Enhanced Threat Detection using Security Orchestration Automation Response and Endpoint Detection Response

A PROJECT REPORT

Submitted by

Harshitaa Ashish 21BCY10123
Neha Lakshmanan 21BCY10128
Praise E Mathew 21BCY10193
Pradyumn Padiyar 21BCY10009
Manish Kumar M 21BCY10196

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING
(Cyber Security and Digital Forensics)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

April 2025

BONAFIDE CERTIFICATE

Certified that this project report titled **"Enhanced Threat Detection using Security Orchestration Automation Response and Endpoint Detection Response"** is the bonafide work of Harshitaa Ashish (21BCY10123),Neha Lakshmanan (21BCY10128),Praise E Mathew (21BCY10193), Pradyumn Padiyar (21BCY10009), Manish Kumar M(21BCY10196) who carried out the Capstone Project-DSN4092 under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR,
Dr. D. Saravanan,
Assistant Professor Sr.,
School of Computing Science and Engineering,
VIT Bhopal University.

PROJECT SUPERVISOR,
Dr.Harihara Sitaraman S,
Associate Professor,
School of Computing Science and Engineering,
VIT Bhopal University.

The Capstone Project Examination is held on _____

ACKNOWLEDGEMENT

I First and foremost like to say a big thank you to the Lord Almighty for His presence and immense blessings that I received throughout the project work.

Moreover, I would like to express my heartfelt gratitude to Dr D. Saravanan, Lead Programme Chair, Division Head, Dr Adarsh Patel, Programme Chair Cyber Security and Digital Forensics for providing the tremendous encouragement throughout the course of this work.

I would like to thank Dr. Harihara Sitaraman S, my internal guide,who procured valuable suggestions for work completion during the Project.

I would like to thank all the technical and teaching staff of the School Computing Science and Engineering, for their direct and indirect support..

What I lacked in many other ways ,I made up in loyalty, determination and this is the last, but not the least,my sane thanks my parents that without their support I have worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

PIN	Personal Identification Number
lsass.exe	Local Security Authority Subsystem Service
MRT.exe	Microsoft Removal Tool
USB	Universal Serial Bus
SOC	Security Operations Center
EDR	Endpoint Detection Response
SIEM	Security Information and Event Management
SOAR	Security Orchestration, Automation, and Response
HKCU	HKEY_CURRENT_USER (Windows Registry Hive)
WMI	Windows Management Instrumentation
WMIC	Windows Management Instrumentation Command-line
DNS	Domain Name System
VSS	Volume Shadow Copy Service

LIST OF FIGURES AND GRAPHS

LIST OF TABLES

ABSTRACT

Cyber related threats are increasing in a massive way, making it a necessity to counter it using proactive and automated security solutions.Our project, “EDR-SOAR Integration using Limacharlie and Tines” is focusing on enhancing endpoint security by integrating Endpoint Detection and Response(EDR) with Security Orchestration,Automation, and Response (SOAR) functionality and capability.

LimaCharlie is a modern EDR platform that provides real-time telemetry, rule-based threat detection, and automated remediation capabilities. It allows for custom detection rules to monitor endpoint activities which helps in detecting and also mitigating the possible threats.

Tines is a powerful SOAR tool that automates incident response by executing predefined workflows. It processes security events from LimaCharlie, triggering appropriate actions to contain threats efficiently.

We have conducted this project by taking LimaCharlie as a real-time endpoint monitoring and Tines for automated incident response, making it effective in detecting ,analyzing and mitigating security threats with minimum effort manually.

Our project implements various detection rules to address critical security concerns, which mainly focuses on an attack which might arise from a ransomware attacker. Our project demonstrates a scalable and adaptable security solution that can be applied across organizations to strengthen their security solution that can be applied across organizations to strengthening their defenses against the cyber threats

TABLE OF CONTENTS

CHAPTER-1: PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

In today's rapidly evolving cybersecurity landscape, efficient threat detection and automated incident response are critical for organizations to protect their digital assets. This project focuses on integrating Endpoint Detection and Response (EDR) with Security Orchestration, Automation, and Response (SOAR) using LimaCharlie and Tines.

