that users and applications receive precise access privileges aligned with their tasks while adhering to the principle of least privilege.

In summary, User Access and Privilege Management solutions are the guardians of digital access. They regulate and protect access, preserving sensitive data and ensuring compliance with security policies. These solutions are indispensable for securing and upholding the integrity of an organization's network and data assets.

#### **2.2.4 Vulnerability Management tools**

Vulnerability Management solutions play a pivotal role in cybersecurity as vigilant protectors. They are responsible for identifying, evaluating, and mitigating vulnerabilities within an organization's digital infrastructure, bolstering security measures, and ensuring resilience.

**Vulnerability Scanning Solutions** act as diligent scouts on the lookout for digital vulnerabilities. They systematically inspect network devices, applications, and systems to pinpoint potential weaknesses, misconfigurations, or security gaps. One example of such a solution is **Qualys Vulnerability Management**. It conducts thorough vulnerability scans, pinpointing potential risks across the network. The tool classifies vulnerabilities by their severity, providing a prioritized list for remediation. This empowers organizations to proactively address security threats and uphold a robust security posture. **Nessus**, developed by Tenable, is another powerful Vulnerability Scanning Solution. It is known for its robust scanning capabilities and extensive vulnerability database. Nessus Professional conducts in-depth scans, detecting potential risks across a wide range of devices and applications. Its reports offer valuable insights for security teams to act upon promptly. **OpenVAS** is an open-source Vulnerability Scanning Solution known for its flexibility and community-driven development. It conducts scans to pinpoint vulnerabilities in networks and applications, providing detailed reports and recommendations for remediation. OpenVAS is a cost-effective option for organizations seeking reliable vulnerability scanning.

**Patch Management Solutions** are the digital safeguards entrusted with fortifying an organization's security. Their mission is to systematically address vulnerabilities within software and systems. They streamline the deployment of patches and updates, minimizing the time during which known vulnerabilities can be exploited. An example is **Ivanti Patch for Windows** which stands as a comprehensive Patch Management Solution, automating the deployment of patches and updates for Windows operating systems and various Microsoft software. It simplifies the patching process, ensuring swift application of critical security updates. Ivanti's solution empowers organizations to mitigate security risks effectively and maintain a robust security posture. **IBM BigFix** is a Patch Management Solution renowned for its versatility in managing and deploying patches across a wide spectrum of operating systems and applications. It offers real-time visibility into the patch status of devices within an organization's network, enabling administrators to efficiently prioritize and apply patches. IBM BigFix is instrumental in helping organizations uphold security and compliance. **WSUS (Windows Server Update Services)**, developed by Microsoft, is a widely embraced Patch Management Solution

tailored for Windows environments. It centralizes the management and distribution of Windows updates and patches. WSUS simplifies the task of keeping Windows-based systems current with the latest security fixes, thus fostering a more secure network environment.

In summary, Vulnerability Scanning Solutions and Patch Management Solutions are essential tools for identifying and addressing security weaknesses. These solutions enable organizations to proactively enhance their cybersecurity posture by uncovering vulnerabilities, prioritizing remediation efforts and maintaining the security and reliability of an organization's software and systems.

## **2.3 RELATED RESEARCH**

There are several research efforts in the context of device, network and user modeling and enumeration. Several research efforts related with the definition of security ontologies are found, although they may differ in their goal and scope. Depending on the investigated problem, each ontology might focus on subjects as specific as security entities like threats [3], vulnerabilities [4], threat agents [9], intrusion detection systems, attacks [10] and countermeasures [11], or as broad as security policies, network security and network management, information security management systems etc.

One of the most recent security ontologies presented in [12] emphasizes on information derived from existing OSCTI gathering and management platforms, which they focus on low level indicators of compromise (IOC). To bridge the existing gap of higher-level IOCs, a knowledge graph called SecurityKG is also presented in [12], which is essentially a system for automated OSCTI gathering and management. SecurityKG is capable of extracting information from semi-structured text reports using AI and NLP.

In [13] a security ontology that connects known security databases such as NVD [14], CVE, CWE and ATT&CK [6] is presented. An aggregate data graph called BRON is presented, which enables the bi-directional, relational path tracing within entities. BRON is then used to identify attack patterns, tactics, and techniques that exploit CVEs. Furthermore, BRON is able to support a hypothesis expressed in plaintext that refers to information that can be indexed through the data graph.

[15] delves deep into the realm of cyber threat intelligence, emphasizing the pivotal role of structured knowledge representation. The authors underscore the importance of automation in intelligence generation, consumption, and utilization. They advocate for ontologies, which serve as structured knowledge representations, to be at the forefront of this automation. However, they also highlight the challenges faced in the current landscape, such as ambiguity in defined concepts and the limited use of existing taxonomies.

One of the paper's hallmark contributions is the introduction of the "Cyber Threat Intelligence Model." This model is meticulously crafted to distinguish various types of information, encapsulating the essential elements of threats and attacks, namely the who, what, why, where, when, and how. Through this model, the authors aim to provide a clearer lens to view the vast and complex domain of cyber threat intelligence. The model

serves as a beacon, guiding efforts towards a more structured and comprehensive representation of threat intelligence.

In their research, the authors embarked on a rigorous evaluation of the existing taxonomies, sharing standards, and ontologies pertinent to cyber threat intelligence. Their findings were revealing, confirming a noticeable gap in the development of a comprehensive cyber threat intelligence ontology. Many of the existing efforts were found to be ambiguous, fragmented, and lacking interoperability. Concluding their research, the authors emphasize the pressing need for a holistic and unambiguous cyber threat intelligence ontology. They envision their proposed reference architecture, the Cyber Threat Intelligence Model, as a foundational blueprint that can steer the development of an ontology that's not only comprehensive but also modular, extensible, and adaptive to the ever-evolving cyber threat landscape.

[16] identifies a significant gap in the existing security enumerations, particularly the Common Platform Enumeration (CPE) and the Common Vulnerabilities and Exposures (CVE) enumeration, in addressing the complexities and interdependencies of components within Cyber-Physical Systems (CPS). The authors argue that while these enumerations are effective for traditional IT systems and vulnerabilities, they fall short in addressing the unique challenges posed by CPS, such as the integration of physical elements into computing and control processes. The paper emphasizes the need for a more comprehensive and usable enumeration system that can improve the overview of numerous and heterogeneous CPS components within an organization and can accommodate novel classes of CPS components.

To address the identified deficiencies, the authors propose models for CPS that aim to extend the current enumerations to cover comprehensiveness and usability. They introduce a conceptual meta model built upon the findings of the requirements phase, describing a formal structure and relationships between entities of CPE and CVE. The proposed model introduces new attributes while keeping the existing ones unchanged, ensuring compatibility with earlier versions. The extensions are grouped into four categories: technically exhaustive security enumerations, recursive security enumerations, application-oriented security enumerations, and coupled security enumerations. These extensions aim to streamline the representation of various components within CPS and address the system of systems concept inherent to CPS.

The authors implement their approach in a prototypical search engine to evaluate the findings, demonstrating the feasibility of the proposed extensions by showcasing key features of security enumerations for CPS. The paper concludes that the proposed extensions to CPE and CVE are crucial in making the security of CPS manageable and in fostering a better understanding of CPS. The enhancements are aimed at reducing complexity and improving the usability of security enumerations for users without specific domain knowledge, thereby contributing to the overall security management of Cyber-Physical Systems.

[17] begins by highlighting the importance of asset inventory in cybersecurity analysis and management. Given the large-scale, heterogeneous, and dynamic nature of modern information systems, manual inventory becomes time-consuming and resource intensive. The authors emphasize the need for continuous asset inventory due to the ever-

changing nature of system objects and topology. The primary contribution of the paper is the introduction of a technique for automated identification of system assets and the connections between them. This technique is built upon event correlation methods, focusing on linking system events to one another.

The authors present a two-stage technique for the automated identification of both physical and information assets within a system and the revelation of connections between them. The first stage, which was previously introduced at the 1st IFIP NTMS Workshop on "Cybersecurity on Hardware," is based on event correlation. By analyzing the volatility of object characteristics, the technique isolates associations of characteristics to specific information categories. The second stage focuses on determining the connections between objects, essential for identifying potential cyber-attack paths. This involves determining the hierarchy of objects and the relationships between them based on event types. Additionally, the paper discusses an automated approach to calculate the criticality of assets, evaluating an object's significance based on the total usage rates of characteristics.

To validate their approach, the authors conduct experiments demonstrating the applicability of the developed technique. The experiments showcase the method's capability to determine object properties and types effectively. The paper's contributions lie in its novel approach to automating asset inventory using event correlation, its comprehensive two-stage technique for identifying assets and their interconnections, and its focus on the criticality assessment of assets. The authors emphasize that their technique offers flexibility and compatibility with different data management systems, making it a valuable tool for organizations seeking to enhance their cybersecurity posture.

[18] underscores the vulnerabilities of critical infrastructures (CIs) like power grids, which link numerous physical components to the software systems controlling them. These systems are under constant threat from sophisticated cyberattacks. The authors emphasize the urgent need to bolster the cybersecurity of CIs through comprehensive system modeling and vulnerability analysis. This task is daunting due to the intricate data from interconnected physical and computational systems. The paper's primary contribution is its exploration of a comprehensive taxonomy designed to model these interconnections and the dependencies within complex CIs. This taxonomy bridges the knowledge gap between IT security and operational technology (OT) security.

The authors introduce an extensible taxonomy that captures the common semantics of both IT and OT entities, facilitating the convergence of IT/OT security. This taxonomy defines CI entities, their types, attributes, and security attributes. It also focuses on intra-dependencies within CIs, especially in the cyber and cyber-physical domains. To validate the applicability of the proposed taxonomy, the authors set up reference models for a CI, specifically the smart grid. These models, based on an in-depth literature review, reflect real infrastructure connections and support vulnerability-centered simulations. The paper also proposes a vulnerability assessment method that links the taxonomy with security repositories, enabling the identification and assessment of vulnerabilities for CIs.

The proposed taxonomy, instantiated reference models, and dependence-analysis deductive rules are implemented in a tool named ConceptBase. The authors have also published their instantiated models and code. The paper's contributions extend research



on enterprise modeling by covering the physical processes beneath the technology layer of traditional enterprise modeling languages. The authors have also defined vulnerability-centered attributes based on the Common Vulnerability Reporting Format (CVRF) framework, enhancing interoperability with other security tools. The paper concludes by highlighting the significance of the proposed taxonomy in describing all layers of power-grid systems, from physical components to software applications.

## 3 Methodology – Architecture

### 3.1 CYBER RANGE DEVELOPMENT

Within this critical subsection, we delve into the foundational aspect of our methodology of creating a controlled and secure environment that serves as the crucible for comprehensive cybersecurity testing and training.

Our overarching objective is to construct an adaptable, cutting-edge platform where an array of simulated cyber threats can be scrutinized, providing a robust framework for assessing asset inventories in real-world scenarios. By orchestrating virtualized networks, diverse attack vectors, and state-of-the-art defensive measures, we aim to provide a fertile ground for cybersecurity professionals to enhance their skills. Furthermore, this Cyber Range Development forms the cornerstone of our methodology, promising to yield invaluable insights into the intricate landscape of IT system security.

To establish our Cyber Range, we have meticulously assembled a set of essential components that serve as the backbone for extracting comprehensive information from the systems, employing a combination of log analysis and custom scripts. The key components include:

**Active Directory Environment:** At the core of our Cyber Range, we've set up an Active Directory environment. This robust foundation facilitates user management, authentication, and access control, enabling us to simulate real-world scenarios with precision.

**Web Server & SQL Server:** To emulate the diverse ecosystem of IT systems, we've deployed a Web Server and an SQL Server. These components enable us to mimic web-based applications and databases, expanding the scope of our cybersecurity testing.

**Firewall (utilized as IDS):** Our Firewall serves a dual purpose as an Intrusion Detection System (IDS). It functions as a sentinel, actively monitoring network traffic for suspicious patterns and potential threats, further enhancing our Cyber Range's realism.

**Windows Client:** Incorporating a Windows Client into our setup allows us to replicate the behavior of Windows-based systems. This client serves as a vital node for assessing security measures and gathering critical telemetry data.

**Linux Client:** Diversifying our environment, we've integrated a Linux Client. This addition broadens the spectrum of operating systems under evaluation, enabling us to address the nuances of Linux-based cybersecurity.

**SIEM Technology:** As a keystone in our arsenal, we've integrated Security Information and Event Management (SIEM) technology. SIEM tools empower us to

centralize and analyze security data from various sources, enhancing our ability to detect and respond to potential threats.

Additionally, we've taken proactive measures to bolster our data collection efforts. **Advanced auditing** [19] [20] is enabled on all Domain Controllers, ensuring a granular level of event tracking. Furthermore, **Sysmon** [21] agents have been deployed across all Windows machines, enhancing telemetry extraction, and augmenting our threat visibility.

This comprehensive infrastructure, comprising the aforementioned components and advanced data extraction techniques, forms the heart of our Cyber Range Development, poised to provide invaluable insights into the intricacies of IT system security.

As an addition to our Cyber Range Development, we've implemented **Nested Virtualization**, an approach that enhances both portability and ease of management. This innovative technique empowers us to streamline the setup and maintenance of our environment, offering a wealth of benefits. It enables us to create snapshots of the entire infrastructure and its components effortlessly. By capturing a snapshot of the host Virtual Machine, we can preserve the entire configuration, state, and interactions of our Cyber Range. We gain the remarkable advantage of exporting the entire infrastructure as a cohesive unit. This not only simplifies backup procedures but also facilitates the effortless transfer of the entire Cyber Range to different physical hosts or environments. Importantly, it does not restrict us from exporting individual components of the infrastructure independently. This flexibility allows us to selectively package and reuse specific elements, ensuring that our Cyber Range remains adaptable to various testing scenarios.

### **3.1.1 Active Directory Environment with IIS & SQL Server**

In our endeavor to construct the initial components, specifically the Active Directory Environment and the Web Server & SQL Server, we have strategically utilized the GOAD environment. GOAD, available as an open-source resource at <https://github.com/Orange-Cyberdefense/GOAD> [22], represents an Active Directory environment that comes pre-loaded with numerous vulnerable elements, including an IIS Web Server and a SQL Server.

This astute choice allows us to tap into a pre-configured environment designed to cater to our cybersecurity testing requirements. The versatility and richness of vulnerabilities within the GOAD environment align seamlessly with our mission to replicate authentic scenarios and meticulously examine the robustness of our asset inventories.

### **3.1.2 Firewall with IDS capabilities**

To enhance the capabilities of our Cyber Range, we've implemented a robust network security solution featuring PfSense with Suricata. In this setup, PfSense [23] serves as our primary router and firewall, bolstered with Intrusion Detection System (IDS) functionality provided by Suricata.

This strategic choice equips us with a potent defensive layer to monitor network traffic, detect anomalies, and safeguard against potential threats. The combination of

pfSense and Suricata ensures that our Cyber Range remains fortified and resilient in the face of evolving cybersecurity challenges.

Furthermore, to enhance accessibility and expand the utility of our Cyber Range, we've configured an OpenVPN Server within pfSense. This addition allows multiple users to securely access and interact with our Cyber Range from remote locations. It offers a versatile and secure means for collaborative testing and training, making our environment more adaptable and user-friendly.

### **3.1.3 Windows and Linux Client**

Incorporating both a Linux and a Windows client into our Cyber Range allows us to faithfully emulate a real-world scenario where the Windows client [24] serves as a representative user's workstation, while the Linux machine [25] acts as a server hosting a critical application.

This duality in our client infrastructure mirrors the diverse IT ecosystem found in practical settings. The Windows client, representing the user's workstation, enables us to simulate typical end-user interactions, assess security measures, and gather crucial telemetry data specific to this operating system.

On the other hand, the Linux client, serving as a server hosting an application, offers a unique perspective into the security challenges and intricacies of managing a vital server in a production environment. It allows us to examine application-specific vulnerabilities and conduct in-depth assessments of server-side security measures.

### **3.1.4 SIEM Technologies**

To bolster our detection capabilities and streamline our cybersecurity efforts, we've integrated Security Information and Event Management (SIEM) technology into our Cyber Range. This SIEM integration facilitates the aggregation of telemetry data and alerts from all the components within our infrastructure, ensuring comprehensive monitoring and analysis.

In our quest for enhanced threat visibility and event correlation, we've opted for two powerful SIEM solutions: Wazuh and Azure Sentinel. These SIEM platforms work in tandem to collect telemetry data and alerts generated by all the elements within our infrastructure, regardless of their nature or location.

By centralizing this wealth of information, we establish a single source of truth for security monitoring and incident response. This consolidation of data empowers us to detect and respond to potential threats swiftly and effectively, while also providing invaluable insights for ongoing cybersecurity analysis and improvement.

SIEM technologies can also assist us in the identification of specific threats against distinct components at real time. This will support the work that will be performed in the next subsections, since the next steps involve the identification of uncatalogued assets with no known threats. By identifying threats at real time for unknown components, threat profiles can be built, and security controls can be suggested towards their mitigation.

In addition to SIEM installation within our range, we will perform crucial configuration changes on the Windows machines in our environment.

To obtain real-time telemetry data from our Windows machines, we will add **Sysmon** agents. Sysmon, short for System Monitor, operates as a Windows system service and device driver. Once it's installed on a system, it remains active even after system reboots. Its primary function is to observe and record system activities in the Windows event log. Sysmon offers in-depth insights into activities such as process initiations, network connections, and modifications to file creation timestamps.

Additionally, on the three domain controllers within our environment, we will enable enhanced logging. This action is crucial for capturing detailed information related to Active Directory attacks. By enabling enhanced logging, we ensure that our SIEM system receives comprehensive data, enabling us to identify and respond to threats targeting our Active Directory infrastructure effectively.

## **3.2 MODULAR ASSET MODELING & CPE EXTRACTION**

In this section, we delve into the foundational aspects of our asset inventorying methodology: Modular Asset Modeling and Automated Common Platform Enumeration (CPE) Management. Our focus here is on understanding the diverse layers of a device's surface that we are examining as part of our experiment. These layers encompass the hardware, operating system (OS), applications, and services, each contributing unique attributes to our comprehensive asset inventory.

### **3.2.1 Why Harvesting Assets in CPE Format is Crucial**

In today's increasingly complex and dynamic IT landscapes, asset management is a fundamental aspect of cybersecurity and risk management. The adoption of Common Platform Enumeration (CPE) [1] provides a standardized approach to asset management that brings several key benefits to organizations.

CPE offers a consistent and structured method for identifying and categorizing assets, including hardware, software, and services. This standardization simplifies the management of assets across diverse environments. It enables efficient vulnerability management by associating assets with known vulnerabilities, helping organizations quickly pinpoint assets that require patching or updates based on CPE identifiers.

CPE assists in building a comprehensive inventory of an organization's assets, including details like hardware specifications, software versions, and more. This inventory helps organizations keep track of changes, updates, and configurations over time.

Moreover, CPE allows organizations to assess the risk associated with each asset by considering its attributes. This information aids in prioritizing security measures based on the criticality of assets. Additionally, compliance reporting becomes more straightforward with CPE, as it provides a structured way to represent asset information. Organizations can efficiently demonstrate compliance with regulatory requirements.

While CPE is a valuable asset management tool, it's important to note that not all assets are included in the CPE catalog. This limitation is due to the ever-evolving nature of technology and the continuous introduction of new components and software.

However, this should not deter organizations from effectively managing and securing these assets.

Instead of waiting for possible vulnerabilities to be disclosed by the vendor or a third party and uploaded to NIST's NVD, organizations can proactively map assets not enumerated in the CPE catalog. By directly connecting these assets to potential threats, organizations can better understand and mitigate risks associated with their entire asset landscape.

To address the challenge of uncatalogued assets, we propose a proactive methodology - Automated Penetration Testing. This methodology involves deploying automated penetration agents to identify and assess assets that may not yet be in the CPE catalog. While this approach may not result in the discovery of CVEs or known vulnerabilities, it serves an essential purpose.

Automated penetration testing verifies the security posture of assets, cataloged or uncataloged, by simulating real-world attacks and testing the effectiveness of security measures. This testing can validate the use of specific ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) [10] techniques against assets. It helps organizations ensure that their defenses can withstand various attack scenarios, whether the target is in the CPE catalog or not.

An additional benefit of this approach is the automatic creation of asset and vulnerability profiles from the user end. Organizations can build detailed profiles based on real-world testing, reducing reliance on vendors for such information.

Furthermore, it's crucial to highlight that this automatic identification process can actively contribute to extending the existing CPE catalog. When agents identify new assets and their associated CPE attributes, this information can be collected and shared within the cybersecurity community.

By linking this information with verified attack techniques through automated penetration testing, organizations can provide valuable insights. These insights can then be uploaded to the CPE catalog, expanding its coverage, and ensuring that the community benefits from collective efforts.

Incorporating assets into the CPE catalog remains a critical component of effective asset management and cybersecurity. However, organizations should also be prepared to manage and secure assets that have not yet been included in the catalog. Our proposed methodology of Automated Penetration Testing not only helps organizations address this challenge but also actively contributes to extending the CPE catalog through community collaboration. By combining standardized asset management with proactive security testing and knowledge sharing, organizations can stay ahead in the ever-changing landscape of cybersecurity and drive continuous improvement in asset identification and protection.

### **3.2.2 Defining Device Layers**

In our asset inventorying approach, we begin by categorizing a device into distinct layers, each representing a crucial facet of its composition and functionality. These layers encompass:

**Hardware Layer:** This encompasses the physical components of the device, including its processors, memory, storage devices, network interfaces, and other tangible elements.

**Operating System (OS) Layer:** Here, we focus on the software environment that manages hardware resources and facilitates the execution of applications. Attributes in this layer include OS type, version, patch level, and configuration settings.

**Application Layer:** Within this layer, we examine the software applications installed and running on the device. Attributes cover application names, versions, dependencies, and configurations.

**Services Layer:** This layer pertains to the network services and processes that the device offers. Attributes encompass service names, ports, protocols, and associated configurations.

### 3.2.3 Enumerating Assets Within Each Layer

To construct a comprehensive asset inventory, we employ specialized tools tailored to each layer:

**Hardware Enumeration Tools:** In the Hardware Layer, we aim to gather comprehensive details about the physical components of the device. This includes critical information such as CPU specifications, RAM capacity, disk drives, and network adapters. To accomplish this, we employ a combination of hardware inventory scanners and system information utilities tailored to the respective operating systems.

On Windows systems, we harness the power of the PowerShell command **Get-ComputerInfo**. This versatile tool allows us to query hardware information with precision. It provides insights into essential components such as CPU specifications, RAM capacity, storage configurations, and network interfaces.

In Linux environments, we turn to the robust and versatile **hwinfo** command-line tool. **hwinfo** is adept at providing an exhaustive breakdown of hardware particulars, including CPU specifications, RAM capacity, storage characteristics, and intricate details about network adapters.

**OS Enumeration Tools:** In the OS Layer, our primary objective is to extract critical data about the operating system that manages the device. This includes information regarding the OS type, version, installed updates, and security settings. To achieve this, we employ a combination of system information utilities, OS-specific commands, and network scanning tools tailored to the specific operating system.

On Windows systems, we rely on the built-in tool, **systeminfo**. This utility allows us to extract a wealth of information about the operating system, including its type, version, installed updates, and various security settings. The data obtained through **systeminfo** provides a comprehensive snapshot of the Windows OS, enabling us to assess its configuration and vulnerabilities effectively.

In Linux environments, we turn to the versatile **uname** command. **uname** not only provides basic information about the OS, such as the kernel version but also helps us gather LSB (Linux Standard Base) data and distribution-specific details. By leveraging **uname**, we ensure that we capture essential OS characteristics specific to Linux, facilitating a precise assessment of the OS layer within our Cyber Range.

**Application Enumeration Tools:** In the Application Layer, our focus is on identifying and documenting the software applications installed on the device. This includes gathering information about application names, versions, dependencies, and relevant configuration data. To accomplish this, we utilize a combination of application inventory tools, package managers, and registry inspections, customized to the specific operating system.

For Windows-based application enumeration, we employ a two-fold approach:

- **PowerShell scripts:** Custom PowerShell scripts are utilized to extract application information from the Windows registry keys, providing details about installed applications, their versions, and configurations.
- **Get-WmiObject cmdlet:** This versatile PowerShell cmdlet further enhances our ability to collect application data, offering insights into software dependencies and attributes.

In Linux environments, the approach varies based on the distribution. For Debian-based systems (e.g., Ubuntu), we turn to the **dpkg** package manager. It allows us to enumerate installed software packages, their versions, and dependencies. For Red Hat-based systems (e.g., CentOS), we utilize the **rpm** package manager to perform a similar task, providing a comprehensive list of installed software packages.

**Services Enumeration Tools:** In the Services Layer, our goal is to identify and catalog the services and processes active on the device, along with their related configurations. To accomplish this, we employ specific methods adapted to each operating system.

Within Windows environments, we rely on the versatile PowerShell cmdlet **Get-Service**. This powerful tool allows us to enumerate and gather information about the services running on the system. We can extract essential details such as service names, statuses, and relevant configurations. This approach provides an encompassing view of the services present within the Windows OS.

For Linux-based systems, we utilize the **systemctl** command, which facilitates the management and inquiry of services within the systemd initialization system—a commonly used framework in modern Linux distributions. By employing **systemctl**, we can accurately identify services, ascertain their statuses, and access configuration specifics. This approach ensures a comprehensive inventory of services within Linux environments.

### 3.2.4 Creating CPEs from Extracted Data

For the creation of CPE representations based on the extracted information, we've devised a structured approach that involves the use of CSV files and custom scripts. This systematic process ensures a unified format for expressing assets and facilitates further analysis and documentation.

We initiate the process by creating CSV files that serve as structured data containers. These files are organized to hold the information extracted from the devices, making it easy to manage and manipulate the data efficiently.

Our custom scripts are designed to iterate through the extracted data, applying necessary formatting to align with the Common Platform Enumeration (CPE) standard.



This includes organizing data into specific fields and ensuring consistency in representation.

Leveraging the formatted data, our scripts generate CPE Uniform Resource Identifiers (URIs) based on the extracted information. CPE URIs adhere to a standardized format, allowing for a uniform representation of assets across the board.

The CPE URIs, now generated, are appended to separate CSV files dedicated to CPE representation. These files serve as a repository of CPEs, making it convenient for users to access and utilize CPE information for various cybersecurity and documentation purposes.

This automated process streamlines the creation of CPE representations, ensuring consistency and accuracy in expressing assets. Additionally, it provides a practical means of organizing and storing asset data, ultimately enhancing our ability to assess and manage assets within the Cyber Range.

### **3.3 SECURITY CONTROL INTEGRATION**

In this section, we outline our comprehensive approach to security control integration within our Cyber Range. We recognize the importance of a proactive and structured strategy to safeguard our assets and mitigate security threats effectively. Our approach is subdivided into four key subsections, each addressing critical aspects of security control integration.

#### **3.3.1 Automated Penetration Testing Agents**

In this second subsection, we explore the deployment and vital role of automated penetration testing agents within our security control integration strategy. These agents are fundamental for identifying and confirming active threats targeting our Cyber Range assets.

Our proactive approach to security control integration revolves around deploying specialized automated penetration testing agents across our Cyber Range infrastructure. These agents are purpose-built to authentically simulate real-world attacks, mirroring the tactics, techniques, and procedures (TTPs) utilized by known threat actors and malicious entities.

Automated agents excel at emulating realistic threats. They faithfully replicate the TTPs commonly employed by known threat actors and malicious entities, covering a wide array of attack scenarios, including those linked to advanced persistent threats (APTs) and prevalent cybercriminal methods.

To facilitate efficient management, we can have centralized control and oversight over these agents via a dedicated console. This central hub empowers us to vigilantly monitor, regulate, and assess the progress of penetration testing activities. It also serves as a hub for aggregating and scrutinizing test results.

Results stemming from these tests, encompassing particulars of successful attacks and their repercussions, can be easily exported for thorough analysis and documentation.

This information is instrumental for gaining valuable insights into our security posture, identifying vulnerabilities, and formulating targeted remediation strategies.

This structured approach empowers us to methodically evaluate the resilience of our defenses against a spectrum of known threats and attack vectors. By faithfully replicating known TTPs and malicious behaviors, we ensure that our security controls are robust and proficient in countering genuine threats.

### 3.3.2 Exploitable Threats in MITRE ATTACK Format

In this subsection, we explore the systematic process of categorizing and documenting exploitable threats using the MITRE ATT&CK framework. This approach not only aids in effective threat identification but also facilitates the alignment of threats with known MITRE DEFEND [11] techniques, bridging the gap between threat recognition and mitigation.

Our approach to threat analysis relies on the MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework, a well-structured methodology for characterizing the behaviors and tactics employed by threat actors.

We commence by identifying threats within our Cyber Range, drawing from the results of the attacks executed by our penetration testing agents. These threats may encompass a range of tactics and techniques, each potentially indicative of an adversarial activity.

Once threats are identified, we categorize and describe them using the MITRE ATT&CK framework. This framework, with its comprehensive ATT&CK Matrix, offers a rich vocabulary for characterizing the tactics and techniques commonly utilized by threat actors.

To effectively address the identified threats, we leverage the MITRE DEFEND techniques, which are directly linked to the MITRE ATT&CK framework. This linkage ensures that our mitigation measures are precisely aligned with the specific tactics and techniques employed by threat actors.

We align our threat profiles with corresponding MITRE DEFEND techniques known to counteract the specific tactics and techniques employed by threat actors. This strategic mapping ensures that our response measures are well-matched to the nature of the threat.

Our approach emphasizes continuous monitoring of the threat landscape to ensure that our mitigation measures remain effective. If new threats emerge, we adapt and refine our defense strategies accordingly.

This integration of MITRE ATT&CK and DEFEND into our threat analysis and mitigation process offers several notable benefits:

**Standardized Classification:** We maintain a standardized and structured approach for classifying and describing threats, enhancing clarity and communication within our security team.

**Effective Mitigation:** The direct mapping of threats to MITRE DEFEND techniques ensures that our mitigation measures are precisely tailored to counter the specific tactics and techniques employed by threat actors.

**Comprehensive Defense:** This approach enables a holistic and comprehensive defense strategy, enabling us to systematically address threats and vulnerabilities.

**Threat Intelligence Integration:** MITRE ATT&CK and DEFEND are continuously updated with threat intelligence, ensuring that our security controls and mitigation measures remain up to date.

By adopting this approach, we establish a robust and structured process for threat identification, classification, and mitigation. This enhances our ability to address threats systematically, strengthens our Cyber Range's overall security posture, and reinforces our readiness to respond to emerging challenges.

### 3.3.3 Building Custom Threat Profiles

To fully leverage the research potential of our CPE asset inventory methodology and automated penetration testing agents, we have devised a comprehensive strategy. We aim to combine the insights from both processes to construct detailed threat profiles. These profiles provide a dynamic view of our security landscape and help us better understand potential threats.

Our approach involves merging the data from CPE extraction, which provides an extensive overview of known and unknown components in our computing environment, with the information generated by automated agents during penetration testing. This combination allows us to gain a comprehensive perspective on potential weaknesses. By cross-referencing the CPE data with the attack techniques and targeted applications identified by our automated agents, we can proactively address security challenges and enhance our incident response capabilities.

In summary, our method of combining CPE asset inventory and automated penetration testing output allows us to create flexible threat profiles that adapt to emerging threats. This proactive stance, supported by empirical data and ongoing monitoring, plays a pivotal role in strengthening our cybersecurity posture and fortifying our computing infrastructure against evolving threats in a research-driven context.

### 3.3.4 Detection Engineering with Microsoft Sentinel

In this phase of our security control integration strategy, we harness the insights obtained from previous efforts, particularly focusing on successful attacks. The goal is to proactively establish security controls by creating custom queries and analytic rules within **Microsoft Sentinel**, our robust Security Information and Event Management (SIEM) platform.

This stage involves leveraging the data gathered from successful attacks and penetration testing. This data provides valuable intelligence about the tactics, techniques, and procedures (TTPs) utilized by threat actors in compromising our infrastructure.

Custom queries are crafted based on this attack data, targeting patterns, anomalies, and indicators of compromise (IoCs) observed during successful attacks. These queries enable us to pinpoint suspicious activities and potential threats within our Cyber Range.

Simultaneously, we develop analytic rules to automate the detection of specific threat scenarios. These rules incorporate advanced analytics and machine learning to identify threats in real-time, trigger alerts, and initiate incident response procedures. Continuous log analysis of network, system, and application logs within the Microsoft Sentinel ecosystem is fundamental to our approach. This ensures our vigilance in monitoring and detecting potential security incidents.

The implementation of detection engineering within Microsoft Sentinel offers numerous benefits. It enables proactive threat identification, reducing response time, while also minimizing false positives by focusing on relevant attack patterns and IoCs. Furthermore, it ensures that our security controls remain adaptive in the face of evolving threats, with continuous integration of real-time threat intelligence.

This phase represents an important milestone in our security control integration strategy. It transforms insights derived from previous phases into actionable security controls, fortifying our defenses, and enabling real-time threat detection within our Cyber Range.

In this comprehensive section on "Methodology & Architecture" for asset inventorying in IT systems, we have meticulously detailed the various components and processes that constitute a robust cybersecurity framework within our Cyber Range. Each aspect has been carefully considered to ensure the highest level of security and threat detection.

**Cyber Range Development:** The establishment of our Cyber Range serves as the foundation, providing a controlled environment for security testing and threat emulation. Utilizing nested virtualization enhances portability and management.

**Modular Asset Modeling Framework:** We have created a systematic approach to asset modeling, which encompasses hardware, operating systems, applications, and services. This framework is pivotal for effective asset inventorying.

**Automated CPE Management Tool:** Our automated tool extracts Common Platform Enumeration (CPE) information from asset data, allowing us to build a standardized and structured inventory.

**Security Control Integration:** By deploying automated penetration testing agents, we proactively identify threats and align them with MITRE ATT&CK and DEFEND techniques for comprehensive threat analysis and mitigation. Detection engineering in Microsoft Sentinel adds an extra layer of protection.

This holistic approach ensures that our Cyber Range remains resilient and adaptable in the face of evolving cybersecurity challenges. Our commitment to structured asset inventorying and threat detection empowers us to proactively defend our IT systems and respond effectively to emerging threats. With a well-rounded methodology and robust architecture, we are well-equipped to safeguard our IT assets in an ever-changing cybersecurity landscape.

## 4 Implementation

In this section, we shift from theory to practice, detailing the practical execution of our cybersecurity framework. This section is divided into three main parts: Cyber Range Development, Modular Asset Modeling & CPE Extraction, and Security Control Integration.

Our aim is to offer a clear account of how we put our cybersecurity strategy into action. We'll explain the technical steps taken, tools used, and outcomes achieved, providing a practical guide for cybersecurity professionals and researchers.

From creating a controlled Cyber Range environment to building a modular asset modeling framework and integrating security controls, this section highlights our efforts to strengthen cybersecurity in a rapidly evolving threat landscape.

### 4.1 CYBER RANGE DEVELOPMENT

In the Cyber Range Development subsection, we dive into the initial steps of setting up our cybersecurity lab environment. To kickstart this setup, our fundamental requirement is an Ubuntu Linux machine running version 20.04. This Linux host serves as the foundation for our lab environment, which will encompass multiple interconnected components.

To accommodate the complexity of this lab environment and ensure optimal performance, we've allocated a substantial 32 GB of RAM to the host virtual machine (VM). Additionally, we've maximized CPU resources and reserved 250 GB of free storage space. These resource allocations are essential to support the diverse range of components and functionalities within our Cyber Range.

#### 4.1.1 Active Directory Environment with IIS & SQL Server

In this phase, we initiate the setup of our cybersecurity lab environment, beginning with the installation of the GOAD (Game of Active Directory) environment, which serves as our Active Directory infrastructure along with IIS (Internet Information Services) and SQL Server components.

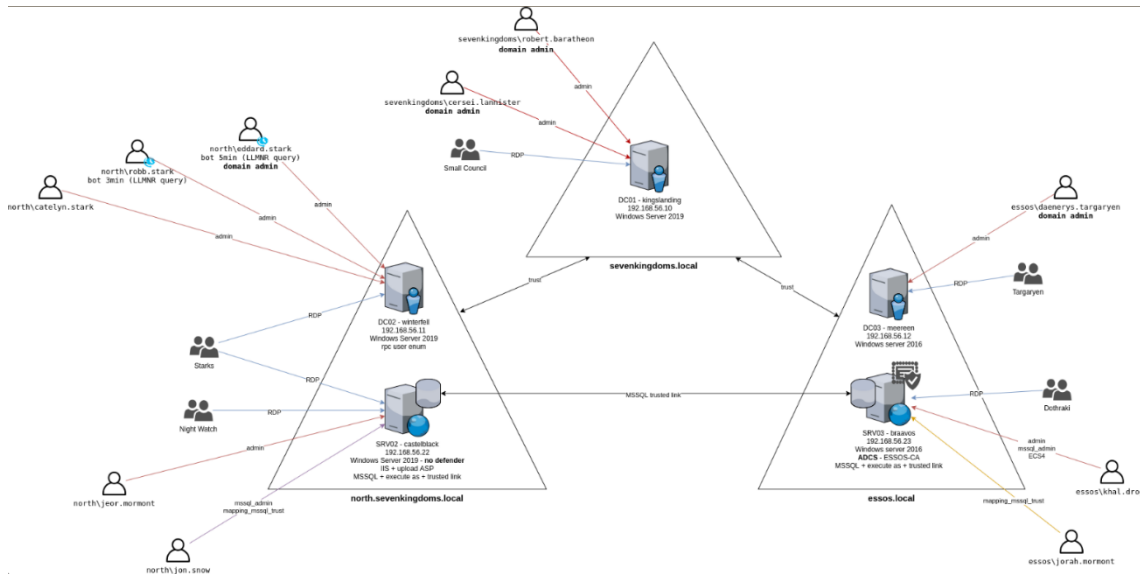


Figure 2: GOAD (Game of Active Directory) Environment

We start by ensuring that our Ubuntu Linux machine, version 20.04, meets specific hardware requirements, including 32 GB of RAM, ample CPU resources, and 250 GB of free storage space. We install VirtualBox on our Linux Virtual Machine using the command `sudo apt install VirtualBox`. This virtualization software is essential for hosting our lab environment.

We access the GOAD environment from its GitHub repository at <https://github.com/Orange-Cyberdefense/GOAD> and download the provided ZIP file.

We begin by installing the Vagrant tool, which assists in managing virtualized environments. The installation involves these commands:

- `wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg`
- `echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list`
- `sudo apt update && sudo apt install vagrant`

Optionally, we can install the Vagrant VirtualBox Guest Additions plugin using `vagrant plugin install vagrant-vbguest` for enhanced compatibility.

Following the completion of the VM provisioning phase, we observe five Virtual Machines running in VirtualBox, each equipped with two network adapters:

- A NAT adapter, primarily used for configuring the Active Directory environment through Ansible. This adapter can be disabled after setup.
- A host-only adapter, facilitating connectivity and functioning as one of our lab networks.

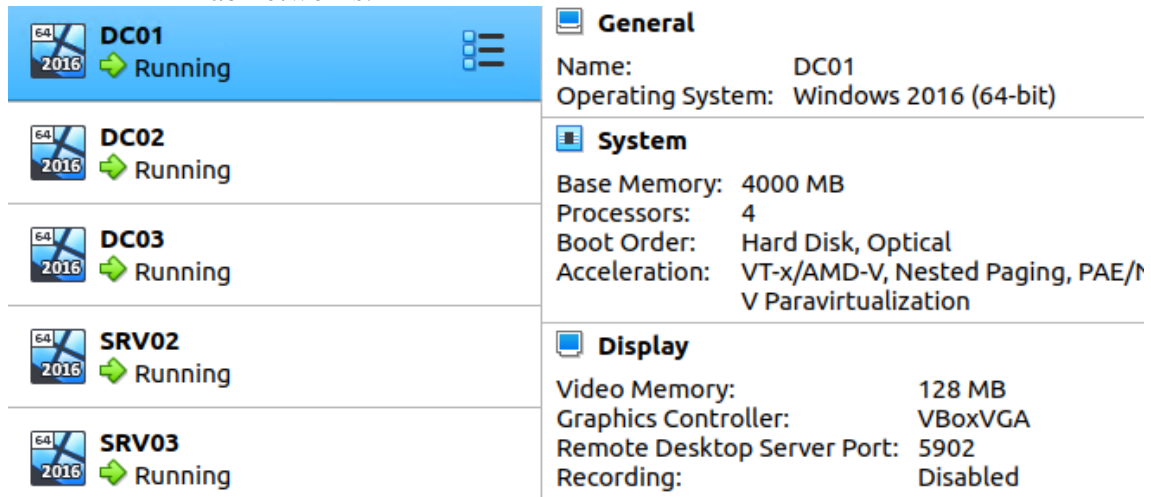


Figure 3: The first 5 Virtual machines of the Lab environment

The subsequent phase involves the configuration of our VMs using Ansible, an automation tool renowned for its efficiency. We initiate this process by launching specific commands within the GOAD directory our Linux Virtual Machine.

To prepare the environment for Ansible we install Git using `sudo apt install git`, clone the GOAD repository using Git and afterwards the installation of Python 3.8 virtual environment (`python3.8-venv`) takes place.

- **`sudo apt install git`**
- **`git clone git@github.com:Orange-Cyberdefense/GOAD.git`**
- **`cd GOAD/ansible`**
- **`sudo apt install python3.8-venv`**
- **`python3.8 -m virtualenv .venv`**

We activate the virtual environment using **`source .venv/bin/activate`**.

Within the virtual environment, we install Ansible and `pywinrm`:

- **`python3 -m pip install --upgrade pip`**
- **`python3 -m pip install ansible-core==2.12.6`**
- **`python3 -m pip install pywinrm`**

We ensure all the Ansible-Galaxy requirements are met using:

- **`ansible-galaxy install -r requirements.yml`**

Finally, we execute the Ansible provisioning process with the following command, provided that the VMs are in a running state (achieved through **`vagrant up`**):

- **`ansible-playbook -i ../ad/sevenkingdoms.local/inventory main.yml`**

Should any errors arise, in most cases, rerunning the main playbook can resolve issues.

For added safety, an alternative approach involves setting up the Active Directory lab step by step, issuing separate commands for each configuration aspect, thus ensuring greater control and precision.

- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory build.yml**  
# Install stuff and prepare vm
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory ad-servers.yml**  
# create main domains, child domain and enroll servers
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory ad-trusts.yml**  
# create the trust relationships
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory ad-data.yml**  
# import the ad datas : users/groups...
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory servers.yml**  
# Install IIS and MSSQL
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory ad-relations.yml**  
# set the rights and the group domains relations
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory adcs.yml**  
# Install ADCS on essos
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory ad-acl.yml**  
# set the ACE/ACL
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory security.yml**  
# Configure some securities (adjust av enable/disable)
- **ansible-playbook -i ../ad/sevenkingdoms.local/inventory vulnerabilities.yml**  
# Configure some vulnerabilities

Upon successful installation completion, we have the option to disable the first Network Adapter on the VMs, retaining only the host-only adapter for inter-VM connectivity. Subsequent networking adjustments will be made as we proceed with the firewall setup, redirecting traffic through it.

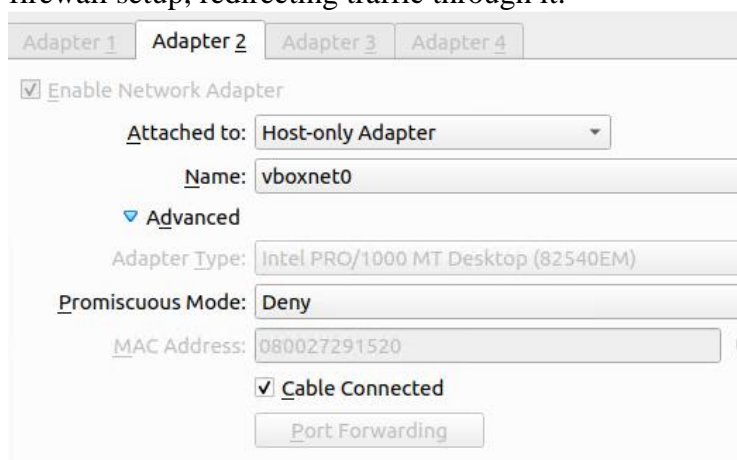


Figure 4: Network Adapter for VM interconnection.



### 4.1.2 Firewall with IDS capabilities

In our previous section, we established an Active Directory environment, creating a network within our host VM that serves as the first LAN segment. However, to avoid routing our Active Directory environment through our host machine and instead route it through the PSense firewall, we need to reconfigure the network interfaces of all machines created in the previous step to utilize an internal network.

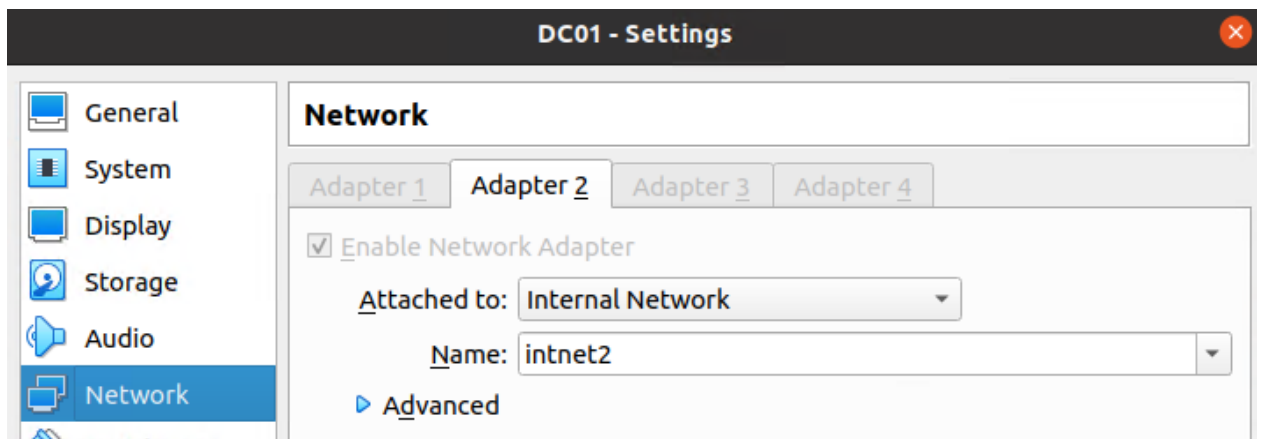


Figure 5: First LAN of Cyber Range's Network

Our entire network will be divided into three distinct subnets, comprising two LAN segments and one WAN segment that provides internet connectivity. To achieve this, we must enable three interfaces in the virtual machine hosting our pfSense firewall.

#### **Setting up Virtual Machine**

To commence, we download the PfSense firewall installation media from <https://www.pfsense.org/download/>.

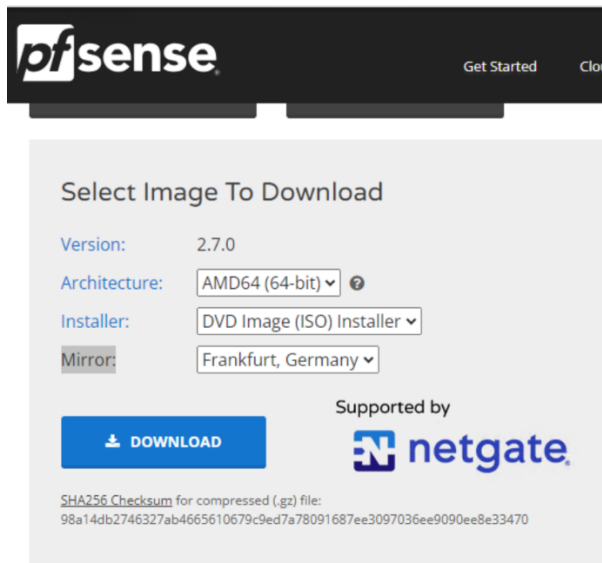


Figure 6: *PfSense installation media download*

In the virtualization software, we create a new virtual machine for our firewall. In the storage settings, we attach the downloaded ISO as the installation media. Once the installation is complete, we can remove the ISO image.

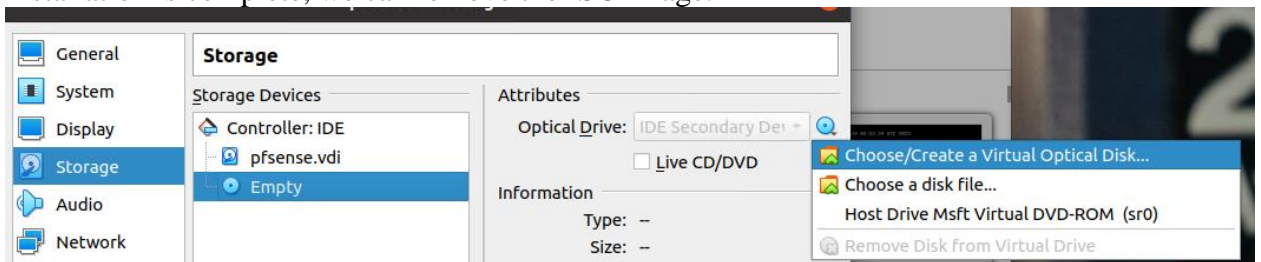


Figure 7: *Adding installation media to the newly created VM.*

Depending on the specific capabilities required for our firewall, we allocate the necessary CPU and RAM resources. For our lab, 2 virtual CPUs and 1024 MB of RAM are more than sufficient to meet our testing needs.