LimaCharlie provides real-time endpoint monitoring and threat detection, while Tines automates security workflows and response actions. By integrating these two technologies, we aim to create an automated cybersecurity framework that minimizes human intervention while enhancing security operations.

1.2 Motivation for the Work

Cyber threats are becoming more sophisticated, and traditional security solutions are often slow and reactive. Security teams face challenges such as:

- High alert volumes, leading to alert fatigue
- Slow response times, increasing incident impact
- Lack of automation, requiring manual intervention

To address these challenges, we propose an integrated EDR-SOAR system that can:

- Detect threats in real-time (LimaCharlie)
- Automate incident response (Tines)
- Reduce alert fatigue through intelligent automation

This integration enhances efficiency, accuracy, and scalability in cybersecurity operations.

1.3 About the Project and Techniques Used

This project combines EDR and SOAR to create an automated security framework.

- LimaCharlie (EDR) – Monitors endpoints, detects threats, and logs security events.
- Tines (SOAR) – Automates security workflows, incident response, and threat intelligence correlation.

- Integration Mechanism – API-based communication between LimaCharlie and Tines to automate incident detection and response.

By leveraging automation, real-time monitoring, and cloud-native security solutions, the project improves cyber resilience and response efficiency.

1.4 Problem Statement

Traditional cybersecurity solutions often fail to provide timely detection and response due to:
Manual intervention requirements
High false positive rates
Limited automation in incident handling

To solve these issues, this project integrates LimaCharlie with Tines to:

- Enable real-time threat detection
- Automate incident response workflows
- Reduce manual workload on security teams

The project's goal is to provide a more efficient and automated cybersecurity framework.

1.5 Objectives of the Work

This project aims to:

- Develop an integrated EDR-SOAR solution using LimaCharlie and Tines
- Automate threat response workflows to reduce human intervention
- Enhance real-time monitoring and detection of security incidents
- Reduce alert fatigue by prioritizing critical security events
- Ensure scalability and efficiency in cybersecurity operations

1.6 Organization of the Project

This report is structured as follows:

Chapter 1: Project Description and Outline – Introduces the project, motivation, and objectives.
Chapter 2: Related Work Investigation – Reviews existing security approaches and their limitations.
Chapter 3: Requirement Artifacts – Details hardware, software, and project-specific requirements.
Chapter 4: Design Methodology and Novelty – Discusses architectural design, functional modules,

and UI design.
Chapter 5: Technical Implementation and Analysis – Covers coding, testing, and performance evaluation.
Chapter 6: Project Outcome and Applicability – Describes key implementations, real-world applications, and outcomes.
Chapter 7: Conclusion and Recommendation – Summarizes findings, limitations, and future improvements.

1.7 Summary

This chapter introduced the EDR-SOAR integration project, its importance, and how it addresses modern cybersecurity challenges. The motivation, problem statement, objectives, and structure of the project were outlined. The next chapter will delve into existing security methodologies and how our project improves upon them.

CHAPTER-2: RELATED WORK INVESTIGATION

2.1 Introduction

This chapter presents a comprehensive review of existing research and methodologies related to EDR-SOAR integration using LimaCharlie and Tines. The focus is on understanding current cybersecurity solutions, their strengths, weaknesses, and areas for improvement to enhance our project's effectiveness.

2.2 Core Area of the Project

The core area of this project revolves around:

- Endpoint Detection and Response (EDR) using LimaCharlie for real-time threat monitoring.
- Security Orchestration, Automation, and Response (SOAR) using Tines to automate security workflows.
- Integrating EDR and SOAR for automated threat intelligence, detection, and incident response.

This project aims to improve threat detection, response automation, and incident management efficiency while minimizing human intervention in cybersecurity operations.

2.3 Existing Approaches/Methods

2.3.1 Approach/Method - 1: Traditional Security Information and Event Management (SIEM)

Overview:

- SIEM platforms like **Splunk, IBM QRadar, and ELK Stack** collect logs from multiple sources to identify security incidents.

Advantages:

- Centralized logging and correlation of security events.
- Offers advanced search and analytics for incident investigation.

Disadvantages:

- Lacks real-time threat response, requiring manual intervention.
- Generates a high number of alerts, leading to alert fatigue.