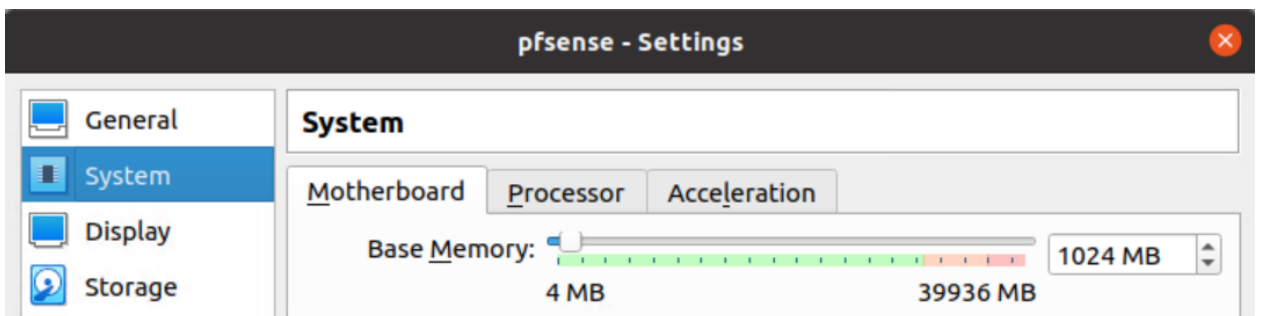


Figure 8: *RAM resources of Firewall*

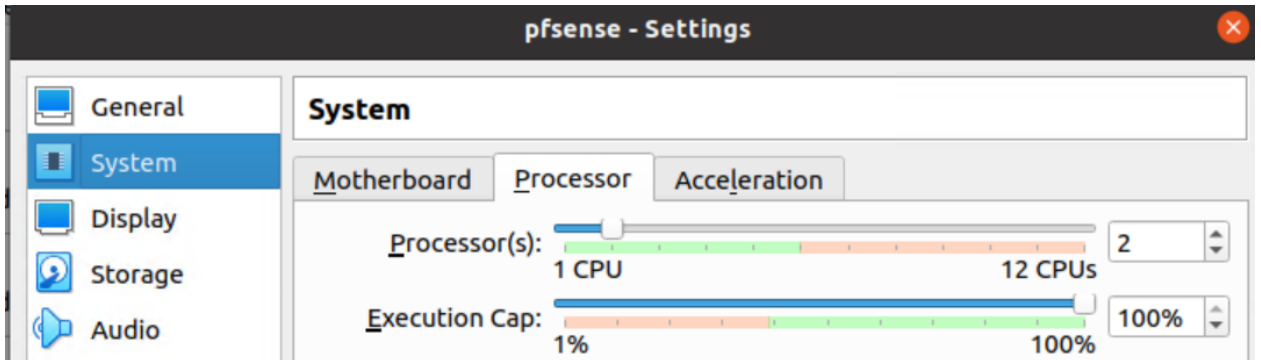


Figure 9: CPU resources of Firewall

One of the most critical configurations of the virtual machine is the setup of its network interfaces in the Network section. We need to define three interfaces, each assigned to a different VirtualBox network:

The first interface corresponds to the first LAN (where the GOAD environment resides).

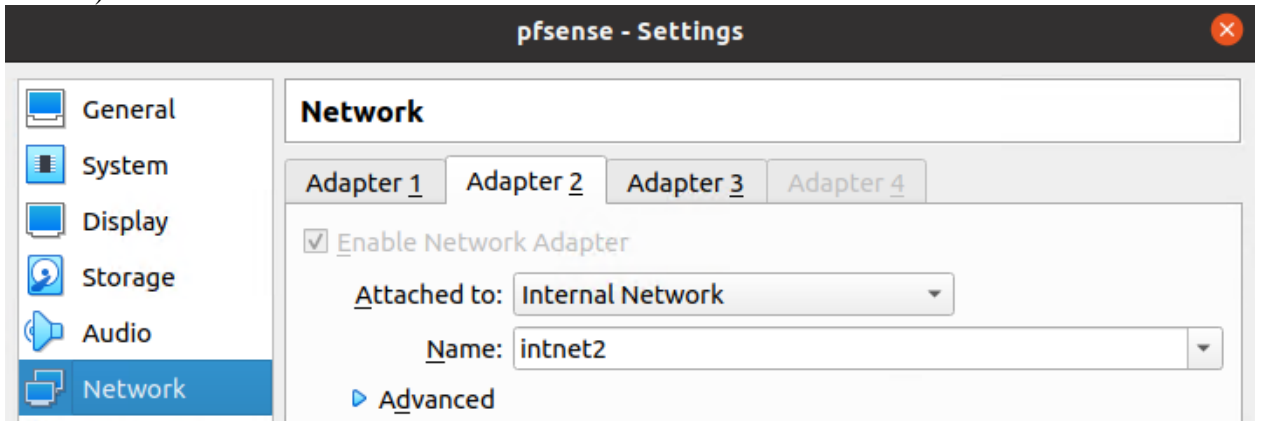


Figure 10: LAN network

The second interface is designated for the second LAN (where our Linux machine and WAZUH machine will be located).

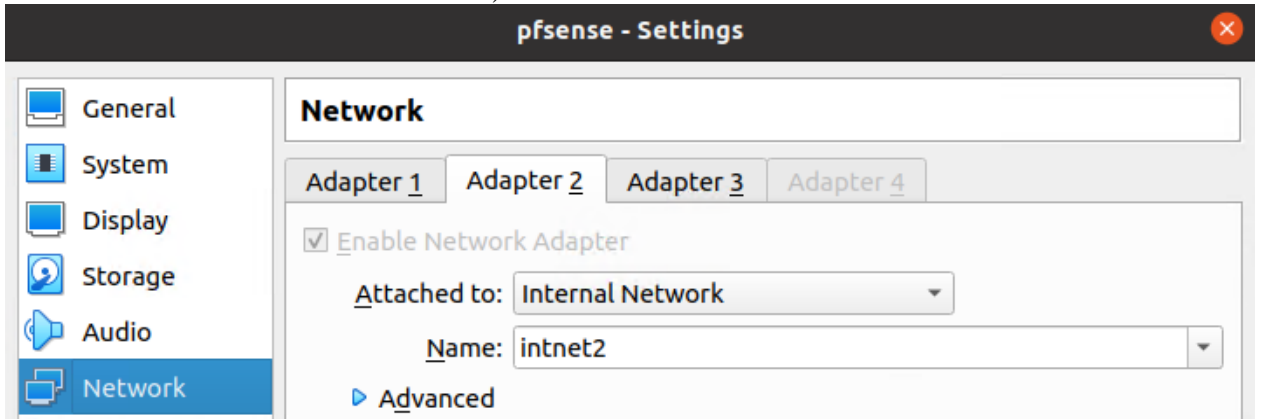


Figure 11: OPT1 Network

The third interface can be configured as a NAT interface, which will handle NAT translation of traffic to the host machine and provide internet connectivity.

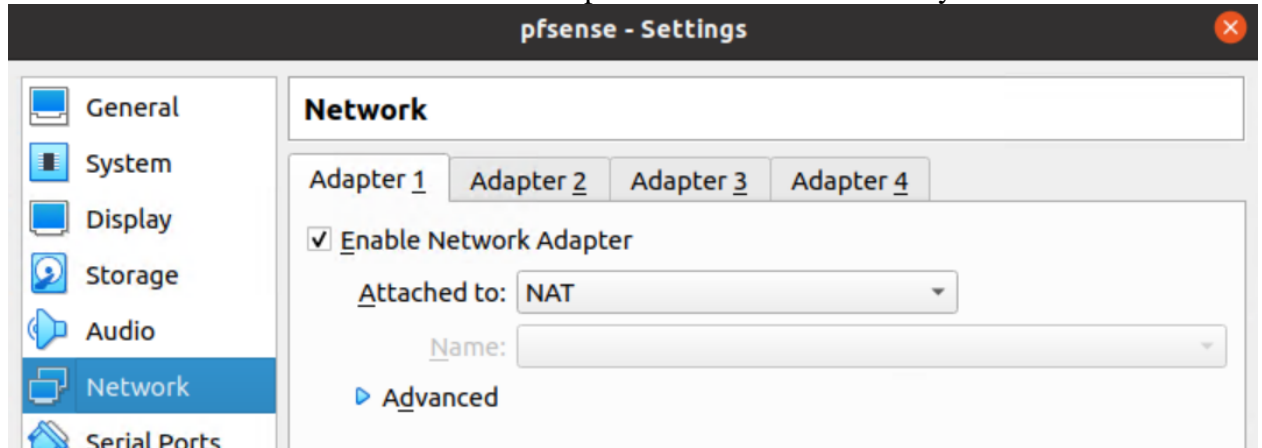


Figure 12: Wan Network

With the virtual machine configured, we start it up. Upon completion of the installation, we can remove the ISO from the storage settings and restart the virtual machine.

After the VM restarts, we encounter a menu with various options.

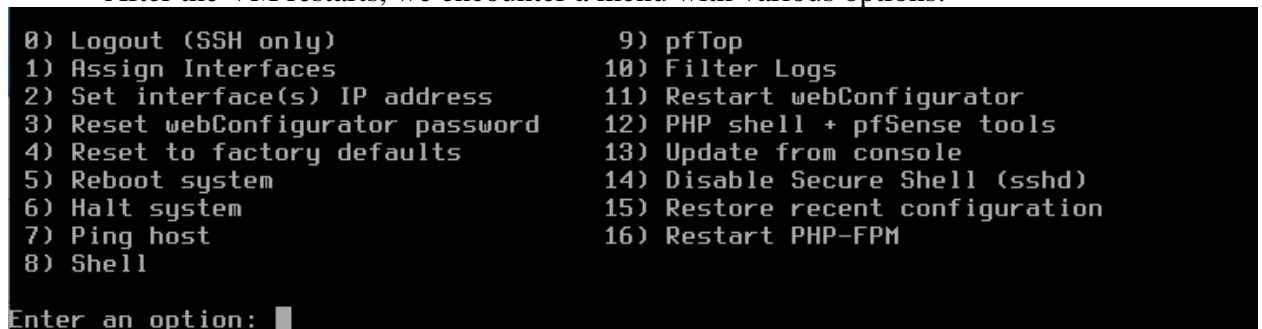


Figure 13: PfSense menu

Afterwards two important steps need to be performed:

- First, we assign interfaces (Option 1).
- Next, we set IP addresses for those interfaces (Option 2).

### **Assigning Interfaces**

Most of the time, the interfaces align with their appearance sequence in the VirtualBox network tab.

```

Enter an option: 1

Valid interfaces are:

em0      08:00:27:1e:4e:89  (up) Intel(R) Legacy PRO/1000 MT 82540EM
em1      08:00:27:5c:77:e8  (up) Intel(R) Legacy PRO/1000 MT 82540EM
em2      08:00:27:2b:4f:34 (up) Intel(R) Legacy PRO/1000 MT 82540EM
    
```

Figure 14: Interfaces of PfSense

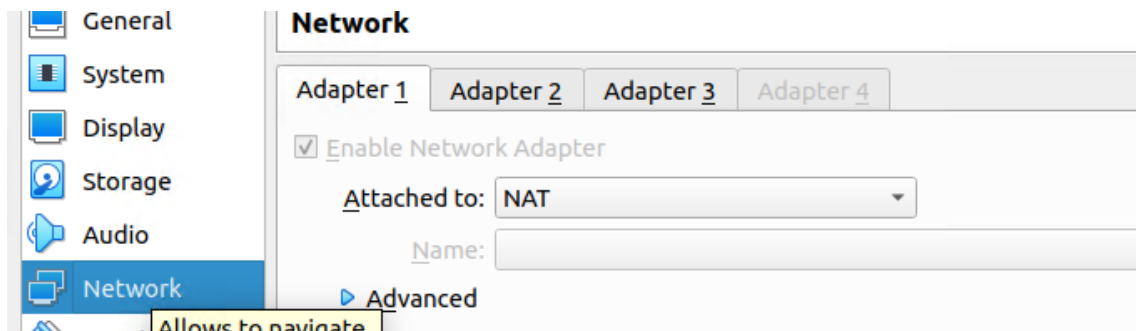


Figure 15: Adapters of Machine in VirtualBox

Typically, we assign em0 to WAN, em1 to LAN (housing the GOAD environment), and em2 to OPT1 (where our other machines will reside).

```

WAN (wan)      -> em0
LAN (lan)      -> em1
OPT1 (opt1)    -> em2
    
```

Figure 16: Interface mapping

### Configuring IP Addresses

For the WAN interface, which solely handles internet connectivity and obtains its IP address through DHCP, we leave it to acquire an IP address from our host machine, as it resides behind NAT.

```

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.2.15/24
    
```

Figure 17: WAN IP configuration

However, it's crucial to manually set up both LAN interfaces of PfSense as default gateways for the two LAN networks. The GOAD network, originally set to 192.168.56.0/24 with a default gateway of 192.168.56.100/24, requires the PfSense LAN interface to have the same IP address, i.e., 192.168.56.100/24.

```
Enter an option: 2
Available interfaces:
1 - WAN (em0 - dhcp)
2 - LAN (em1 - static)
3 - OPT1 (em2 - static)
Enter the number of the interface you wish to configure: 2
Configure IPv4 address LAN interface via DHCP? (y/n) n
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 192.168.56.100
Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8
Enter the new LAN IPv4 subnet bit count (1 to 32):
> 24
```

Figure 18: Configuring IP addressing of LAN interface.

For OPT1, we can assign an IP address of our choosing, such as 192.168.60.1/24.

```
Available interfaces:
1 - WAN (em0 - dhcp)
2 - LAN (em1 - static)
3 - OPT1 (em2 - static)
Enter the number of the interface you wish to configure: 3
Configure IPv4 address OPT1 interface via DHCP? (y/n) n
Enter the new OPT1 IPv4 address. Press <ENTER> for none:
> 192.168.60.1
Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8
Enter the new OPT1 IPv4 subnet bit count (1 to 32):
> 24
```

Figure 19: Configuring IP addressing of OPT1 interface.

This final configuration of the firewall results in a network layout that resembles the following:

```

*** Welcome to pfSense 2.7.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0          -> v4/DHCP4: 10.0.2.15/24
LAN (lan)      -> em1          -> v4: 192.168.56.100/24
OPT1 (opt1)    -> em2          -> v4: 192.168.60.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Disable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █

```

Figure 20: Final Configuration of PfSense

So, we have successfully configured the PfSense firewall, effectively segmenting our network into distinct LAN and WAN sections.

Once the PfSense firewall is set up in our lab environment, we proceed to configure its firewall rules.

First, we access the firewall's web configurator by navigating to <http://192.168.56.100> from a machine located in the LAN segment of our network, specifically the GOAD Environment. In the Firewall menu, we select the "Rules" tab, which provides us with an overview of the existing firewall rules.

Upon inspection, we notice that the existing rules permit all traffic from the LAN interface but restrict any traffic from the OPT1 interface. To enable seamless communication from the OPT1 segment, we need to add an "allow all" traffic rule for this interface.

In the "OPT1" menu, we click on "Add" to create a new rule. The rule's configuration will appear as follows:

- Action: Pass
- Interface: OPT1
- Address Family: IPv4
- Protocol: Any
- Source: Any
- Destination: Any

The screenshot shows the configuration for a firewall rule. The 'Action' is set to 'Pass'. The 'Disabled' checkbox is unchecked. The 'Interface' is 'OPT1'. The 'Address Family' is 'IPv4'. The 'Protocol' is 'Any'. The 'Source' section is set to 'Source Address' with 'OPT1 net' and 'Invert match' unchecked. The 'Destination' section is set to 'Destination Address' with 'any' and 'Invert match' unchecked. The 'Log' checkbox is checked. The 'Description' is 'Allow OPT1 traffic to all Networks'. At the bottom, there is an 'Advanced Options' section with a 'Display Advanced' button.

Figure 21: Allow all rule for OPT1 network segment.

It's important to note that for every firewall rule we establish or for the default rules created during installation, we must enable the logging of packets handled by that rule, as it appears above. This logging capability is crucial for our forthcoming steps in the Cyber Range Development. It will allow us to collect and send all these logs to Azure Sentinel, enhancing our detection capabilities significantly.

### Suricata configuration

To enable Intrusion Detection System (IDS) capabilities in pfSense using Suricata we first need to access the Package Manager by navigating to the top menu of the firewall. Go to System → Package Manager → Available Packages. In the search term box, type 'Suricata,' and then click the 'Install' button on the right to initiate the installation.

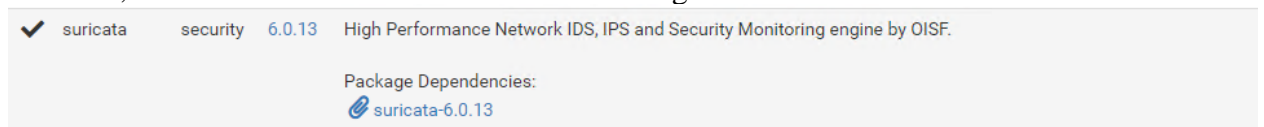


Figure 22: Suricata package

Once Suricata is installed, access the firewall's top menu. Go to Services → Suricata → Interfaces. Add a Suricata instance to the LAN interface. Enable the option 'Suricata will send Alerts from this interface to the firewall's system log.' This will allow Suricata to send logs to the SIEM later.



General Settings	
<b>Enable</b>	<input checked="" type="checkbox"/> Checking this box enables Suricata inspection on the interface.
<b>Interface</b>	<input type="text" value="LAN (em1)"/> Choose which interface this Suricata instance applies to. In most cases, you will want to choose LAN here if this is the first Suricata-configured interface.
<b>Description</b>	<input type="text" value="Suricata on LAN Interface"/> Enter a meaningful description here for your reference. The default is the pfSense interface friendly description.
Logging Settings	
<b>Send Alerts to System Log</b>	<input checked="" type="checkbox"/> Suricata will send Alerts from this interface to the firewall's system log. NOTE: the FreeBSD syslog daemon will automatically truncate exported messages to 480 bytes max.
<b>Log Facility</b>	<input type="text" value="LOCAL1"/> Select system log Facility to use for reporting. Default is LOCAL1.
<b>Log Priority</b>	<input type="text" value="CRIT"/> Select system log Priority (Level) to use for reporting. Default is NOTICE.
<b>Enable Stats Collection</b>	<input type="checkbox"/> Suricata will periodically gather performance statistics for this interface. Default is Not Checked.
<b>Enable HTTP Log</b>	<input checked="" type="checkbox"/> Suricata will log decoded HTTP traffic for the interface. Default is Checked.
<b>Append HTTP Log</b>	<input checked="" type="checkbox"/> Suricata will append-to instead of clearing HTTP log file when restarting. Default is Checked.
<b>Log Extended HTTP Info</b>	<input checked="" type="checkbox"/> Suricata will log extended HTTP information. Default is Checked.
<b>Enable TLS Log</b>	<input type="checkbox"/> Suricata will log TLS handshake traffic for the interface. Default is Not Checked.
<b>Enable File-Store</b>	<input type="checkbox"/> Suricata will extract and store files from application layer streams. Default is Not Checked. WARNING: Enabling file-store will consume a significant amount of disk space on a busy network!

Figure 23: General Configuration of Suricata

To download rules for Suricata, navigate to Services → Suricata → Global Settings. Enable the following options: "Install ETOpen Emerging Threats rules," "Install Snort GPLv2 Community rules," "Install Feodo Tracker Botnet C2 IP rules," and "Install ABUSE.ch SSL Blacklist rules." Also, enable the option 'Copy Suricata messages to the firewall system log.'

Please Choose The Type Of Rules You Wish To Download		
<b>Install ETOpen Emerging Threats rules</b>	<input checked="" type="checkbox"/> ETOpen is a free open source set of Suricata rules whose coverage is more limited than ETPro.	<input type="checkbox"/> Use a custom URL for ETOpen downloads Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETOpen rules.
<b>Install ETPro Emerging Threats rules</b>	<input type="checkbox"/> ETPro for Suricata offers daily updates and extensive coverage of current malware threats.	<input type="checkbox"/> Use a custom URL for ETPro rule downloads The ETPro rules contain all of the ETOpen rules, so the ETOpen rules are not required and are disabled when the ETPro rules are selected. <a href="#">Sign Up for an ETPro Account</a> . Enabling the custom URL option will force the use of a custom user-supplied URL when downloading ETPro rules.
<b>Install Snort rules</b>	<input type="checkbox"/> Snort free Registered User or paid Subscriber rules <a href="#">Sign Up for a free Registered User Rules Account</a> <a href="#">Sign Up for paid Snort Subscriber Rule Set (by Talos)</a>	<input type="checkbox"/> Use a custom URL for Snort rule downloads Enabling the custom URL option will force the use of a custom user-supplied URL when downloading Snort Subscriber rules.
<b>Install Snort GPLv2 Community rules</b>	<input checked="" type="checkbox"/> The Snort Community Ruleset is a GPLv2 Talos-certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions.	<input type="checkbox"/> Use a custom URL for Snort GPLv2 rule downloads This ruleset is updated daily and is a subset of the subscriber ruleset. If you are a Snort Subscriber Rules customer (paid subscriber), the community ruleset is already built into your download of the Snort Subscriber rules, and there is no benefit in adding this rule set separately.
<b>Install Feodo Tracker Botnet C2 IP rules</b>	<input checked="" type="checkbox"/> The Feodo Botnet C2 IP Ruleset contains Dridex and Emotet/Heodo botnet command and control servers (C&Cs) tracked by Feodo Tracker.	
<b>Install ABUSE.ch SSL Blacklist rules</b>	<input checked="" type="checkbox"/> The ABUSE.ch SSL Blacklist Ruleset contains the SSL cert fingerprints of all SSL certs blacklisted by ABUSE.ch.	
<b>Hide Deprecated Rules Categories</b>	<input checked="" type="checkbox"/> Hide deprecated rules categories in the GUI and remove them from the configuration. Default is Not Checked.	

Figure 24: Suricata rule feeds

General Settings	
<b>Remove Blocked Hosts Interval</b>	NEVER Please select the amount of time you would like hosts to be blocked. Note this setting is only applicable when using Legacy Mode blocking! This setting is ignored when using Inline IPS Mode. Hint: in most cases, 1 hour is a good choice.
<b>Log to System Log</b>	<input checked="" type="checkbox"/> Copy Suricata messages to the firewall system log.
<b>Log Facility</b>	LOCAL1 Select system log facility to use for reporting. Default is LOCAL1.
<b>Log Priority</b>	NOTICE Select system log Priority (Level) to use for reporting. Default is NOTICE.
<b>Keep Suricata Settings After Deinstall</b>	<input checked="" type="checkbox"/> Settings will not be removed during package deinstallation.
<b>Clear Blocked Hosts After Deinstall</b>	<input checked="" type="checkbox"/> Click to clear all blocked hosts added by Suricata when removing the package. Default is checked.

Figure 25: Copy Suricata logs to Firewall logs

To download updated rules of the options enabled in the previous step, move to Services → Suricata → Updates tab.

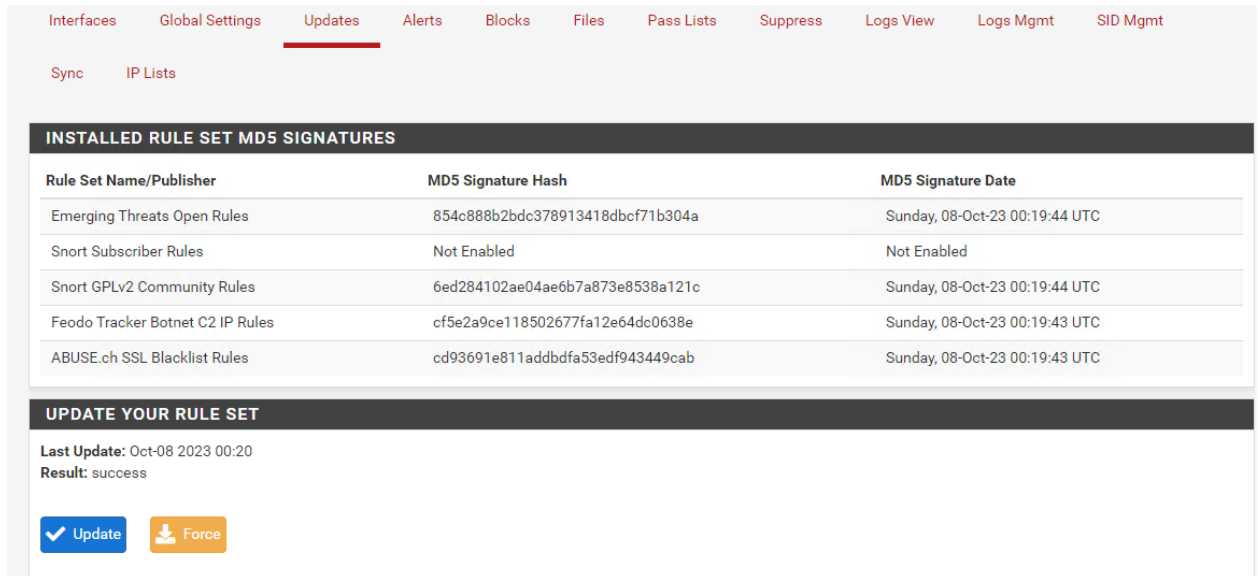


Figure 26: Update Suricata rule signatures

To enable Suricata service return to Services → Suricata → Interfaces tab. If the Suricata service is not running, enable it.

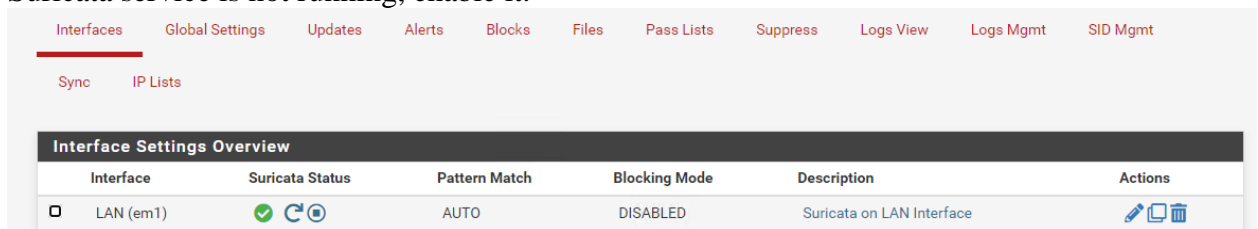


Figure 27: Enabled Suricata service in LAN interface

Once Suricata is running, it will actively monitor all traffic passing through the LAN interface. Optionally, you can add the Suricata service to the OPT1 interface to monitor traffic there as well. However, keep in mind that adding Suricata to multiple interfaces may impact the firewall's performance.

So, Suricata is set up within pfSense, enhancing our network security by providing intrusion detection capabilities and generating logs that can be sent to the SIEM for further analysis and detection of potential threats.

### 4.1.3 Windows and Linux machine

In this subsection, we focus on setting up the essential client machines, representing common endpoints in a network. We'll provide practical, step-by-step instructions for configuring both Windows and Linux clients, simulating real-world usage scenarios. Windows will serve as a user's workstation, while Linux will host critical applications. Additionally, the Linux host will be used at a later stage as a log collector which will forward the Syslog and CEF logs in Azure Sentinel SIEM.

## Linux Installation

The installation of Linux within our cybersecurity environment is a straightforward process. Download Ubuntu 20.04 LTS by visiting <https://releases.ubuntu.com/20.04.6/>. It is preferred to Download the Server install image since we won't be using a Graphical User Interface (GUI) for this Linux machine, but it is no mandatory.

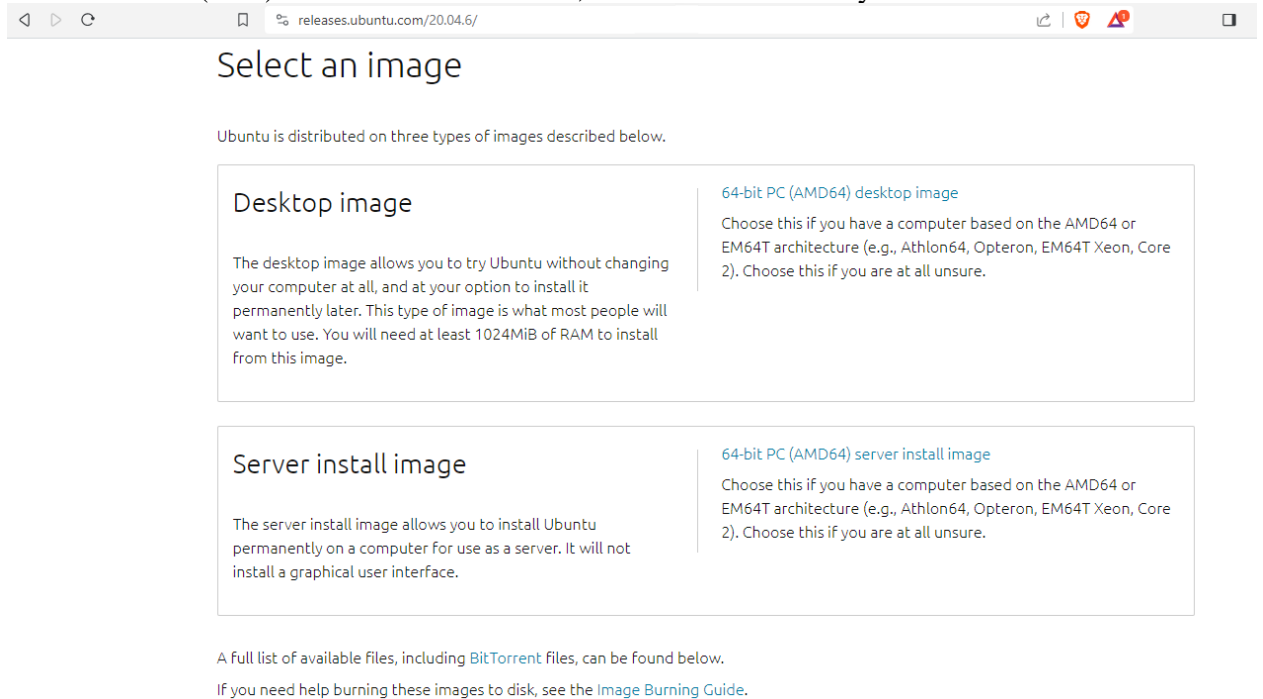


Figure 28: [Ubuntu 20.04 LTS installation page](https://releases.ubuntu.com/20.04.6/)

Set up a Virtual Machine (VM) to host the Linux installation. Allocate 2 virtual CPUs (vCPUs), 2 GB of RAM, and 22 GB of storage for the VM. These resources are sufficient for our purposes.

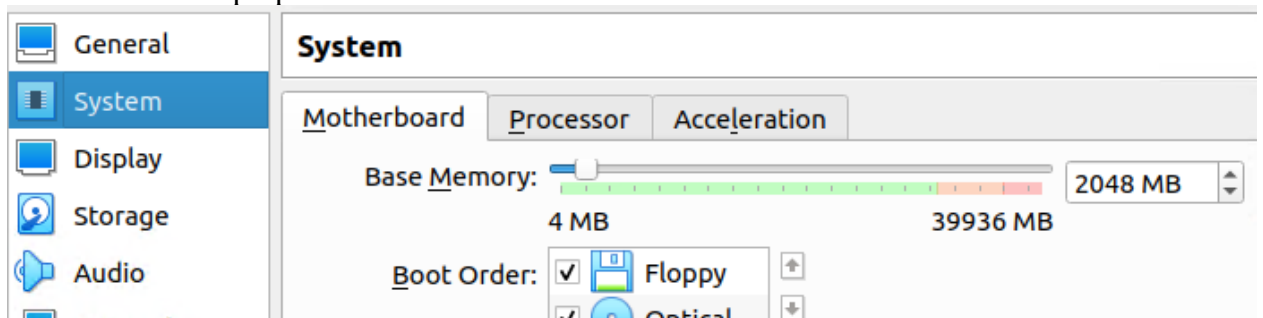


Figure 29: RAM resources of Linux machine

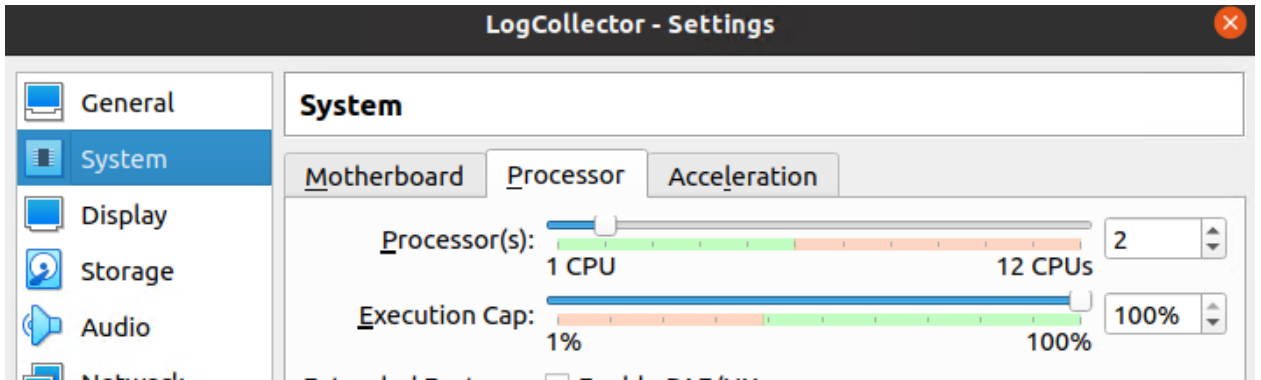


Figure 30: CPU resources of Linux machine

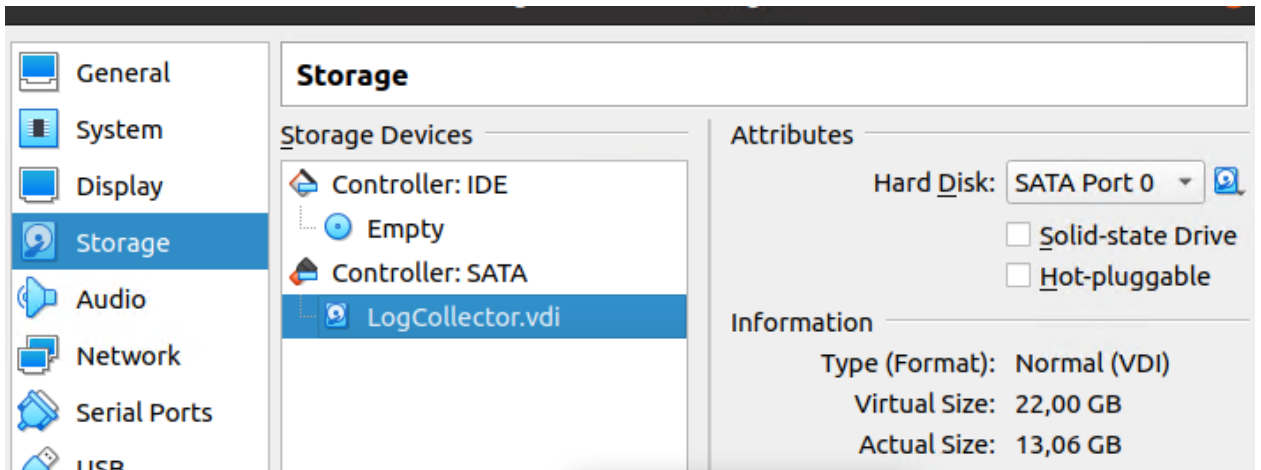


Figure 31: Storage resources of Linux machine

If not already done, attach the installation media downloaded previously to the VM's controller (IDE).

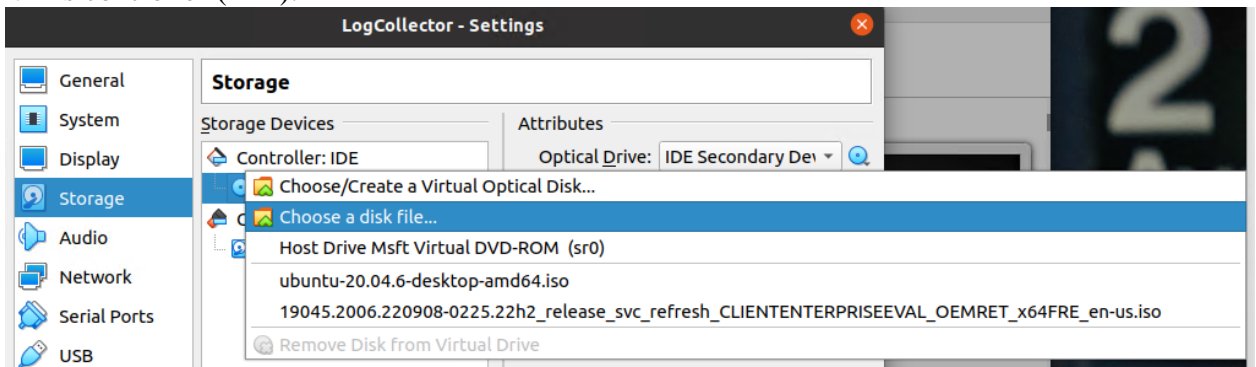


Figure 32: Attaching installation media.

In the VM's Network tab, assign it to the 'Internal Network' labeled as 'intent.' This configuration places the Linux machine in the 'OPT1' network of the PfSense firewall.

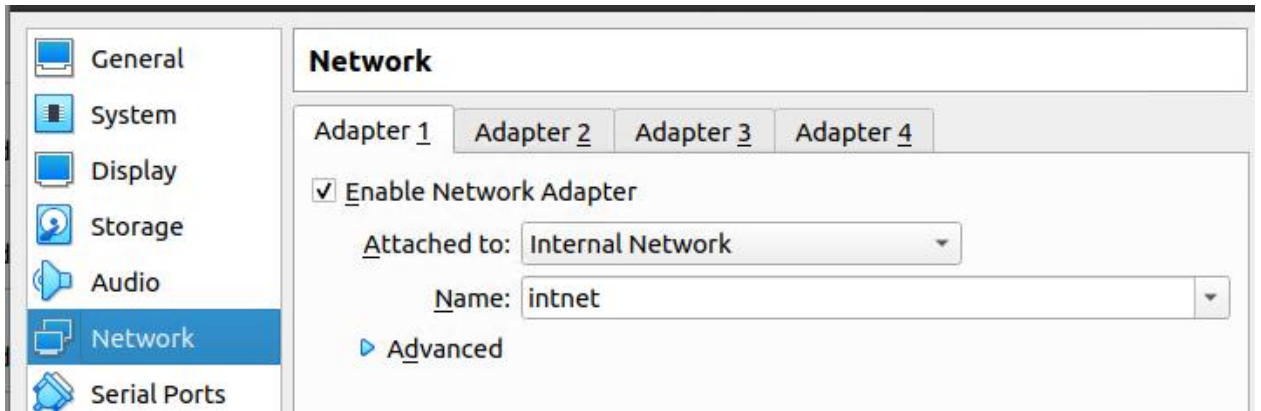


Figure 33: Network resource of Linux machine

Begin the installation process for the Linux machine. No additional packages are required for this installation, and networking settings can be configured automatically through DHCP.

### Windows Installation

Since Windows is proprietary software and not free, you can download the free evaluation media from <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise>.

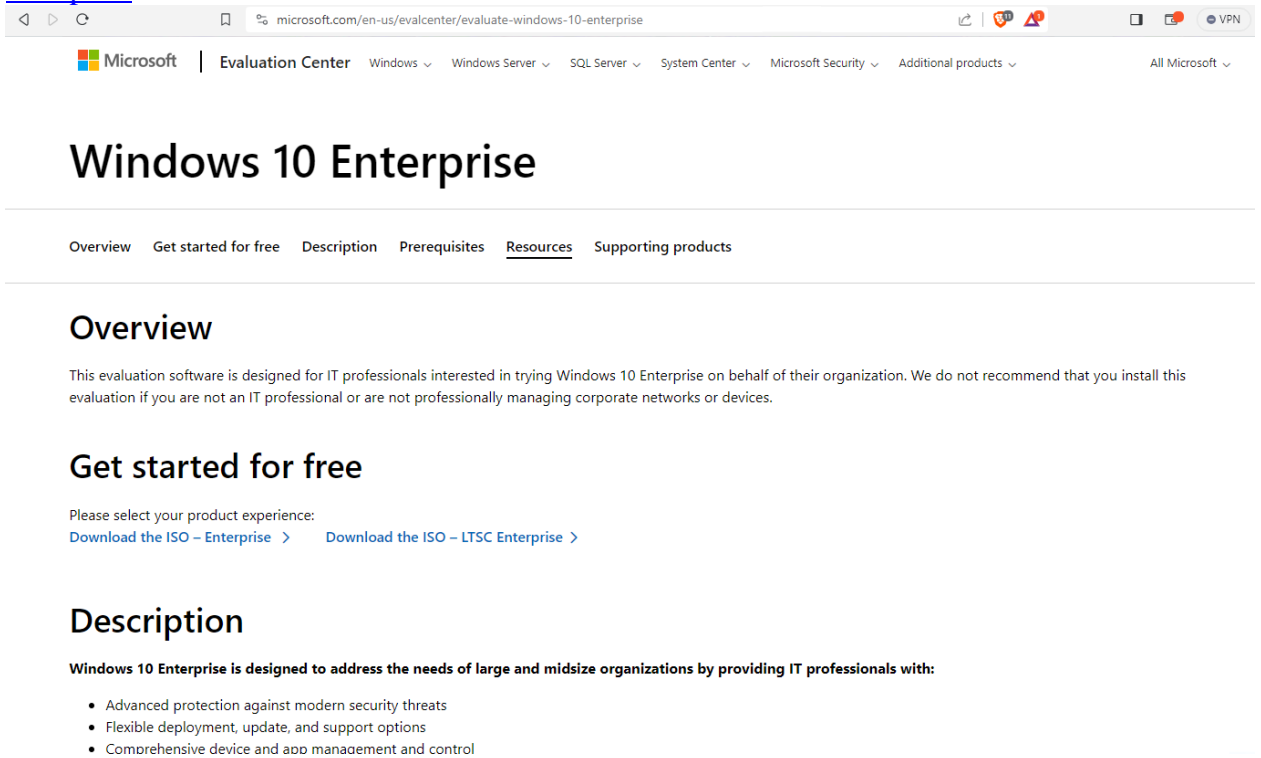


Figure 34: [Windows 10 Installation media Download page](https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise)

To set up a Virtual Machine (VM) to host the Windows installation, allocate 2 virtual CPUs (vCPUs), 2 GB of RAM, and 50 GB of storage for the VM. These resources are sufficient for this purpose.

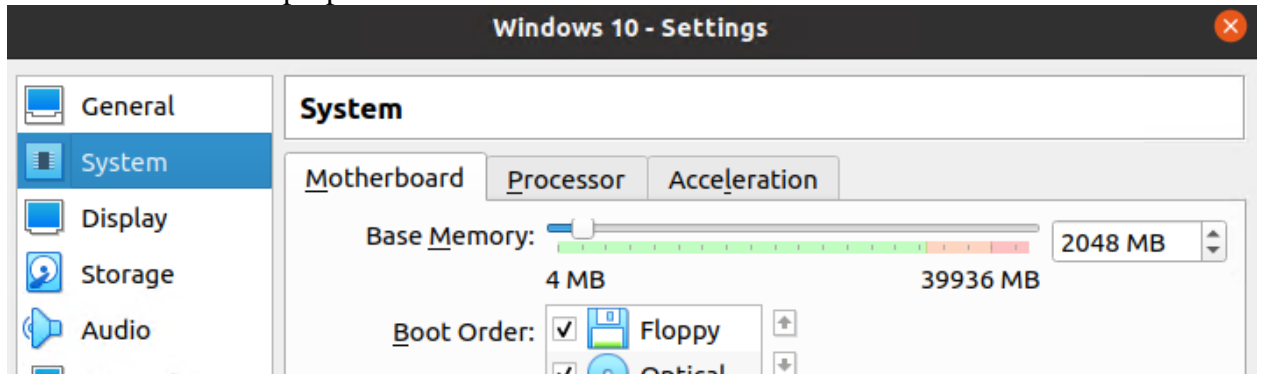


Figure 35: RAM resources for the Windows machine

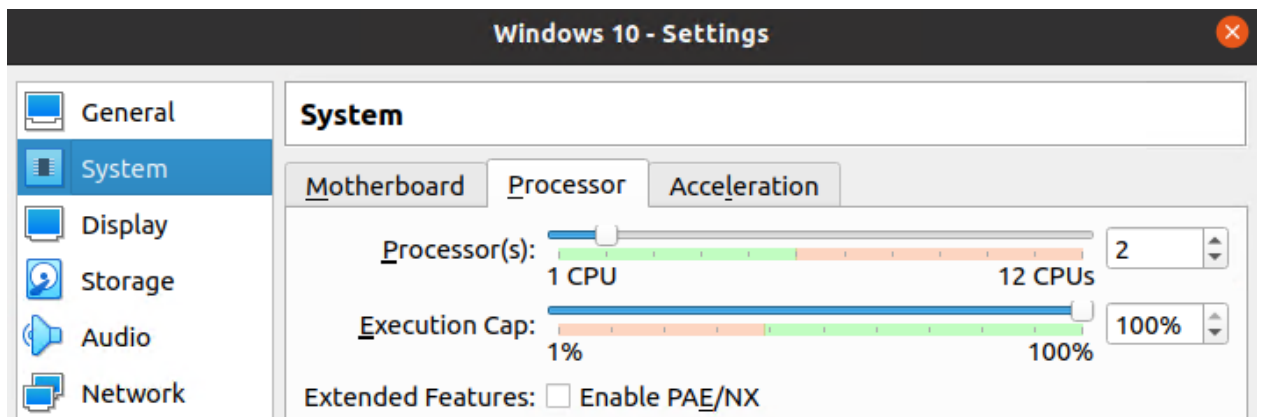


Figure 36: CPU resources for the Windows machine

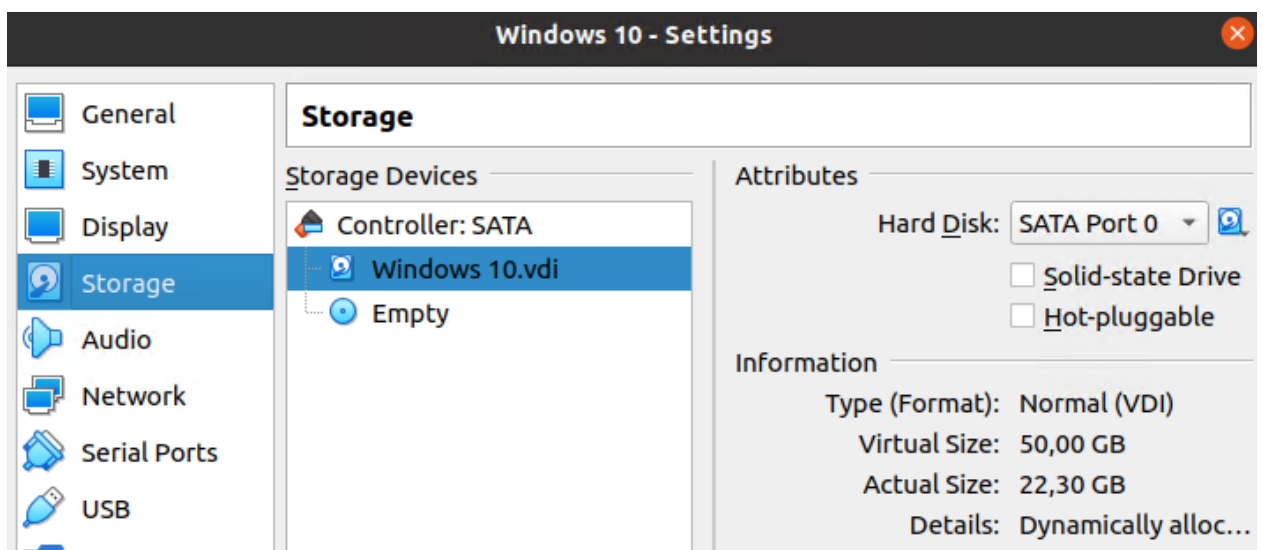


Figure 37: Storage resources for the Windows machine

Again, attach the Windows installation media you downloaded in step 1 to the VM's controller (IDE) if it's not already done.

In the VM's Network tab, assign it to the 'Internal Network' labeled as 'intent2.' This configuration places the Windows machine in the 'LAN' network of PfSense, the same as the GOAD environment.

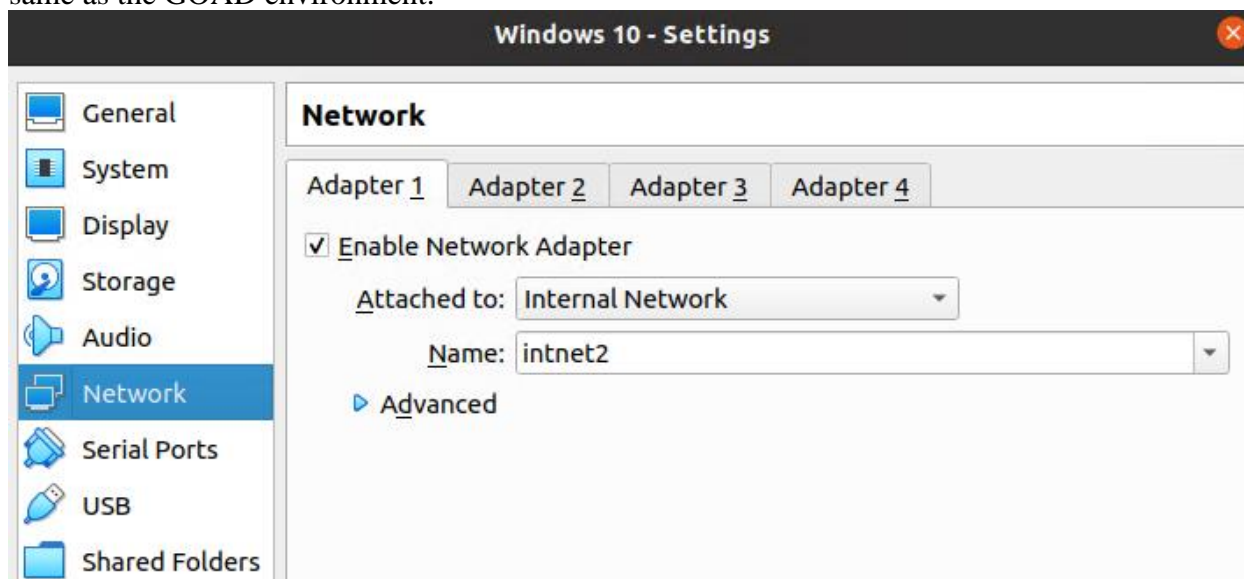


Figure 38: Same Network as the GOAD environment

Afterwards, begin the installation process for the Windows machine. Similar to the Linux installation, no additional packages are required, and networking settings can be configured automatically through DHCP.

Optionally, the Windows machine can be added to one of the three domains that the GOAD environment provides, although it is not necessary for our testing purposes.