2.3.2 Approach/Method - 2: Standalone Endpoint Detection and Response (EDR)

Overview:

- Tools like CrowdStrike Falcon, Microsoft Defender ATP, and LimaCharlie monitor endpoints for malicious activities.

Advantages:

- Provides real-time threat detection at the endpoint level.
- Detects fileless malware, ransomware, and privilege escalation attempts.

Disadvantages:

- Focuses only on endpoints, lacking network-wide threat correlation.
- Requires manual response handling, increasing security team workload.

2.3.3 Approach/Method - 3: Security Orchestration, Automation, and Response (SOAR) Without EDR

Overview:

- SOAR tools like Tines, Cortex XSOAR, and Splunk Phantom automate security operations but rely on external data sources.

Advantages:

- Reduces manual effort through automated playbooks.
- Improves incident response time by integrating multiple security tools.

Disadvantages:

- Lacks direct endpoint monitoring and depends on other security solutions.
- Requires extensive integration and fine-tuning to be fully effective.

2.4 Pros and Cons of the Stated Approaches/Methods

Approach	Pros	Cons
Traditional EDR	Deep visibility	Requires manual response
SIEM-based	Centralized logging	High false positives
SOAR-only	Automated response	Limited endpoint insights

Table 1:Pros and Cons of the stated approaches

2.5 Issues/Observations from Investigation

From the investigation, the following **key issues and observations** were identified:

1. Need for Real-Time Detection & Automated Response

- SIEM lacks real-time response, while EDR lacks automation. Combining EDR with SOAR bridges this gap.

2. **High Volume of False Positives in Security Alerts**

- Standalone EDR and SIEM solutions generate excessive alerts, leading to alert fatigue for security teams.
- Using SOAR with LimaCharlie can automate triaging to filter out false positives.

3. **Scalability & Flexibility Concerns**

- Traditional SIEM and EDR tools are resource-intensive and require high maintenance.
- LimaCharlie's cloud-native architecture and Tines' no-code automation offer a more scalable solution.

4. **Integration Complexity**

- Many SOAR solutions require custom scripting for playbooks, making implementation complex.
- Tines' UI-based automation simplifies integration with LimaCharlie, reducing configuration effort.

CHAPTER 3: REQUIREMENT ARTIFACTS

3.1 Introduction

This chapter defines the requirements necessary for the implementation of **EDR-SOAR integration using LimaCharlie and Tines**. It covers hardware and software prerequisites, project-specific requirements, data needs, functional aspects, performance and security expectations, and user interface considerations.

3.2 Hardware and Software Requirements

Hardware Requirements

- **Cloud-based Infrastructure**: Required for deploying LimaCharlie and Tines.
- **Endpoints**: Test machines (Windows/Linux) to evaluate detection rules.
- **Storage**: Sufficient storage for logs, alerts, and reports.

Software Requirements

- **LimaCharlie**: EDR platform for threat detection and response.
- **Tines**: SOAR platform for automating security workflows.
- **Python**: For scripting automation tasks.
- **Virtual Machines**: Windows and Linux VMs for testing detection rules.

3.3 Specific Project Requirements

3.3.1 Data Requirement

- Real-time security logs collected from endpoints.
- Detection alerts generated by LimaCharlie rules.
- Incident response actions logged within Tines.
- Encrypted traffic and file activity logs to monitor mass encryption attempts.

3.3.2 Functional Requirement

- **Threat Detection**:
  - Monitor endpoint activities in real time.
  - Detect USB insertions, execution policy bypass, ransomware activity, persistence mechanisms, and mass encryption attempts.
- **Incident Response Automation**:
  - Automatically trigger actions in Tines based on LimaCharlie alerts.
  - Quarantine endpoints and block malicious activity.
- **Alerting and Reporting**:
  - Generate security reports.
  - Notify administrators via email or SIEM integration.

3.3.3 Performance and Security Requirement

- **Low-latency threat detection,** alerts should trigger in real time.
- **Minimal false positives,** rules should be well-optimized.
- **Scalability**: Must support large-scale deployments.
- **Data security**: Encrypted storage for logs and alerts.
- **Access Control**: Restrict access based on user roles.