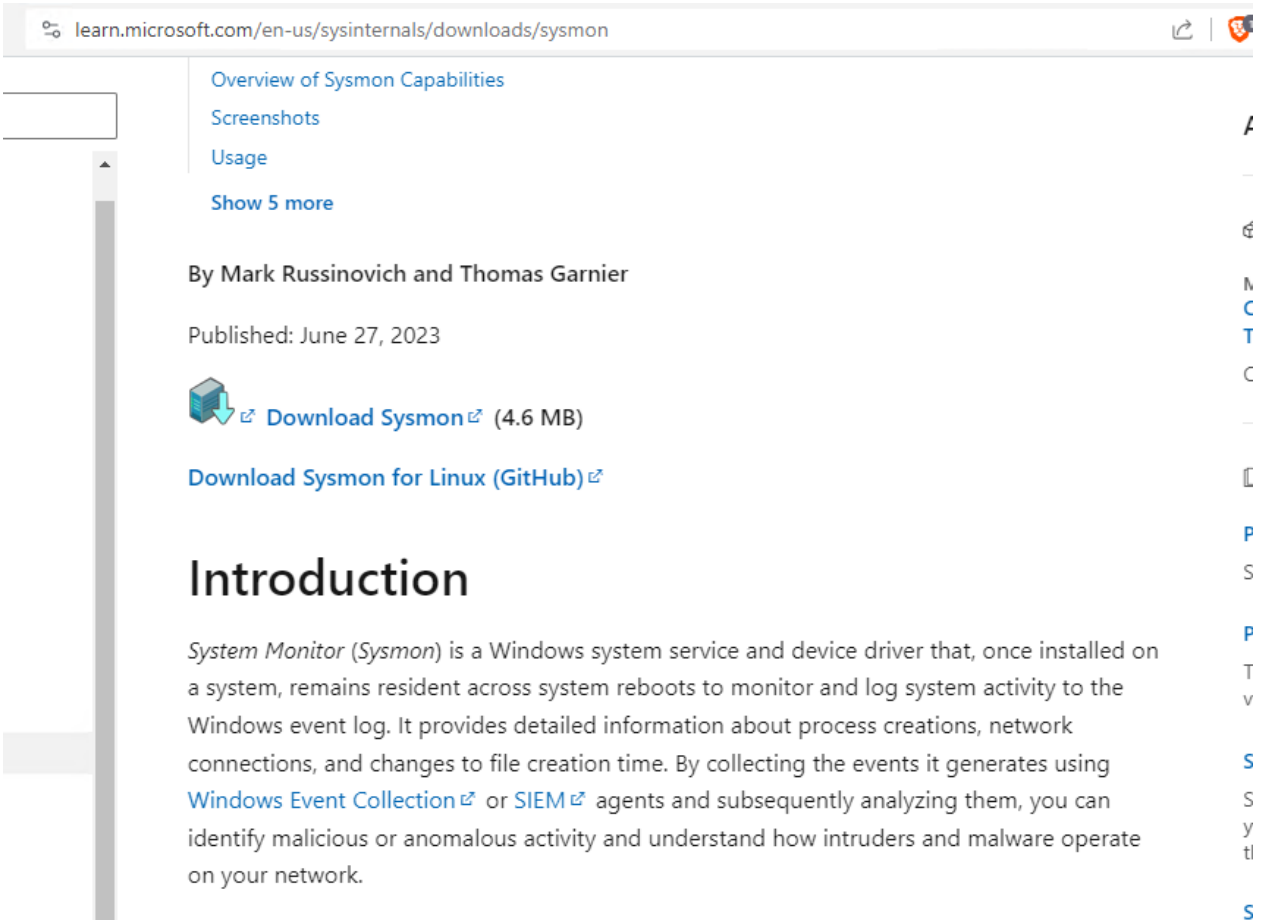
#### 4.1.4 SIEM Technologies

Effective SIEM technologies heavily rely on robust and comprehensive logging. Therefore, our initial step in this section involves enabling additional logging on our Windows machines. Without proper logging, SIEM solutions may not effectively detect and respond to potential threats.

##### ***Enhanced Logging: Sysmon***

To bolster our SIEM capabilities, we will commence by installing the Sysmon agent on all Windows machines within our range. **Sysmon** [21] [26], in conjunction with Wazuh [27] and Sentinel [28] SIEMs, plays a crucial role as a pseudo-Endpoint Detection and Response (EDR) system, providing us with enhanced visibility into potential threats across our infrastructure.






learn.microsoft.com/en-us/sysinternals/downloads/sysmon

- Overview of Sysmon Capabilities
- Screenshots
- Usage
- Show 5 more

By Mark Russinovich and Thomas Garnier

Published: June 27, 2023

 [Download Sysmon](#) (4.6 MB)

[Download Sysmon for Linux \(GitHub\)](#)

## Introduction

*System Monitor (Sysmon)* is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. By collecting the events it generates using [Windows Event Collection](#) or [SIEM](#) agents and subsequently analyzing them, you can identify malicious or anomalous activity and understand how intruders and malware operate on your network.

Figure 39: Sysmon download page

For Sysmon to function effectively, it requires a well-defined configuration. Without proper configuration, it may either miss monitoring critical activities or inundate our SIEM with excessive event logs. To ensure optimal Sysmon configuration, we will utilize a meticulously crafted configuration file developed by Olaf Hartong, available on GitHub at <https://github.com/olafhartong/sysmon-modular>. This configuration file strikes a balance by monitoring only suspicious activities, preventing an overwhelming influx of logs.

**Pre-Generated configurations** [↗](#)

Type	Config	Description
default	<a href="#">sysmonconfig.xml</a>	This is the balanced configuration, most used, more information <a href="#">here</a>
default+	<a href="#">sysmonconfig-with-filedelete.xml</a>	This is the balanced configuration, most used, more information including FileDelete file saves
verbose	<a href="#">sysmonconfig-excludes-only.xml</a>	This is the very verbose configuration, all events are included, only the exclusion modules are applied. This should not be used in production without validation, will generate a significant amount of data and might impact performance. More information <a href="#">here</a>
super verbose	<a href="#">sysmonconfig-research.xml</a>	A configuration with extreme verbosity. The log volume expected from this file is significantly high, really DO NOT USE IN PRODUCTION! This config is only for research, this will use way more CPU/Memory. Only enable prior to running the to be investigated technique, when done load a lighter config.
MDE augment	<a href="#">sysmonconfig-mde-augmentation.xml</a>	A configuration to augment Defender for Endpoint, intended to augment the information and have as little overlap as possible. This is based on the default/balanced config and will <i>not generate all events</i> for Sysmon, there are comments in the config. In the benefit of IR, consider using the excludes only config and only ingest the enriching events. (Blog with more rationale soon)

Figure 40: Sysmon configuration file. <https://github.com/olafhartong/sysmon-modular>.

To set up Sysmon with the provided configuration, we will execute the following command **Sysmon64.exe -c sysmonconfig.xml**.

```
C:\Windows\system32>"Z:\GOAD files\Sysmon64.exe" -c "Z:\GOAD files\sysmonconfig.xml"

System Monitor v15.0 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2023 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.90
Configuration file validated.
Configuration updated.
```

Figure 41: Sysmon Installation

### **Enhancing Domain Controller Logging**

In addition to Sysmon, we will enhance logging on our Domain Controllers to effectively identify and respond to Active Directory attacks. To achieve this, follow the guidance provided by Microsoft at <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations> for the three Domain Controllers.

These tables contain the Windows default setting, the baseline recommendations, and the stronger recommendations for these operating systems.

**Audit Policy Tables Legend**

Notation	Recommendation
Yes	Enable in general scenarios
No	Do not enable in general scenarios
If	Enable if needed for a specific scenario, or if a role or feature for which auditing is desired is installed on the machine
DC	Enable on domain controllers
[Blank]	No recommendation

Figure 42: Enhanced logging in Active Directory service provided by Microsoft <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations>

Additionally, enable additional logging for 4662 (An operation was performed on an object) events, as detailed in the guide at <https://learn.microsoft.com/en-us/defender-for-identity/configure-windows-event-collection#configure-object-auditing>. Please note that during the configuration process, we will ignore the part that advises unchecking Read permissions.

## Configure object auditing

To collect 4662 events, it's also necessary to configure object auditing on the user, group and computer objects. Here's how to enable auditing on all users, groups, and computers in the Active Directory domain:

### Note

It is important to **review and verify your audit policies** before enabling event collection to ensure that the domain controllers are properly configured to record the necessary events.

If configured properly, this auditing should have minimal effect on server performance.

1. Go to the **Active Directory Users and Computers** console.
2. Select the domain you want to audit.
3. Select the **View** menu and select **Advanced Features**.
4. Right-click the domain and select **Properties**.

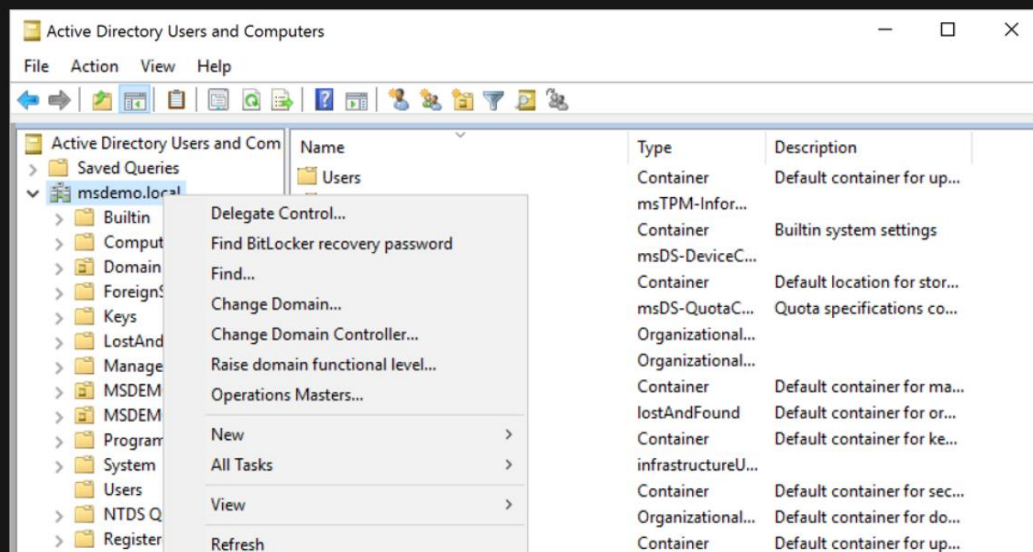


Figure 43: Additional logging for 4662 events

By implementing these logging enhancements, we fortify our SIEM technologies with essential data that is needed to properly monitor Active Directory windows environments.

### Setting Up Wazuh SIEM

Our next step in integrating SIEM technologies into our cyber range is the installation and configuration of Wazuh. Wazuh is an open-source SIEM solution with powerful response capabilities. To streamline the setup process, we have the option of using a preconfigured appliance, which is ready for use. This appliance can be obtained from

<https://documentation.wazuh.com/current/deployment-options/virtual-machine/virtual-machine.html> and is based on Amazon Linux 2.

Download the [virtual appliance \(OVA\)](#), which contains the following components:

- Amazon Linux 2
- Wazuh manager 4.5.2
- Wazuh indexer 4.5.2
- Filebeat-OSS 7.10.2
- Wazuh dashboard 4.5.2

### Packages list

Distribution	Architecture	VM Format	Version	Package
Amazon Linux 2	64-bit	OVA	4.5.2	wazuh-4.5.2.ova (sha512)

Figure 44: Preconfigured Wazuh server

While the preconfigured appliance comes with default resource settings (4 CPUs, 8 GB of RAM, and 50 GB of storage), we'll adjust better align with our non-production environment. We'll reduce the appliance's resource allocation to 2 vCPUs and 4 GB of RAM, ensuring efficient resource usage without compromising functionality.

Out of the box, the Wazuh VM is configured with the following specifications:

Component	CPU (cores)	RAM (GB)	Storage (GB)
Wazuh v4.5.2 OVA	4	8	50

However, this hardware configuration can be modified depending on the number of protected endpoints and indexed alert data. More information about requirements can be found [here](#).

Figure 45: Default resources of the preconfigured server

In the VirtualBox Network settings, we'll configure the appliance to interface with the OPT1 LAN interface of PfSense. This network alignment ensures proper communication within our cyber range environment.

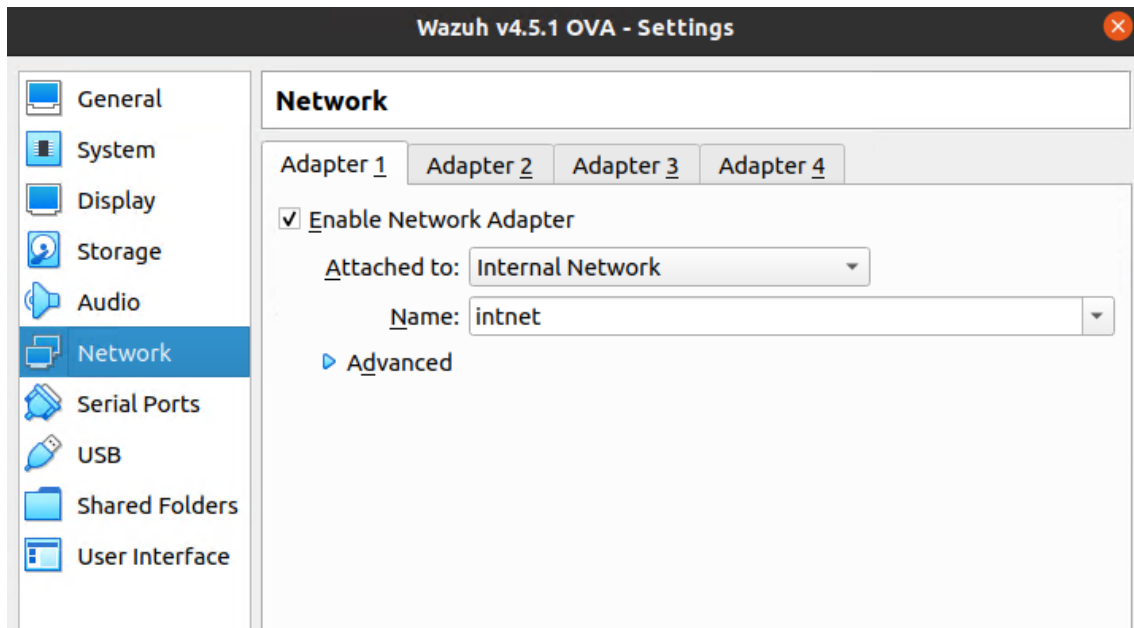


Figure 46: Network interface of Wazuh Server

After booting the appliance, we'll access its terminal and use the **'ip addr'** command to identify its IP address, so we can access the Wazuh web interface.

```
[wazuh-user@wazuh-server ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:3c:53:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.60.100/24 brd 192.168.60.255 scope global dynamic eth0
        valid_lft 7149sec preferred_lft 7149sec
    inet6 fe80::a00:27ff:fe3c:536e/64 scope link
        valid_lft forever preferred_lft forever
```

Figure 47: Wazuh IP address obtained through DHCP.

From any machine within our cyber range (as all traffic is permitted), we'll navigate to the Wazuh web interface at <http://192.168.60.100>. Upon arrival, we'll log in using the default credentials 'admin:admin.'

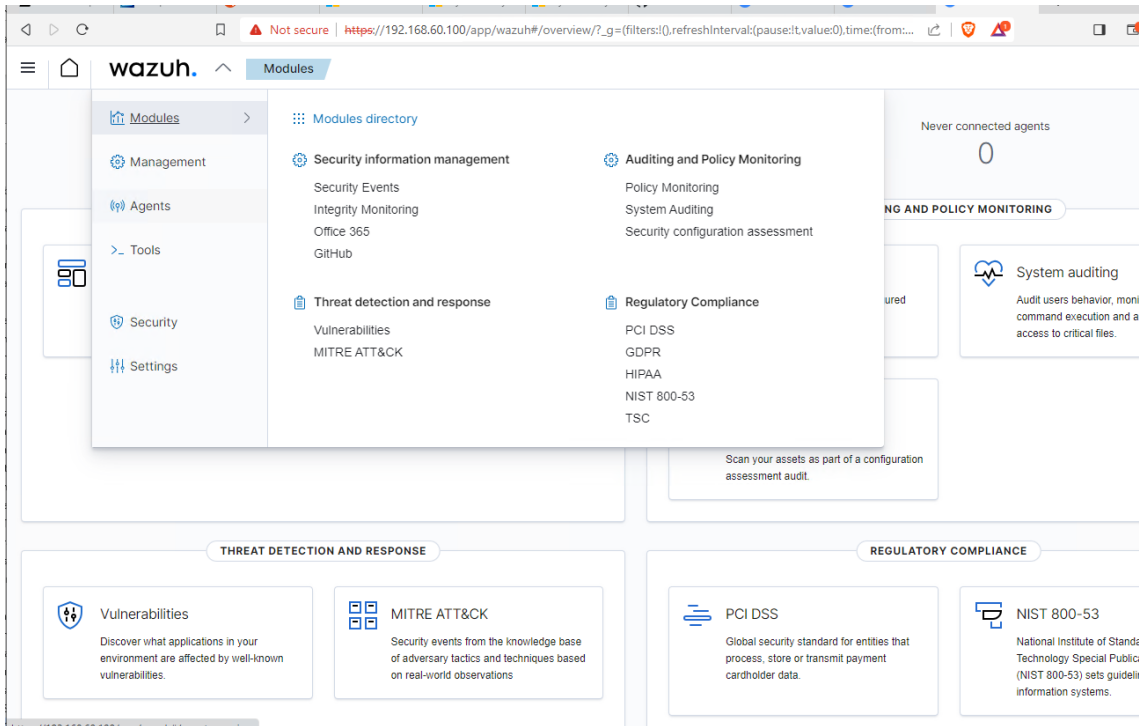


Figure 48: Wazuh Web Interface

To begin collecting logs from our Windows machines, we'll follow the Wazuh installation guide for Windows agents. Within the Wazuh interface, we'll navigate to 'WAZUH' -> 'Agents' -> 'Deploy new agent.'

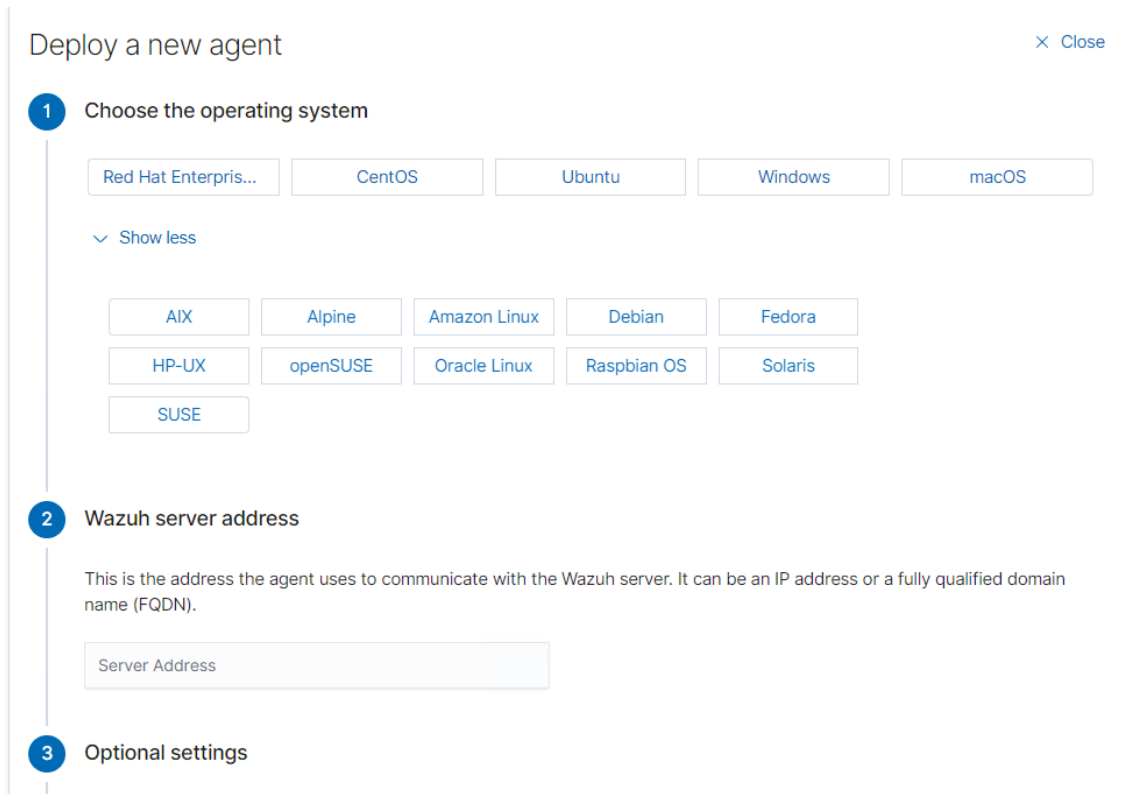


Figure 49: Wazuh agent installation page

From there, we'll select the Windows operating system and provide the IP address of our Wazuh server (192.168.60.100). The system will generate an installation command to be executed via PowerShell on each Windows machine. It important to note that during the installation the Windows machines must have network connectivity to both the Wazuh server and the internet.

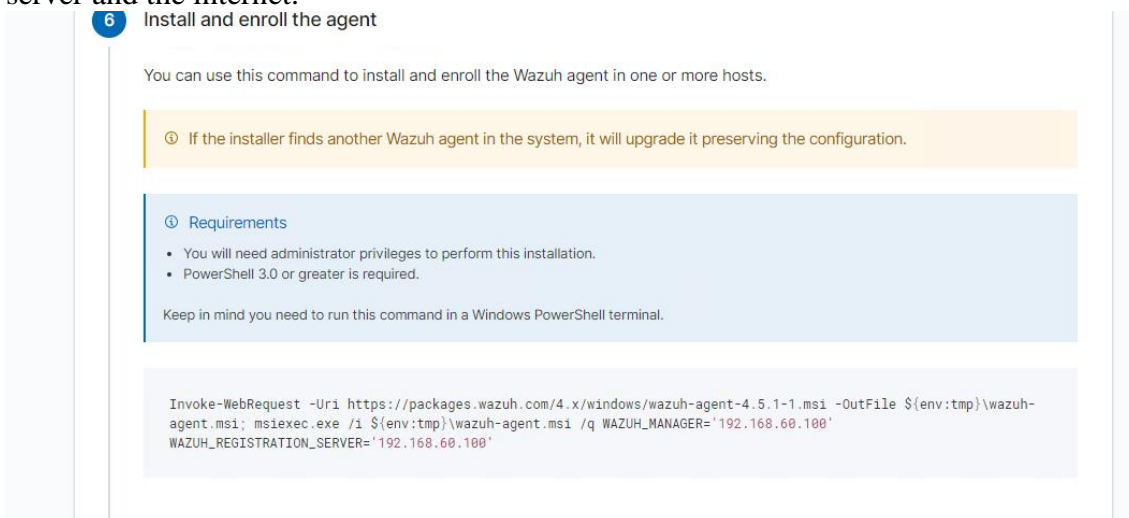
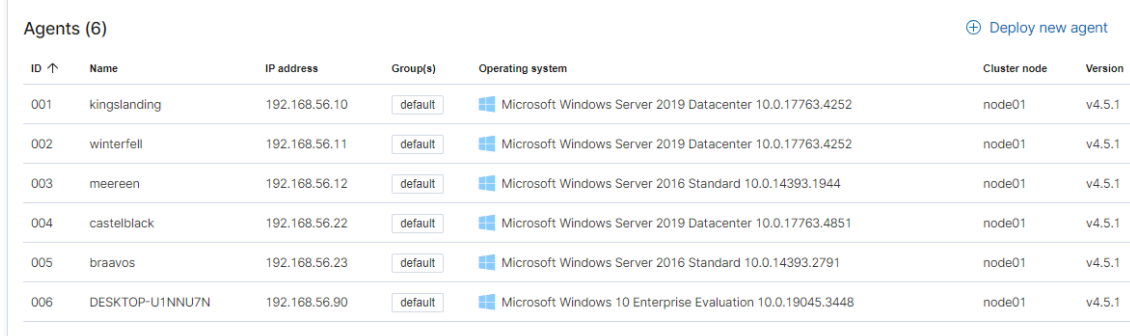


Figure 50: Wazuh Windows installation command



As we proceed with installing the Wazuh agents on all servers and the Windows client, we will monitor their status within the 'Agents' tab of the Wazuh interface.



ID ↑	Name	IP address	Group(s)	Operating system	Cluster node	Version
001	kingslanding	192.168.56.10	default	Microsoft Windows Server 2019 Datacenter 10.0.17763.4252	node01	v4.5.1
002	winterfell	192.168.56.11	default	Microsoft Windows Server 2019 Datacenter 10.0.17763.4252	node01	v4.5.1
003	meereen	192.168.56.12	default	Microsoft Windows Server 2016 Standard 10.0.14393.1944	node01	v4.5.1
004	castelblack	192.168.56.22	default	Microsoft Windows Server 2019 Datacenter 10.0.17763.4851	node01	v4.5.1
005	braavos	192.168.56.23	default	Microsoft Windows Server 2016 Standard 10.0.14393.2791	node01	v4.5.1
006	DESKTOP-U1NNU7N	192.168.56.90	default	Microsoft Windows 10 Enterprise Evaluation 10.0.19045.3448	node01	v4.5.1

Figure 51: Wazuh agents

To enable Windows servers to send Sysmon logs to the Wazuh server, we'll need to modify the configuration file located at *C:\Program Files (x86)\ossec-agent\ossec.conf* on each Windows machine. In this file, we will add the following code snippet:

```
<localfile>
<location>Microsoft-Windows-Sysmon/Operational</location>
<log_format>eventchannel</log_format>
</localfile>
```

This addition to the configuration file instructs the OSSEC agent on each Windows machine to start sending Sysmon logs to the Wazuh server. With this configuration in place, Wazuh will be able to receive and analyze these critical security logs.

### Setting Up Azure Sentinel

To setup Azure Sentinel, login to <https://portal.azure.com> [29] and start by creating a subscription with a valid email address. Use the search bar at the top, type "subscriptions," and click on "Add" to create one.

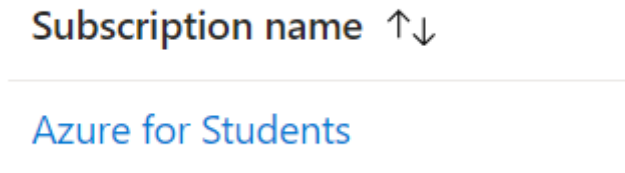


Figure 52: Azure Subscription

Following the successful creation of the subscription, establish a resource group to house the Log Analytics workspace, which serves as the repository for storing logs. Use the search feature once again, this time to find "Resource group", choose "Create" and define the name for your resource group.



Figure 53: Resource Group

Once the resource group is in place, search for "Sentinel" in the top bar and select "Microsoft Sentinel." This initiates the setup of an Azure Sentinel instance, prompting you to create a new Log Analytics workspace.

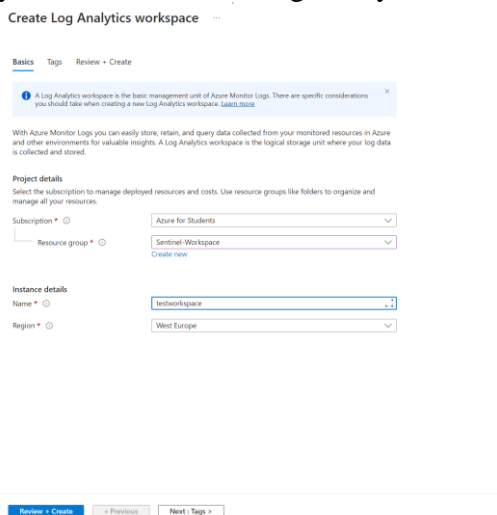


Figure 54: New Analytic Workspace

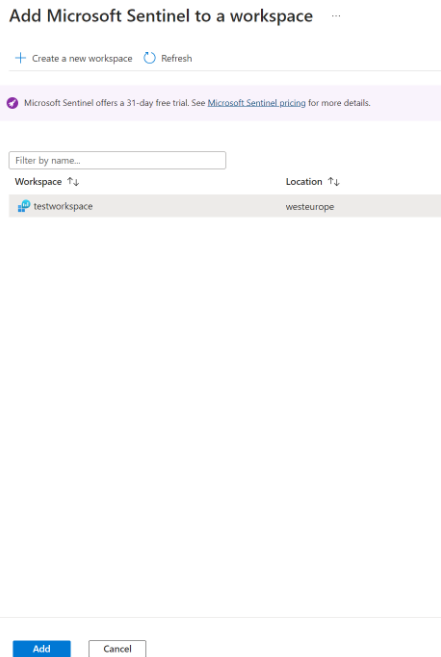


Figure 55: Add Sentinel to Log Analytics workspace.

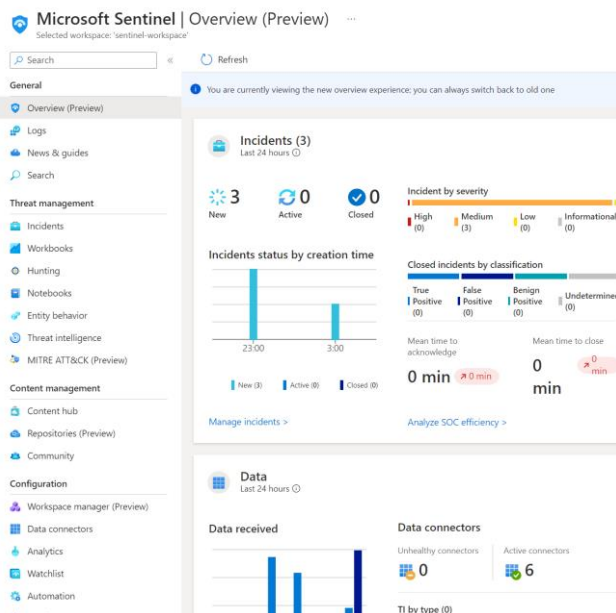


Figure 56: Sentinel Web interface

When the installation is complete, to enable the reception of Windows, Common Event Format (CEF), and Syslog events within the Log Analytics workspace, go to the Content Hub. Search for "Windows" and install the following content: "Windows Security Events," "Common Event Format," and "Syslog."

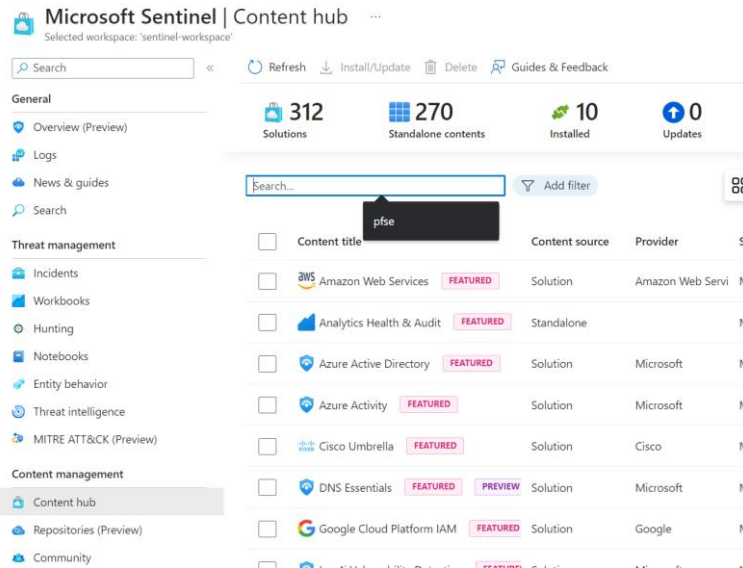


Figure 57: Content Hub

Within Data Connectors, you will find connectors related to the enabled content. Look for "Legacy Agent" and select "Security Events via Legacy Agent." This will provide guidance on sending logs to Windows using the agent.

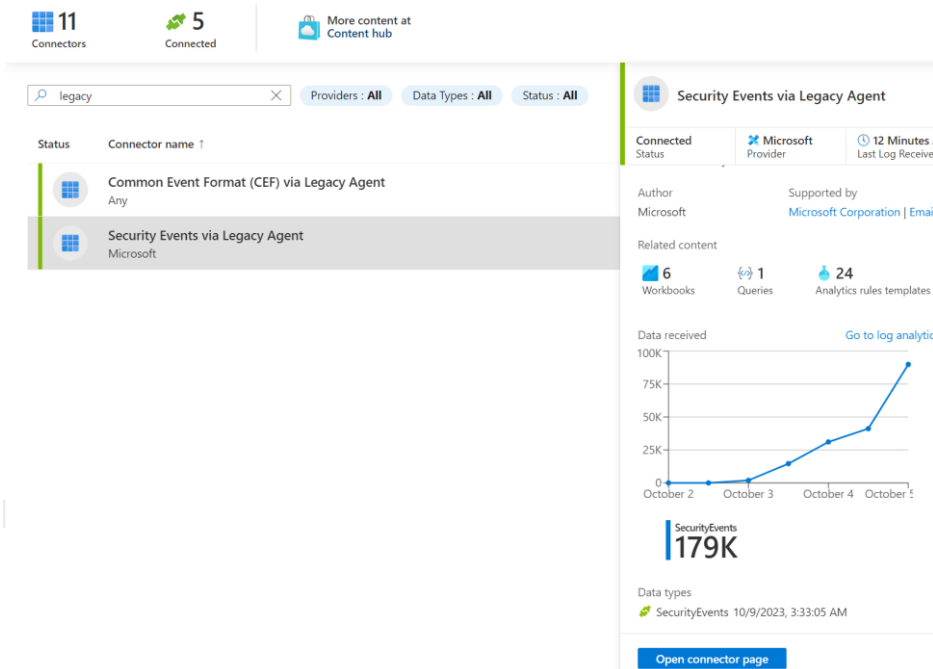


Figure 58: Connectors page

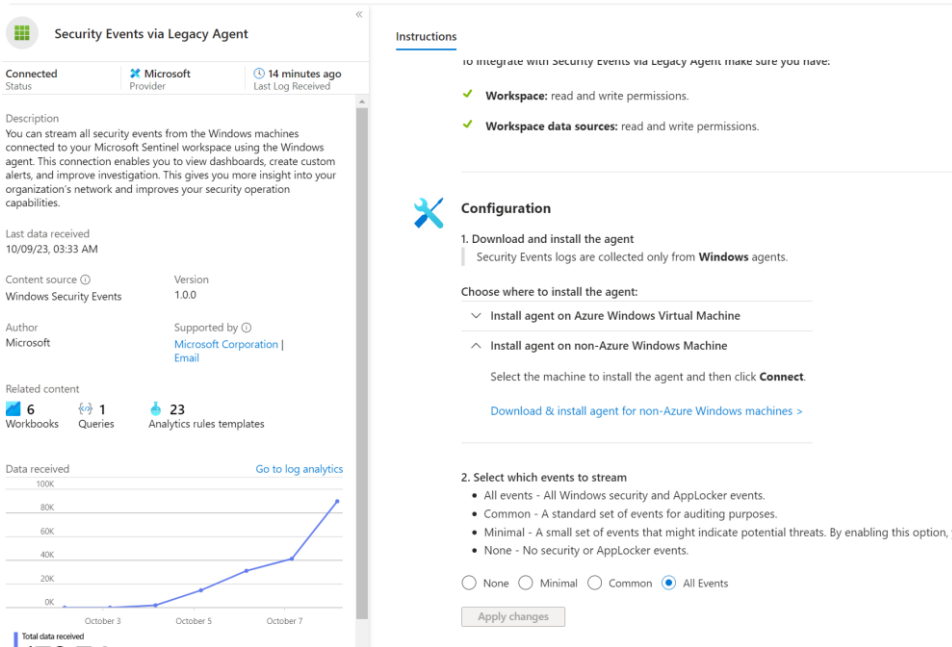


Figure 59: Security Events via Legacy Agent page

In the agent configuration, select all events and apply the changes at the bottom. Then, click on "Install agent on non-Azure Windows Machine" and "Download & install agent for non-Azure Windows machines."

Download the Microsoft Monitoring Agent executable from the provided link. Install this agent on your server. During installation, choose "Azure Log Analytics (OMS)" and add the Workspace ID and Primary Key obtained earlier.

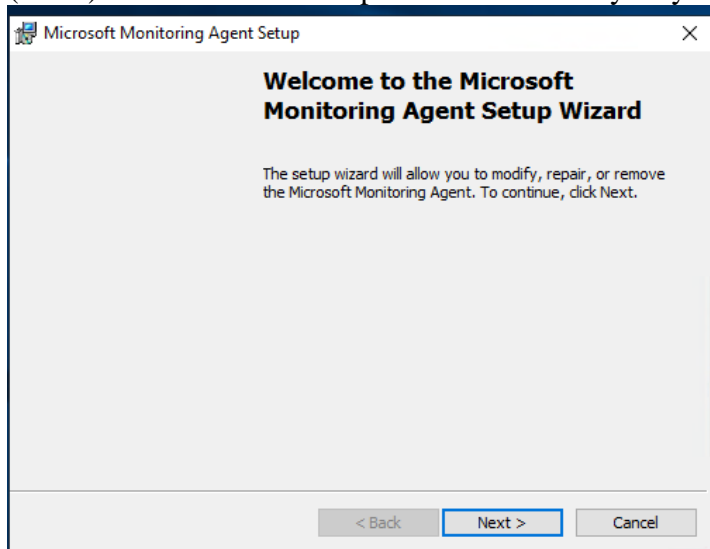


Figure 60: Microsoft Monitoring Agent installation

After completing the installation, the Microsoft Monitoring Agent will start sending logs to Azure Sentinel. To confirm the successful installation and communication with Azure, run the following query in the logs tab of Sentinel.

### Heartbeat | summarize count() by Computer

The screenshot shows the Azure Sentinel query editor with the following query:

```
1 Heartbeat
2 | summarize count() by Computer
```

The results are displayed in a table with the following data:

Computer	count_
> goad-VirtualBox	283
> kingslanding.sevenkingdoms.local	237
> winterfell.north.sevenkingdoms.local	234
> meereen.essos.local	235
> castelblack.north.sevenkingdoms.local	147
> braavos.essos.local	217
> DESKTOP-U1NNU7N.essos.local	32

Figure 61: Heartbeat received

Choose the desired time range, and if you see the Windows hosts in the list of Computers, the installation is successful.

To receive Sysmon logs and Windows Defender logs within Azure Sentinel, an additional configuration step is required:

In the Azure Sentinel portal, navigate to the "Settings" tab located in the left pane.

1. Under "Workspace Settings," select "Legacy Agent Management."
2. Within the "Legacy Agent Management" settings, click on 'Add Windows Event Log.'
3. In the dialog that appears, add the following two event logs:
  - a. Microsoft-Windows-Sysmon/Operational
  - b. Microsoft-Windows-Windows Defender/Operational
4. Save your changes.

The screenshot displays the 'Legacy agents management' page in the Azure Sentinel workspace. The left-hand navigation pane includes sections for 'Settings' (Tables, Agents, Usage and estimated costs, Data export, Network isolation, Linked storage accounts, Properties, Locks) and 'Classic' (Legacy agents management, Legacy activity log connector, Legacy storage account logs). The main content area is titled 'Windows event logs' and features a warning banner about the August 31, 2024 migration deadline. Below the banner, there is a section for 'Add windows event log' with a search filter. A table lists the configured logs:

Log name	Error	Warning	Information	
Microsoft-IIS-Logging/Logs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Microsoft-Windows-Sysmon/Operational	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Microsoft-Windows-Windows Defender/Operational	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure 62: Additional logs configured.

In summary, the implementation of SIEM technologies, including Wazuh and Azure Sentinel, has fortified our cyber range's security framework. By configuring and integrating these SIEM solutions, we've established a robust system for gathering, analyzing, and responding to security incidents. Wazuh, as an open-source SIEM, provides real-time threat visibility, while Azure Sentinel, a cloud-based SIEM, centralizes log data and enhances threat detection.

Installing Sysmon agents on Windows machines and optimizing domain controller logging increases our ability to spot security threats. These measures bolster our proactive defense and risk mitigation capabilities within the cyber range. With Wazuh and Azure Sentinel in place, we're well-prepared for the next phases of our cybersecurity strategy, focusing on threat detection, response, and mitigation. These SIEM tools serve as essential components of our mission to secure the cyber range effectively.

## 4.2 MODULAR ASSET MODELING & CPE EXTRACTION

In this section, we delve into the essential process of Modular Asset Modeling and Common Platform Enumeration (CPE) Extraction. Asset modeling is a fundamental practice in cybersecurity and system management, enabling organizations to gain insights into their hardware, software, and services inventory. Through the use of both Linux shell

scripts and PowerShell scripts, we will showcase how to systematically extract valuable information about installed applications, services, and hardware components.

These scripts will not only help in comprehensively documenting the assets within a system but also demonstrate how to represent this information in a standardized and universally recognized format - the Common Platform Enumeration (CPE). The CPE format is crucial for asset management, vulnerability assessment, and overall cybersecurity, as it enables a structured and consistent representation of system attributes.

By the end of this section, you will have a clear understanding of how to extract, format, and document crucial asset information, facilitating better security practices, asset management, and informed decision-making within your organization. Let's explore these scripts and the power of asset modeling and CPE extraction in enhancing the security and management of your computing environment.

#### **4.2.1 CPE Extraction – Linux Services**

The script starts by defining a function named `convert_to_cpe_format`. This function is crucial for converting service information into Common Platform Enumeration (CPE) format.

Using the `systemctl` command, the script retrieves a list of installed services. It specifies options such as `--type=service` to filter only services, `--all` to list all units (including inactive ones), `--plain` to output data in plain text, and `--no-legend` to omit the legend/header from the output.

The script creates two CSV files, `Installed_Services.csv` with a header line containing the column name "Name." and `Installed_Services_CPE.csv` with a header line containing the column name "CPE."

It iterates through the list of installed services, appending each service's name to the `Installed_Services.csv` file. For this script, other CPE attributes (version, update, etc.) are left empty, and commas indicate missing values.

In a second iteration, the script calls the `convert_to_cpe_format` function. It passes "Linux" as the vendor (indicating a Linux system) and the service name as the product. Other attributes in the CPE format are set to "\*", indicating missing or unknown values. The generated CPE URIs are then appended to the `Installed_Services_CPE.csv` file.



```

1  #!/bin/bash
2
3  # Function to convert service info to CPE format
4  function convert_to_cpe_format {
5      local part="a"
6      local vendor=$1
7      local product=$2
8      local version=$3
9      local update=$4
10     local edition=$5
11     local language=$6
12     local sw_edition=$7
13     local target_sw=$8
14     local target_hw=$9
15
16     # Remove commas from all attributes
17     vendor=${vendor//,/}
18     product=${product//,/}
19     version=${version//,/}
20     update=${update//,/}
21     edition=${edition//,/}
22     language=${language//,/}
23     sw_edition=${sw_edition//,/}
24     target_sw=${target_sw//,/}
25     target_hw=${target_hw//,/}
26
27     # Remove anything within "<code>" and "<code>"
28     vendor=${vendor//["<code>"]*/}
29     product=${product//["<code>"]*/}
30     version=${version//["<code>"]*/}
31     update=${update//["<code>"]*/}
32     edition=${edition//["<code>"]*/}
33     language=${language//["<code>"]*/}
34     sw_edition=${sw_edition//["<code>"]*/}
35     target_sw=${target_sw//["<code>"]*/}
36     target_hw=${target_hw//["<code>"]*/}
37
38     # Replace spaces with underscores in all attributes
39     vendor=${vendor// /_}
40     product=${product// /_}
41     version=${version// /_}
42     update=${update// /_}
43     edition=${edition// /_}
44     language=${language// /_}
45     sw_edition=${sw_edition// /_}
46     target_sw=${target_sw// /_}
47     target_hw=${target_hw// /_}
48
49     # Remove trailing spaces from all attributes
50     vendor=${vendor% }
51     product=${product% }
52     version=${version% }
53     update=${update% }
54     edition=${edition% }
55     language=${language% }
56     sw_edition=${sw_edition% }
57     target_sw=${target_sw% }
58     target_hw=${target_hw% }
59
60     # Remove the last character (if it is an underscore) from the product attribute
61     product=${product%_}
62
63     # Replace empty fields with ""
64     [ -z "$vendor" ] && vendor=""
65     [ -z "$product" ] && product=""
66     [ -z "$version" ] && version=""
67     [ -z "$update" ] && update=""
68     [ -z "$edition" ] && edition=""
69     [ -z "$language" ] && language=""
70     [ -z "$sw_edition" ] && sw_edition=""
71     [ -z "$target_sw" ] && target_sw=""
72     [ -z "$target_hw" ] && target_hw=""
73
74     echo "cpe:2.3:$part:$vendor:$product:$version:$update:$edition:$language:$sw_edition:$target_sw:$target_hw"
75 }
76
77 # Get installed services using Systemd
78 services=$(systemctl list-units --type=service --all --plain --no-legend | awk '{print $1}')
79
80 # Convert to CPE format and export to CSV with separate columns
81 echo "Name,Version,Update,Edition,Language,SwEdition,TargetSw,TargetHw" > Installed_Services.csv
82 echo "$services" | while read -r service; do
83     echo "$service,,,,,,,,," >> Installed_Services.csv
84 done
85
86 # Convert to CPE format and export to CSV with CPE in one column
87 echo "CPE" > Installed_Services_CPE.csv
88 echo "$services" | while read -r service; do
89     # Clean out commas and convert to CPE format with "" for empty attributes
90     cpe=$(convert_to_cpe_format "$service" "" "" "" "" "" "" "" "")
91
92     # Append the modified CPE URI to the CSV file
93     echo "$cpe" >> Installed_Services_CPE.csv
94 done
95

```

Figure 63: CPE extraction from Linux services.

#### 4.2.2 CPE Extraction – Linux Applications

Similar to the services script, this script also defines the `convert_to_cpe_format` function for converting application information into CPE format.

The script retrieves a list of installed applications using the **`dpkg-query`** command, which interacts with the Debian package manager's database. It specifies a custom format for outputting package information in a semicolon-separated format, including package name, version, architecture, and maintainer.

The script creates two CSV files, `Installed_Applications.csv` with a header line containing column names ("`Name`," "`Version`," "`Architecture`," and "`Maintainer`") and `Installed_Applications_CPE.csv` with a header line containing the column name "`CPE`."

It iterates through the list of installed applications, extracting the package name, version, architecture, and maintainer for each application. This data is appended as lines to the `Installed_Applications.csv` file.

Similar to the services script, in a second iteration, the script calls the `convert_to_cpe_format` function. It passes "`Linux`" as the vendor (indicating a Linux system) and the application name as the product. Other attributes in the CPE format are set to "`*`", indicating missing or unknown values. The generated CPE URIs are then appended to the `Installed_Applications_CPE.csv` file.

```

#!/bin/bash

# Function to convert application info to CPE format
function convert_to_cpe_format {
    local part="a"
    local vendor=$1
    local product=$2
    local version=$3
    local update=$4
    local edition=$5
    local language=$6
    local sw_edition=$7
    local target_sw=$8
    local target_hw=$9

    # Remove commas from all attributes
    vendor=${vendor//,/}
    product=${product//,/}
    version=${version//,/}
    update=${update//,/}
    edition=${edition//,/}
    language=${language//,/}
    sw_edition=${sw_edition//,/}
    target_sw=${target_sw//,/}
    target_hw=${target_hw//,/}

    # Replace space in vendor attribute with an underscore
    vendor=${vendor// /_}

    # Remove anything within "<" and ">"
    vendor=${vendor//<[^>]*>/}
    product=${product//<[^>]*>/}
    version=${version//<[^>]*>/}
    update=${update//<[^>]*>/}
    edition=${edition//<[^>]*>/}
    language=${language//<[^>]*>/}
    sw_edition=${sw_edition//<[^>]*>/}
    target_sw=${target_sw//<[^>]*>/}
    target_hw=${target_hw//<[^>]*>/}

    # Remove trailing spaces from vendor attribute
    vendor=${vendor%% }

    # Remove the last character (if it is an underscore) from the vendor attribute
    vendor=${vendor%_}

    # Replace empty fields with "*"
    [ -z "$vendor" ] && vendor="*"
    [ -z "$product" ] && product="*"
    [ -z "$version" ] && version="*"
    [ -z "$update" ] && update="*"
    [ -z "$edition" ] && edition="*"
    [ -z "$language" ] && language="*"
    [ -z "$sw_edition" ] && sw_edition="*"
    [ -z "$target_sw" ] && target_sw="*"
    [ -z "$target_hw" ] && target_hw="*"

    echo "cpe:2.3:$part:$vendor:$product:$version:$update:$edition:$language:$sw_edition:$target_sw:$target_hw"
}

# Get installed applications using APT package manager
applications=$(dpkg-query -W -f='${Package};${Version};${Architecture};${Maintainer}\n')

# Convert to CPE format and export to CSV with CPE in one column
echo "CPE" > Installed_Applications_CPE.csv
echo "$applications" | while IFS=';' read -r package version architecture maintainer; do
    # Clean out commas and then convert to CPE format with "*" for empty attributes
    cpe=$(convert_to_cpe_format "$maintainer" "$package" "$version" "*" "*" "*" "*" "*" "*")

    # Append the modified CPE URI to the CSV file
    echo "$cpe" >> Installed_Applications_CPE_4.csv
done

```

Figure 64: CPE Extraction for Linux Applications

### 4.2.3 CPE Extraction – Windows Applications

The script initially gathers application data by accessing specific uninstall keys within the Windows Registry. It systematically explores registry paths such as **HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall**, **HKLM:\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall**, and **HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall**.

For each application discovered in these registry paths, the script captures essential information including the application's name, version, installation location, and vendor.

This collected information is consolidated into an array, which serves as the foundation for the subsequent steps.

In parallel to the first method, the script also employs Windows Management Instrumentation (WMI) through the **Get-WmiObject** cmdlet to retrieve data about installed applications.

By querying the **Win32\_Product** class, the script extracts additional details about applications, such as their names, versions, installation locations, and vendors.

The data obtained through this method is integrated into the same array used for storing application information.

With both methods contributing to the array, the script proceeds to export this collected application data to CSV files.

In the first export step, the script generates a traditional CSV file named "Installed\_Applications.csv." This file segregates application details into distinct columns, including names, versions, installation locations, and vendors.