3.3.4 Look and Feel Requirements

- **LimaCharlie Dashboard**: Clear visualization of alerts and detections.
- **Tines UI**: Intuitive workflow builder for automation.
- **Reports**: Well-structured and easy to interpret, including timestamps, affected endpoints, and remediation actions taken.

3.3.5 Integration and Deployment Requirements

- Seamless integration between LimaCharlie and Tines.
- API-based communication for log and event sharing.
- Compatibility with existing SIEM solutions.
- Cloud and on-premises deployment options.

3.4 Summary

This chapter outlined the essential requirements for implementing the EDR-SOAR integration project. It detailed the hardware, software, data, functional, performance, security, UI, and integration requirements. The next chapter will discuss system design and architecture.

CHAPTER 4: DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and Goal

The EDR-SOAR integration using LimaCharlie and Tines follows a structured methodology that combines real-time endpoint threat detection with automated incident response workflows.

Goals of the Project:

- Develop custom LimaCharlie detection rules for key security events.
- Automate security response actions using Tines.
- Enhance threat intelligence capabilities through log analysis.
- Ensure seamless integration between EDR and SOAR for rapid incident resolution.

4.2 Functional Modules Design and Analysis

Key Functional Modules:

1. **Threat Detection Module**
   - USB Device Monitoring
   - Execution Policy Bypass Detection
   - Ransomware Detection (VSS Admin, WMIC, Cipher)
   - Persistence Mechanism Detection
   - Entertainment Site Blocking
   - Mass File Encryption Monitoring
   - Detection and Termination of Malicious Tool
   - Unauthorised Logon Detection
   - Sensitive System Process Access Detection

2. **Incident Response Automation Module**

   ○ Integrates LimaCharlie with Tines for automated security workflows.

   ○ Triggers quarantine actions and logs alerts.

3. **Alerting and Reporting Module**

   ○ Generates alerts for security teams.

   ○ Logs security incidents for forensic analysis.

4.3 Software Architectural Designs

The architecture consists of:

- **Endpoint Agents (LimaCharlie Sensors)**: Installed on monitored systems.
- **Cloud-based EDR (LimaCharlie)**: Receives and processes endpoint logs.
- **SOAR Platform (Tines)**: Automates response workflows based on LimaCharlie alerts.
- **Database/Storage**: Stores logs, alerts, and incident data.
- **Visualization Layer**: Dashboards and reports for monitoring.

System Flow:

1. Endpoint agents collect security data.
2. LimaCharlie processes logs and applies detection rules.
3. Tines automates responses to security incidents.
4. Security teams receive alerts and take action.

4.4 Subsystem Services

- **Data Collection Service**: Gathers logs from endpoints.
- **Threat Detection Service**: Applies custom LimaCharlie rules.
- **Incident Response Service**: Automates workflows in Tines.

- **Reporting and Analysis Service**: Generates security reports.

4.5 User Interface Designs

- **LimaCharlie Dashboard**: Displays real-time security events.
- **Tines Workflow Builder**: Graphical interface for automation workflows.
- **Alerting System**: Email and dashboard notifications for incidents.

4.6 Summary

This chapter covered the methodology, system architecture, functional modules, subsystem services, and UI designs of the EDR-SOAR integration project. The next chapter will focus on implementation and testing.

CHAPTER 5: TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Outline

This chapter provides a detailed breakdown of the implementation process for the EDR-SOAR integration using LimaCharlie and Tines. It covers the coding aspects, system workflows, prototype submission, validation, and performance analysis.

5.2 Technical Coding and Code Solutions

Implementation of LimaCharlie Detection Rules

I. USB Rule

- **Trusted USB Inserted**

- Objective: Detect when a pre-approved USB device is inserted and log the event.

Implementation Approach:

- Maintain a whitelist of trusted USB devices based on removable device name .

- Generate logs when an authorized device is connected.
- Take no restrictive action but maintain an audit trail for security analysis.

LimaCharlie Detection Rule:

```
events:
  - VOLUME_MOUNT
op: and
rules:
  - op: is windows
  - op: contains
    path: event/DEVICE_TYPE
    value: REMOVABLE
  - op: is
    path: event/VOLUME_NAME
    value: SAFEDRIVE
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  metadata:
    author: TerryVIT
    description: Trusted USB Connected !
    level: safe
    tags:
      - usb_protected
      - autorun_block
  name: VIT Trusted USB Inserted
```

Code Snippet

- **Unrecognized USB Inserted**

Objective: Detect when an unrecognized USB device is inserted and trigger an alert for security monitoring.