The second export step focuses on converting the acquired information into Common Platform Enumeration (CPE) format. For each application in the array, the script invokes the **Convert-ToCPEFormat** function, generating corresponding CPE URIs.

These CPE URIs are collated into a separate array named `$cpeApplications` and subsequently exported to a CSV file called "Installed\_Applications\_CPE.csv." This particular file features a single column titled "CPE," housing CPE URIs for each of the installed applications.

```

# Function to convert application info to CPE format
# Function to convert application info to CPE format
# Function to convert application info to CPE format
function Convert-ToCPEFormat {
    param (
        [string]$part,
        [string]$vendor,
        [string]$product,
        [string]$version,
        [string]$update,
        [string]$edition,
        [string]$language,
        [string]$swEdition,
        [string]$targetSw,
        [string]$targetHw
    )

    # Clean each attribute (replace invalid characters)
    $product = $product -replace ",, "
    $version = $version -replace ",, "
    $update = $update -replace ",, "
    $edition = $edition -replace ",, "
    $language = $language -replace ",, "
    $swEdition = $swEdition -replace ",, "
    $targetSw = $targetSw -replace ",, "
    $targetHw = $targetHw -replace ",, "

    # Concatenate attributes with colons and replace spaces with underscores
    $cpeAttributes = "${part}:${vendor}:${product}:${version}:${update}:${edition}:${language}:${swEdition}:${targetSw}:${targetHw}"
    $cpeAttributes = $cpeAttributes -replace " ", "_"
    $cpeAttributes = $cpeAttributes -replace ":", "."

    # Create the CPE URI
    $cpeUri = "cpe:2.3:${cpeAttributes}"

    return $cpeUri
}

# Define an array to store application information
$applications = @()

# Get installed applications using uninstall keys
$uninstallKeys = @(
    'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall',
    'HKLM:\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall',
    'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall'
)

foreach ($key in $uninstallKeys) {
    $uninstallData = Get-ChildItem $key | ForEach-Object {
        [PSCustomObject]@{
            Name = $_.GetValue("DisplayName")
            Version = $_.GetValue("DisplayVersion")
            InstallLocation = $_.GetValue("InstallLocation")
            Vendor = $_.GetValue("Publisher")
        }
    }
    $applications += $uninstallData
}

# Get installed applications using Win32_Product
$win32ProductData = Get-WmiObject -Class Win32_Product | ForEach-Object {
    [PSCustomObject]@{
        Name = $_.Name
        Version = $_.Version
        InstallLocation = $_.InstallLocation
        Vendor = $_.Vendor
    }
}

$applications += $win32ProductData

# Export to CSV with separate columns
$applications | Export-Csv -Path ".\Installed_Applications.csv" -NoTypeInformation

# Convert to CPE format and export to CSV with CPE in one column
$cpeApplications = $applications | ForEach-Object {
    $cpe = Convert-ToCPEFormat -part "a" -vendor $_.Vendor -product $_.Name -version $_.Version -update "" -edition "" -language "" -swEdition "" -targetSw "" -targetHw ""
    [PSCustomObject]@{
        CPE = $cpe
    }
}

$cpeApplications | Export-Csv -Path ".\Installed_Applications_CPE.csv" -NoTypeInformation

```

Figure 65: CPE Extraction - Windows Applications

#### 4.2.4 CPE Extraction – Windows Services

Convert-ToCPEFormat function takes a product name as a parameter and is responsible for converting service information into CPE format. It performs the following steps.

- Cleans the product name by replacing invalid characters (, and :).
- Constructs a CPE URI using the cleaned product name, leaving other CPE attributes as wildcard values.
- Returns the CPE URI.

The script retrieves a list of installed services using the **Get-WmiObject** cmdlet, specifically the **Win32\_Service** class. This cmdlet queries the Windows Management

Instrumentation (WMI) to gather information about services and an array named `$serviceData` is created to store the service data.

The script iterates through the list of services obtained in the previous step. For each service, it performs the following actions:

- Retrieves the display name of the service, which serves as the product name.
- Calls the `Convert-ToCPEFormat` function, passing the product name to obtain a CPE URI.
- Creates a custom PowerShell object containing two properties: `Product` (the service's display name) and `CPE` (the CPE URI).
- Appends this custom object to the `$serviceData` array.

After collecting and formatting the service information, the script exports it to a CSV file named "ServicesInfo.csv" using the `Export-Csv` cmdlet. The `-NoTypeInfo` parameter ensures that the CSV file doesn't include type information.

The script displays a confirmation message, "Services information exported to ServicesInfo.csv," to indicate that the task has been completed.

```

# Function to convert service info to CPE format
function Convert-ToCPEFormat {
    param (
        [string]$product
    )

    # Clean the product attribute (replace invalid characters)
    $product = $product -replace ",", ""
    $product = $product -replace ":", "_"

    # Create the CPE URI
    $cpeUri = "cpe:2.3:a:${product}:*:*:*:*:*"

    return $cpeUri
}

# Get a list of services using Get-WmiObject
$services = Get-WmiObject -Class Win32_Service

# Create an array to store the service data
$serviceData = @()

# Loop through the services and convert them to CPE format
foreach ($service in $services) {
    $product = $service.DisplayName

    # Convert to CPE format
    $cpeUri = Convert-ToCPEFormat -product $product

    # Create an object for the service
    $serviceObject = [PSCustomObject]@{
        Product = $product
        CPE = $cpeUri
    }

    # Add the service object to the array
    $serviceData += $serviceObject
}

# Export the service data to a CSV file
$serviceData | Export-Csv -Path "ServicesInfo.csv" -NoTypeInformation

Write-Host "Services information exported to ServicesInfo.csv"

```

Figure 66: CPE Extraction - Windows Services

## 4.3 SECURITY CONTROL INTEGRATION

### 4.3.1 Automated Penetration Testing Agents

After gaining initial access to one of the Windows machines in the GOAD environment through a Webshell vulnerability, we installed an automated penetration testing agent.

This agent was configured to simulate Active Directory (AD) attacks in the style of a known Advanced Persistent Threat (APT) group.

The results of this agent come in a detailed reporting format starting with an overview containing statistics concerning the attacks. We can see this overview in Figure 67

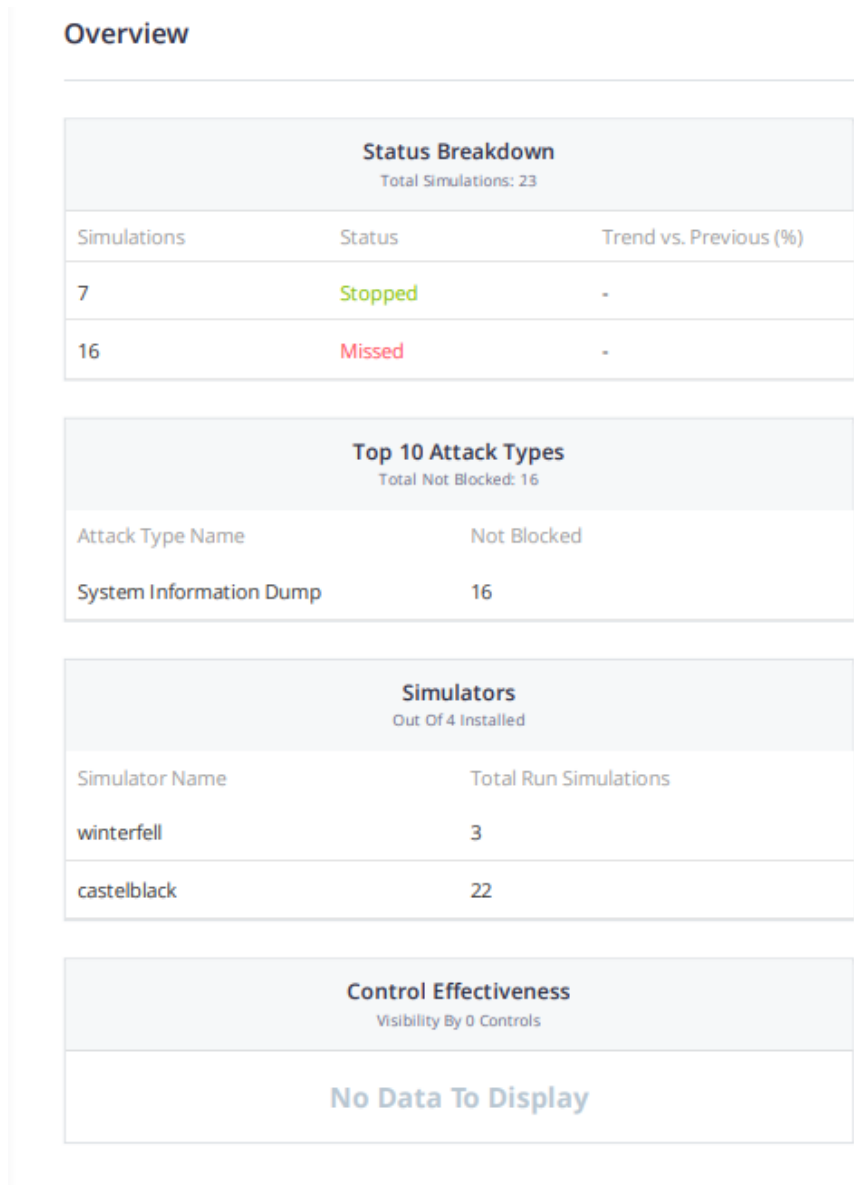


Figure 67: Overview of Penetration Testing



The report continues with the summary of the steps followed as seen in Figure 67.  
**Step Overview**

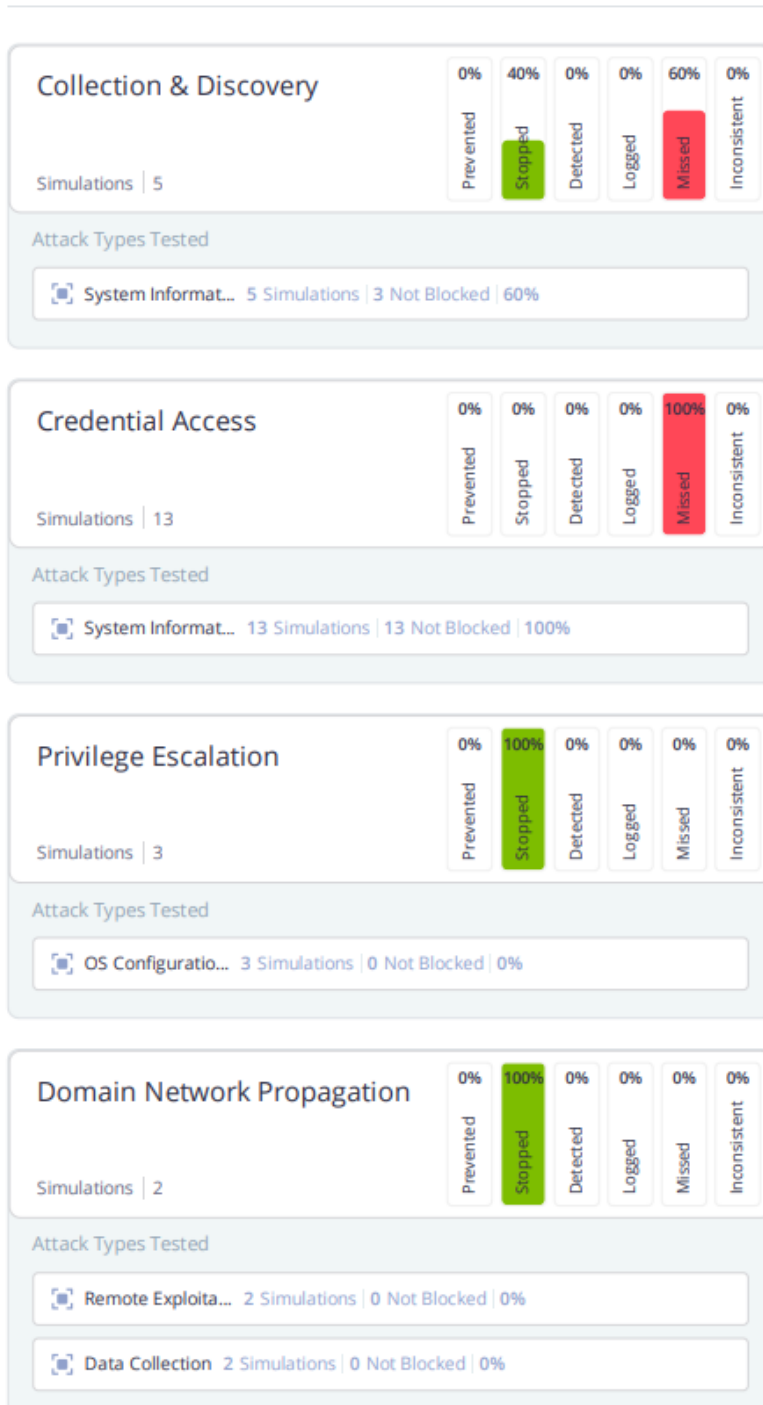
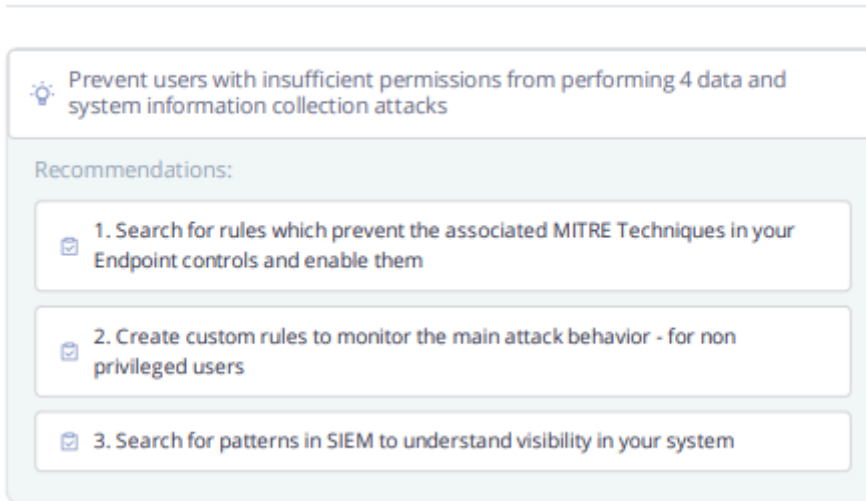


Figure 68: Summary of Penetration Testing steps

Finally, the report contains a summary of available Insight recommendations regarding the attacks that Passed.

### Number of available Insight Recommendations



The results of these attacks are also extracted in a csv file, the attributes that extracted from this output are displayed in the screenshot that follows.

0	Simulation ID	55 non-null	int64	41	Web App Target	0 non-null	float64
1	Status	55 non-null	object	42	Client HTTP headers	0 non-null	float64
2	Result	55 non-null	object	43	Server HTTP headers	0 non-null	float64
3	Result Code	55 non-null	object	44	Content type headers	0 non-null	float64
4	Result Details	55 non-null	object	45	Hash (SHA256)	26 non-null	object
5	Last Change (UTC)	23 non-null	object	46	Hash (SHA1)	26 non-null	object
6	Simulation Time (UTC)	55 non-null	object	47	Hash (MD5)	26 non-null	object
7	Attack ID	55 non-null	int64	48	Command	25 non-null	object
8	Attack Name	55 non-null	object	49	Content	0 non-null	float64
9	Attack Phase	55 non-null	object	50	Cookies	0 non-null	float64
10	Attack Type	55 non-null	object	51	Other	34 non-null	object
11	Attack Description	55 non-null	object	52	URL	0 non-null	float64
12	Attacker OS	55 non-null	object	53	Remediation	55 non-null	object
13	Target OS	55 non-null	object	54	Test	55 non-null	object
14	Attacker OS Version	55 non-null	int64	55	Test Time (UTC)	0 non-null	float64
15	Target OS Version	55 non-null	int64	56	Plan	0 non-null	float64
16	Source IP	2 non-null	object	57	Tags	0 non-null	float64
17	Destination IP	2 non-null	object	58	Industry	48 non-null	object
18	Threat Group	48 non-null	object	59	Campaign	0 non-null	float64
19	Port	2 non-null	float64	60	Category	55 non-null	object
20	Protocol	5 non-null	object	61	IOC/TTP	0 non-null	float64
21	Attacker Node	55 non-null	object	62	Threat Name	49 non-null	object
22	Direction	2 non-null	object	63	Threat Last Seen (UTC)	0 non-null	float64
23	Target Node	55 non-null	object	64	Target Countries	0 non-null	float64
24	Mitre ATT&CK Tactics	55 non-null	object	65	Attacker Countries	0 non-null	float64
25	MITRE Sub Technique	51 non-null	object	66	IOC Type	0 non-null	float64
26	Mitre ATT&CK Techniques	55 non-null	object	67	NIST Controls	50 non-null	object
27	Tools/Malware	0 non-null	float64	68	Malware Type	0 non-null	float64
28	Labels	0 non-null	float64	69	Security Action	55 non-null	object
29	Data Asset	0 non-null	float64	70	Visibility by product	0 non-null	float64
30	Security Control Category	55 non-null	object	71	Alerted by product	0 non-null	float64
31	Attacker Profile	0 non-null	float64	72	Alert Id	0 non-null	float64
32	Leak Rate	0 non-null	float64	73	Alert Name	0 non-null	float64
33	Footprint	0 non-null	float64				
34	Proxy	0 non-null	float64				
35	Impersonated User	51 non-null	object				
36	HOST	0 non-null	float64				
37	FQDN / IP	0 non-null	float64				
38	Registry	0 non-null	float64				
39	Path	28 non-null	object				
40	URI	0 non-null	float64				

Figure 69: Attributes of the attacks

In order to perform further analysis in our dataset we only maintain column that will assist us the creation of threat profiles as illustrated in the Figure below.

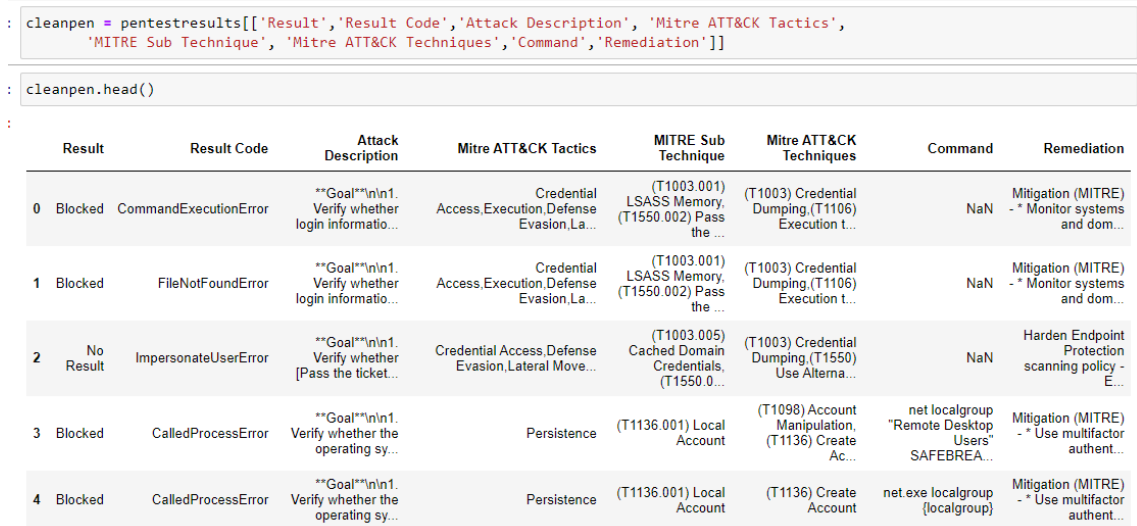


Figure 70: Interesting fields extracted.

Finally, we remove any threats that were automatically blocked in order to prioritize on the active threats inside the environment as illustrated in the below figure, we notice that from 55 attempts 16 were successful.



Figure 71: Successful attacks.

To create direct connection between this data and with the extensive list of Common Platform Enumeration (CPE) products we identified throughout the system, we'll need to assess this dataset by hand and offer a security expert's opinion about the targeted software of each implemented threat. This procedure can be automated through the use of AI, targeted towards conceptual connections among related strings and it would be interesting future work. This step is crucial for gaining a deeper understanding of the security landscape within the GOAD environment.

### 4.3.2 Exploitable Threats in MITRE Attack Format

Derived from the data parsing that we performed earlier we get 2 columns containing the ATTACK Tactics & Techniques utilized by this attack instance.

```

: clean['Mitre ATT&CK Tactics']
: 11 Credential Access
: 13 Credential Access
: 15 Credential Access
: 18 Credential Access
: 19 Credential Access
: 20 Credential Access
: 21 Credential Access
: 24 Credential Access
: 26 Execution,Credential Access
: 28 Credential Access
: 32 Credential Access
: 35 Credential Access
: 37 Credential Access
: 41 Discovery
: 48 Discovery
: 54 Discovery
Name: Mitre ATT&CK Tactics, dtype: object

: clean['Mitre ATT&CK Techniques']
: 11 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 13 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 15 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 18 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 19 (T1003) Credential Dumping
: 20 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 21 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 24 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 26 (T1059) Command-Line Interface,(T1558) Steal or Forge Kerberos Tickets
: 28 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 32 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 35 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 37 (T1003) Credential Dumping,(T1552) Unsecured Credentials
: 41 (T1087) Account Discovery,(T1069) Permission Groups Discovery
: 48 (T1087) Account Discovery,(T1069) Permission Groups Discovery
: 54 (T1087) Account Discovery,(T1069) Permission Groups Discovery
Name: Mitre ATT&CK Techniques, dtype: object

```

Figure 72: MITRE ATTACK Tactics & Techniques

### 4.3.3 Building Custom Threat Profiles

In the output there are also included detailed attack descriptions along with a list of executed commands. By studying this output, we can create conceptual relationships to the CPEs that might have been affected from the executed commands.

We study the applications and services along with their related executables and derive any connections or references found in the ATTACK description and command columns of the csv file extracted by the automated penetration testing.

### 4.3.4 Detection Engineering with Sentinel

Taking the list of the successful attack techniques throughout our system we derive a set of security controls from the MITRE DEFEND framework and implement detection rules

to monitor similar future attacks. This is implemented by search the mappings of ATTACK techniques within the DEFEND matrix as illustrated in the figure below.

### D3FEND Inferred Relationships

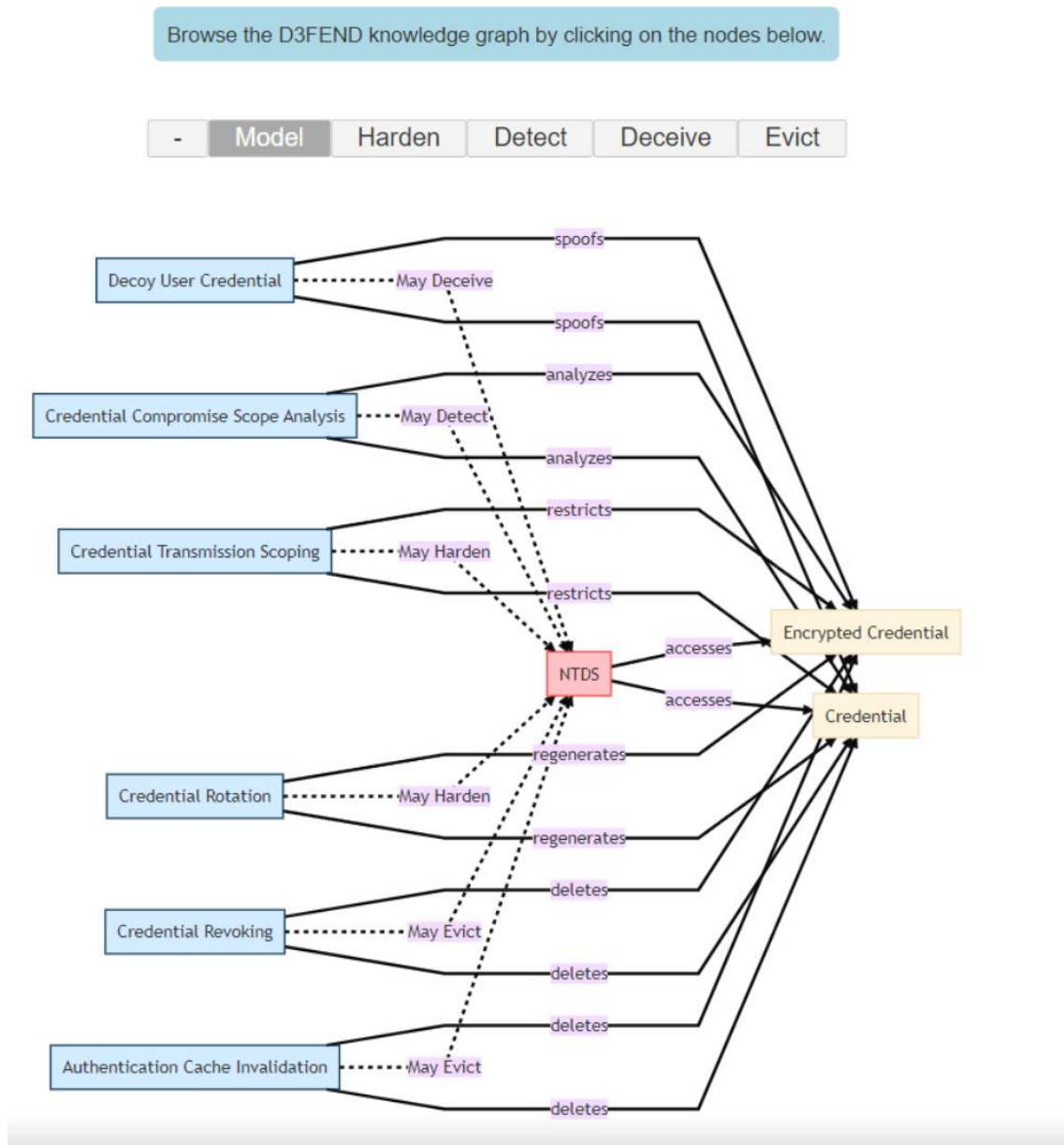


Figure 73: Credential Compromise Scope Analysis May Detect NTDS

Having derived the security controls appropriate for each technique we implement controls related to detection in the form of analytic rules.

Active rules   Rule templates   Anomalies

Search by ID, name, tactic or technique   Add filter

<input type="checkbox"/>	Severity	Name	Rule t...	Status	Tactics	Techniques	Source name
<input type="checkbox"/>	Medium	Suspicious Process Execution from Temp directory	Scd	Enabled	Execution	T1059 +3	Custom Content
<input type="checkbox"/>	Medium	Suspicious number of local reconnaissance commands	Scd	Enabled	Discovery	T1087 +9	Custom Content
<input type="checkbox"/>	Low	Possible Port Scan Detected	Scd	Enabled	Discovery	T1046	Custom Content
<input type="checkbox"/>	High	Webshell Detection	Scd	Enabled	Persistence	T1505	Custom Content
<input checked="" type="checkbox"/>	High	Suspicious Process accessing LSASS process	Scd	Enabled	Credential ...	T1003	Custom Content
<input type="checkbox"/>	High	Possible DCSync Attack Detected	Scd	Enabled	Credential ...	T1003	Custom Content
<input type="checkbox"/>	Medium	Kerberos Tickets requested for multiple accounts from the same host	Scd	Enabled	Cre... +1	T1558 +1	Custom Content
<input type="checkbox"/>	High	Local Privilege Escalation to SYSTEM	Scd	Enabled	Privilege Es...	T1548 +3	Custom Content
<input type="checkbox"/>	Medium	Kerberoasting Detection - RC4 Encryption	Scd	Enabled	Credential ...	T1558	Custom Content
<input type="checkbox"/>	Medium	Suspicious Registry Run Keys / Startup Folder Files modification	Scd	Enabled	Persistence	T1547	Custom Content
<input type="checkbox"/>	Medium	Multiple Object Accessed by same account	Scd	Enabled	Discovery	T1087 +1	Custom Content
<input type="checkbox"/>	High	Advanced Multistage Attack Detection	Fu:	Enabled	Col... +11		Gallery Content

Figure 74: Analytic rules in Azure Sentinel.

## 5 Testing

In the testing phase, our primary objective is to validate our hypothesis by mapping the successful attack techniques identified by the Automated Penetration Agents to the Common Platform Enumeration (CPE) components extracted by our PowerShell scripts.

Upon reviewing the results of the penetration testing, we observed one successful attack that managed to bypass the existing security controls. This attack occurred on the Domain Controller named Winterfell and specifically exploited **(T1003) Credential Dumping** with **Sub Technique (T1003.003) NTDS**.

E	I	X	Y	Z	AA
Result Details	Attack Name	Target Node	Mitre ATT&CK Tactics	Mitre Sub Technique	Mitre ATT&CK Techniques
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	NTDS.dit dump using nt	winterfell	Credential Access	(T1003.003) NTDS	(T1003) Credential Dumping
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Extract NTLM Hashes us	castelblack	Execution,Credential Ac	(T1059.001) PowerShell,(T1558.003) Kerberoasting	(T1059) Command-Line Interface,(T1558) Steal or Forge Kerberos TIC
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Windows Credentials	castelblack	Credential Access	(T1003.001) LSASS Memory,(T1003.004) LSA Secrets,(T1003.005) Cached Domain Cr	(T1003) Credential Dumping,(T1552) Unsecured Credentials
The simulation was not	Extract users and group	castelblack	Discovery	(T1087.001) Local Account,(T1087.002) Domain Account,(T1069.002) Domain Group;(T1087) Account Discovery,(T1069) Permission Groups Discovery	(T1087) Account Discovery,(T1069) Permission Groups Discovery
The simulation was not	Extract users and group	castelblack	Discovery	(T1087.001) Local Account,(T1087.002) Domain Account,(T1069.002) Domain Group;(T1087) Account Discovery,(T1069) Permission Groups Discovery	(T1087) Account Discovery,(T1069) Permission Groups Discovery
The simulation was not	Extract users and group	castelblack	Discovery	(T1087.001) Local Account,(T1087.002) Domain Account,(T1069.002) Domain Group;(T1087) Account Discovery,(T1069) Permission Groups Discovery	(T1087) Account Discovery,(T1069) Permission Groups Discovery

Figure 75: NTDSUTIL Attack

Path	Command
%temp%\alskdjfldsk.exe	N/A
%temp%\alskdjfldsk.exe	N/A
%temp%\alskdjfldsk.exe	N/A
%temp%\alskdjfldsk.exe	N/A
C:\Windows\System32\ntdsutil.exe,%temp%\SBTemp	ntdsutil.exe 'ac i ntds' 'ifm' 'create full {dump_path}' q q
%temp%\alskdjfldsk.exe	N/A
%temp%\alskdjfldsk.exe	N/A
%temp%\alskdjfldsk.exe	N/A
%temp%\sbkrb.ps1	Import-Module %temp%/sbkrb.ps1; Invoke-Kerberoast

Figure 76: NTDSUTIL command

The NTDSUTIL attack is a well-known method used to target Active Directory services, exposing all credentials stored within the AD on a Domain Controller.

Analyzing the data from our PowerShell script, which enumerates services on the Winterfell host, we discovered a direct match between specific services unique to Domain Controllers and the attack technique that exclusively affects Domain Controllers.

```

ServicesInfo - Notepad
File Edit Format View Help
"Product", "CPE"
"Microsoft Monitoring Agent Audit Forwarding", "cpe:2.3:a:Microsoft Monitoring Agent Audit Forwarding:*:*:*:*:*"
"Active Directory Web Services", "cpe:2.3:a:Active Directory Web Services:*:*:*:*:*"
"AllJoyn Router Service", "cpe:2.3:a:AllJoyn Router Service:*:*:*:*:*"
"Application Layer Gateway Service", "cpe:2.3:a:Application Layer Gateway Service:*:*:*:*:*"
"Application Identity", "cpe:2.3:a:Application Identity:*:*:*:*:*"
"Application Information", "cpe:2.3:a:Application Information:*:*:*:*:*"
"Application Management", "cpe:2.3:a:Application Management:*:*:*:*:*"
"App Readiness", "cpe:2.3:a:App Readiness:*:*:*:*:*"
"Microsoft App-V Client", "cpe:2.3:a:Microsoft App-V Client:*:*:*:*:*"
"AppX Deployment Service (AppXSVC)", "cpe:2.3:a:AppX Deployment Service (AppXSVC):*:*:*:*:*"
"Windows Audio Endpoint Builder", "cpe:2.3:a:Windows Audio Endpoint Builder:*:*:*:*:*"
"Windows Audio", "cpe:2.3:a:Windows Audio:*:*:*:*:*"
"ActiveX Installer (AxInstSV)", "cpe:2.3:a:ActiveX Installer (AxInstSV):*:*:*:*:*"
"Base Filtering Engine", "cpe:2.3:a:Base Filtering Engine:*:*:*:*:*"
"Background Intelligent Transfer Service", "cpe:2.3:a:Background Intelligent Transfer Service:*:*:*:*:*"
"Background Tasks Infrastructure Service", "cpe:2.3:a:Background Tasks Infrastructure Service:*:*:*:*:*"
"Bluetooth Audio Gateway Service", "cpe:2.3:a:Bluetooth Audio Gateway Service:*:*:*:*:*"
"AVCTP service", "cpe:2.3:a:AVCTP service:*:*:*:*:*"
"Bluetooth Support Service", "cpe:2.3:a:Bluetooth Support Service:*:*:*:*:*"
"Capability Access Manager Service", "cpe:2.3:a:Capability Access Manager Service:*:*:*:*:*"

```

Figure 77: Services Output: Active Directory Web Services

```

"Windows Camera Frame Server", "cpe:2.3:a:Windows Camera Frame Server:*:*:*:*:*"
"Group Policy Client", "cpe:2.3:a:Group Policy Client:*:*:*:*:*"
"GraphicsPerfSvc", "cpe:2.3:a:GraphicsPerfSvc:*:*:*:*:*"
"Microsoft Monitoring Agent", "cpe:2.3:a:Microsoft Monitoring Agent:*:*:*:*:*"
"Human Interface Device Service", "cpe:2.3:a:Human Interface Device Service:*:*:*:*:*"
"HV Host Service", "cpe:2.3:a:HV Host Service:*:*:*:*:*"
"Windows Mobile Hotspot Service", "cpe:2.3:a:Windows Mobile Hotspot Service:*:*:*:*:*"
"IKE and AuthIP IPsec Keying Modules", "cpe:2.3:a:IKE and AuthIP IPsec Keying Modules:*:*:*:*:*"
"Microsoft Store Install Service", "cpe:2.3:a:Microsoft Store Install Service:*:*:*:*:*"
"IP Helper", "cpe:2.3:a:IP Helper:*:*:*:*:*"
"Intersite Messaging", "cpe:2.3:a:Intersite Messaging:*:*:*:*:*"
"Kerberos Key Distribution Center", "cpe:2.3:a:Kerberos Key Distribution Center:*:*:*:*:*"
"Microsoft Key Distribution Service", "cpe:2.3:a:Microsoft Key Distribution Service:*:*:*:*:*"
"CNG Key Isolation", "cpe:2.3:a:CNG Key Isolation:*:*:*:*:*"
"KDC Proxy Server service (KPS)", "cpe:2.3:a:KDC Proxy Server service (KPS):*:*:*:*:*"
"KtmRm for Distributed Transaction Coordinator", "cpe:2.3:a:KtmRm for Distributed Transaction Coordinator:*:*:*:*:*"
"Server", "cpe:2.3:a:Server:*:*:*:*:*"
"Workstation", "cpe:2.3:a:Workstation:*:*:*:*:*"
"Geolocation Service", "cpe:2.3:a:Geolocation Service:*:*:*:*:*"
"Windows License Manager Service", "cpe:2.3:a:Windows License Manager Service:*:*:*:*:*"
"Link-Layer Topology Discovery Mapper", "cpe:2.3:a:Link-Layer Topology Discovery Mapper:*:*:*:*:*"
"TCP/IP NetBIOS Helper", "cpe:2.3:a:TCP/IP NetBIOS Helper:*:*:*:*:*"

```

Figure 78: Services Output: Kerberos Key Distribution Center





# NTDS - T1003.003

(ATT&CK® Technique)

## D3FEND Inferred Relationships

Browse the D3FEND knowledge graph by clicking on the nodes below.

- Model Harden Detect Deceive Evict

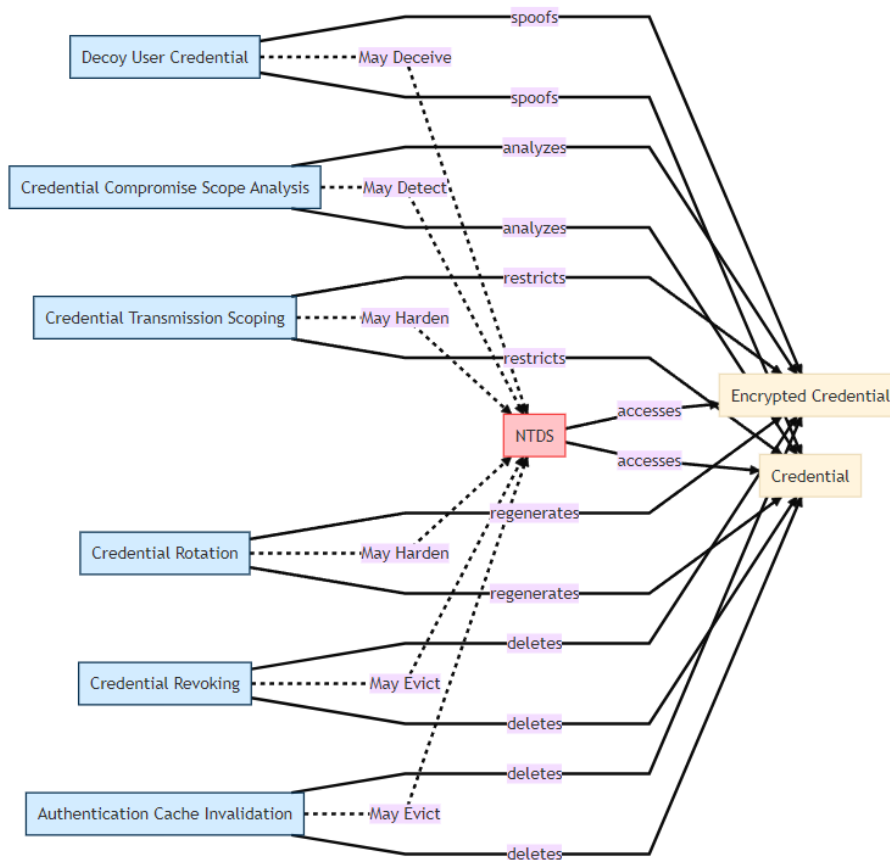


Figure 80: Credential Compromise Scope Analysis May Detect NTDS

To conclude our test, we strengthened our security controls within the Cyber Range environment to address this specific attack technique. We accomplished this by creating

### an analytic rule in Azure Sentinel and evaluating its effectiveness using existing data. Analytics rule wizard - Create a new Scheduled rule

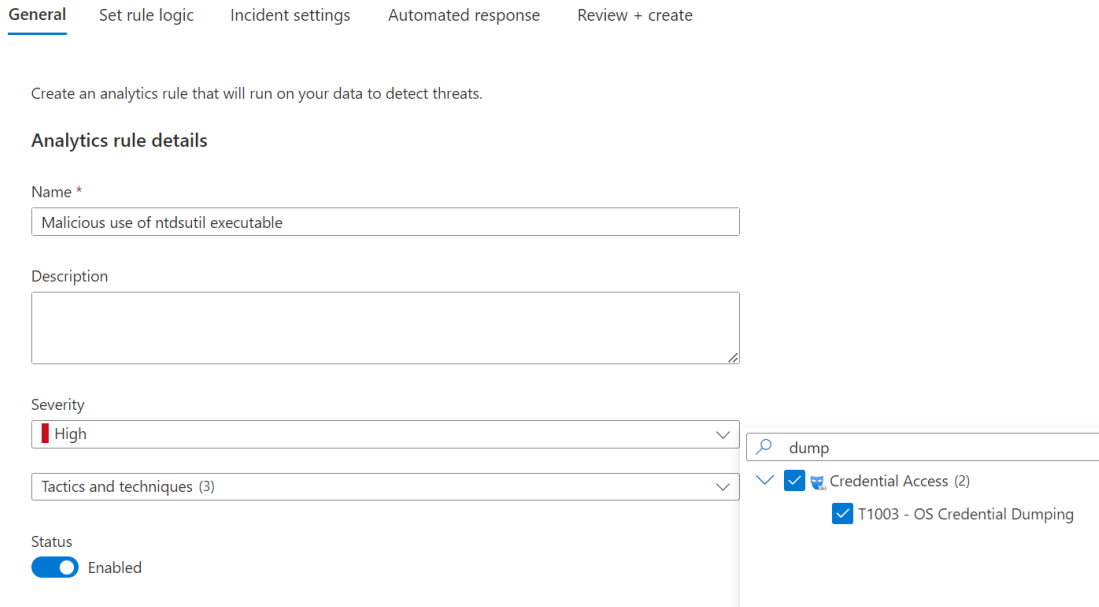


Figure 81: Mapping of the Analytic rule to the MITRE ATTACK Techniques

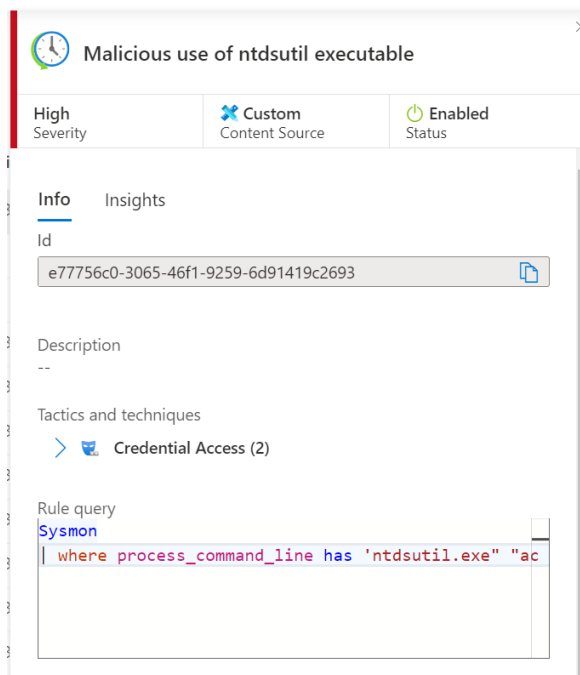
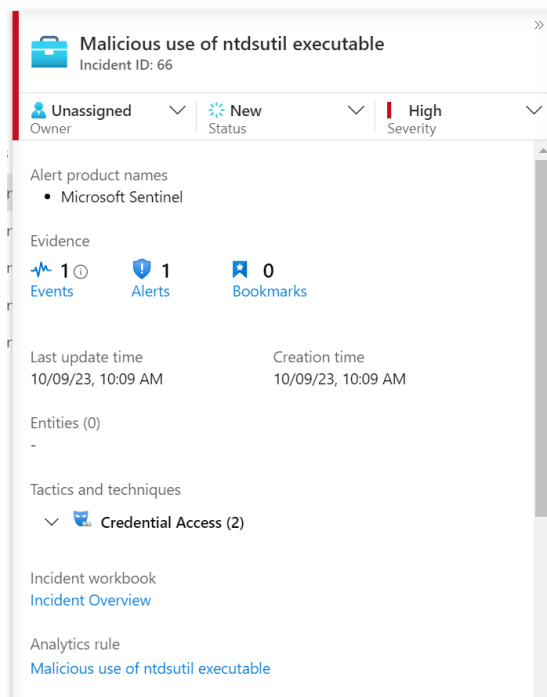


Figure 82: Analytic rule created.



*Figure 83: Incident triggered.*

In summary, our test successfully correlated the attack technique with the CPE components associated with the relevant services. We also aligned it with the defensive measures outlined in the DEFEND framework. Lastly, we implemented a security control to enhance the overall security posture of the environment.

## 6 Conclusions & Future work

Throughout this thesis we explored current asset inventory methodologies and tools along with their correlated threat profiling implementations. To present a complete methodology that supports the automation of such actions, we defined three distinct phases to our methodology and implementation: 1) Cyber Range development, 2) Modular Asset Modelling and CPE Extraction and 3) Security Control Integration.

Throughout the cyber range development, we designed and implemented an environment containing devices with varying operating systems and ensured to properly monitor their system logs along with potential network traffic using Intrusion Detection Systems, Firewalls and host level logging mechanisms, which in turn forward their data to centralized open source and commercial Security Information Event Management systems (Wazuh & Sentinel).

Using this cyber range as a testing bed we moved forward to the next step of our methodology, the modular asset inventory modelling, where we developed and deployed bash and PowerShell scripts, to derive asset inventories from Linux and windows machines by utilizing native commands. To present our results in a commonly accepted format we performed some further string transformations and brought the harvested inventories in CPE Uri format version 2.3. Some assets enumerated were known to the CPE catalogue, while others could not be matched.

In the final step of our methodology, we deployed automated penetration agents to verify attacks against both known and unknown assets, and hence build custom profiles. Using the combined output of the agents and the asset inventory we derive ATT&CK techniques used and assign them to the newly created or existing CPEs they were deployed against; this matching is performed based on a security experts' opinion that studies the logs in the SIEM. This procedure yields a custom threat profile, a critical step towards identifying attack paths against unidentified assets and vulnerabilities, and thus securing them before installing them in working infrastructures as a final step we match the ATT&CK techniques to their correlated mitigations that reside in the DEF3ND catalogue and implement Detection Measures to get alerts in case of similar future attempts.

A next step to our methodology would be the optimization of a matching algorithm that will perform thorough searches in the CPE catalogue and ensure that the custom CPEs enumerated don't already have a threat profile, with a slightly different product name. Utilizing such a functionality an automated mechanism can be fine-tuned and shared through an open repository so that various organizations can map existing

assets in their environments, identify their threat profiles and then share them to a common open repository. Utilizing such a sharing mechanism common threat profiles can be identified in multiple working environments, hence validating those entries. This procedure could also be implemented directly to open-source databases such as the CPE catalogue to shorten the time a component takes to be submitted to the official repository and in turn the time its threat profile takes to be built and uploaded.

Furthermore, the matching of the attack technique to the custom assets can be automated as a procedure through further scripting and through the deployment of expert models that can simulate red team operations in this specific pipeline. A great optimization on this matter could be the derivation of specific threats that repeatedly work against specific types of assets derived from specific vendors, by fine graining the procedure, automated penetration testing can be executed in a shorter timeframe by not initiating models known to fail against specific types of assets.

Finally deriving the possible mitigation and enforcing controls can be automated by creating predefined Firewall, IDPS and SIEM detection and prevention rules assigned to specific DEF3ND technique implementations. DEF3ND techniques are already mapped to the ATT&CK techniques they mitigate while ATT&CK techniques can be automatically identified by existing security solutions this creates a solution that can deal with attacks on runtime. This output can be further enhanced by optimizing mitigation solutions that might overlap to save resources.

## 7 References

- [1] [Online]. Available: <https://cpe.mitre.org/>.
- [2] [Online]. Available: <https://cwe.mitre.org/about/index.html>.
- [3] MITRE, "CAPEC," [Online]. Available: <https://capec.mitre.org/>.
- [4] [Online]. Available: <https://cve.mitre.org/>.
- [5] [Online]. Available: <https://www.cve.org/About/Process#CVERecordLifecycle>.
- [6] CVSS, "Common Vulnerability Scoring System," [Online]. Available: <https://www.first.org/cvss/>.
- [7] [Online]. Available: <https://www.first.org/cvss/specification-document>.
- [8] "Exploit Prediction Scoring System," [Online]. Available: <https://www.first.org/epss/>.
- [9] MITRE, "Threat Groups," [Online]. Available: <https://attack.mitre.org/groups/>.
- [10] [Online]. Available: <https://attack.mitre.org/>.
- [11] M. DEFEND. [Online]. Available: <https://d3fend.mitre.org/>.
- [12] P. a. L. X. a. C. E. a. S. B. a. M. C. a. F. K. a. S. D. Gao, "A system for automated open-source threat intelligence gathering and management," in *Proceedings of the 2021 International Conference on Management of Data*, 2021.
- [13] J. K. M. S.-R. B. R. K. X. N. R. U.-M. O. Erik Hemberg, "Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting," in *Cryptography and Security*.
- [14] [Online]. Available: <https://nvd.nist.gov/>.
- [15] V. a. B. S. Mavroeidis, "Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence," *arXiv*, 2023.
- [16] D. a. M. F. a. B. T. a. P. G. Schlette, "Security Enumerations for Cyber-Physical Systems".
- [17] I. a. D. E. a. F. A. a. D. V. Kotenko, "Automation of Asset Inventory for Cyber Security: Investigation of Event Correlation-Based Technique," *Electronics*, 2023.
- [18] Y. a. J. M. A. a. D. J. a. S. E. Jiang, "Model-Based Cybersecurity Analysis: Extending Enterprise Modeling to Critical Infrastructure Cybersecurity," 2023.

- [19] Iainfoulds, "Audit policy recommendations," [Online]. Available: <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations>.
- [20] Batamig, "Configure windows event collection - Microsoft defender for identity," [Online]. Available: <https://learn.microsoft.com/en-us/defender-for-identity/configure-windows-event-collection#configure-object-auditing>.
- [21] Markruss, "Sysmon - Sysinternals," [Online]. Available: <https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>.
- [22] Orange-Cyberdefense, "Orange-Cyberdefense/Goad: Game of Active Directory," [Online]. Available: <https://github.com/Orange-Cyberdefense/GOAD>.
- [23] [Online]. Available: <https://www.pfsense.org/download/>.
- [24] [Online]. Available: <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise>.
- [25] Canonical. [Online]. Available: <https://releases.ubuntu.com/20.04.6/>.
- [26] Olafhartong, "Olafhartong/sysmon-modular: A repository of sysmon configuration modules," [Online]. Available: <https://github.com/olafhartong/sysmon-modular>.
- [27] Wazuh, "Wazuh Virtual machine (OVA)," [Online]. Available: <https://documentation.wazuh.com/current/deployment-options/virtual-machine/virtual-machine.html>.
- [28] [Online]. Available: <https://azure.microsoft.com/en-us/products/microsoft-sentinel>.
- [29] [Online]. Available: <https://portal.azure.com/>.