Implementation Approach:

- Generate logs and alerts when a non-whitelisted (unrecognized) USB is connected.

Trigger **automated response actions**, such as:

- **Notifying SOC analysts** via email, SIEM, or SOAR platform.
- **Logging the event** for forensic analysis and auditing.

LimaCharlie Detection Rule:

```
events:
  - VOLUME_MOUNT
op: and
rules:
  - op: is windows
  - op: contains
    path: event/DEVICE_TYPE
    value: REMOVABLE
```

Code snippet

LimaCharlie Respond Rule:

```
- action: report
  metadata:
    author: TerryVIT
    description: Unrecognized USB Connected !
    level: safe
    tags:
      - usb_protected
      - autorun_block
  name: Unrecognized USB is Plugged
- action: isolate network
  metadata:
    description: Isolates the network when autorun.inf is detected.
    level: critical
  name: isolate-Network
```

II.

Code Snippet

ExecutionPolicy Bypass

- **Bypass using PowerShell**

Objective: Detect and alert when PowerShell execution policies are bypassed using common attack techniques.

Common Attack Techniques:

- Using "-ExecutionPolicy Bypass" to run scripts without restrictions
- Invoking scripts via encoded commands
- Using Windows Management Instrumentation (WMI) or rundll32.exe to execute PowerShell
- Executing obfuscated PowerShell commands

LimaCharlie Detection Rule:

```
event: NEW_PROCESS
op: and
rules:
  - op: is
    path: event/FILE_PATH
    value: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
  - op: starts with
    path: event/COMMAND_LINE
    value: >-
      "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
      -ExecutionPolicy Bypass -File C:\Scripts\
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  metadata:
    author: TerryVIT
    description: Bypass using Powershell Detected.
    level: high
    tags:
      - attack.elevation_of_privilege
  name: Bypass using Powershell Detected.
```

Code Snippet

- **Bypass False Positive (Safe Script)**

Objective: Allow legitimate PowerShell scripts to run while avoiding false positives.

Implementation Approaches:

- Writing a false positive rule with respective to new scripts.
- Maintain a whitelist of trusted scripts based on file hashes, script paths, or specific command-line arguments

- Use event filtering to differentiate between benign and malicious PowerShell execution

LimaCharlie False Positive Detection Rule:

```
op: and
rules:
  - op: is
    path: cat
    value: Bypass using Powershell Detected.
  - op: is
    path: detect/event/FILE_PATH
    value: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
  - op: is
    path: detect/event/COMMAND_LINE
    value: >-
      "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
      -ExecutionPolicy Bypass -File C:\Scripts\safe_script.ps1
  - op: is
    path: detect/event/HASH
    value:
9785001b0dcf755eddb8af294a373c0b87b2498660f724e76c4d53f9c217c7a3
  - op: is
    path: routing/hostname
    value: desktop-plitmnu
```

Code Snippet

III. Ransomware Detection (Powershell)

- **VSS Admin Deletion**

Objective: Detect and prevent attackers from deleting Volume Shadow Copies (VSS) using the *vssadmin delete shadows* command.

Why it's dangerous?

- Shadow copies allow users to restore files to previous versions.
- Ransomware deletes these copies to prevent recovery without paying ransom.

LimaCharlie Detection Rule:

```
events:
  - NEW_PROCESS
  - EXISTING_PROCESS
op: and
rules:
  - op: is windows
  - op: or
    rules:
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: vssadmin delete shadows /all /quiet
      - case sensitive: false
        op: is
        path: event/HASH
        value:
8c1fabcc2196e4d096b7d155837c5f699ad7f55edbf84571e4f8e03500b7a8b0
      - case sensitive: false
        op: is
        path: event/HASH
        value:
bf4fa71c1495f95adbcf3f7c7d41837e2661622c2ee3b24cd9647676047578da
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: wmic shadowcopy delete
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: cipher /w
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: rundll32.exe cryptbase.dll
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  metadata:
    author: Terry VIT
    description: Ransomware activity detected.
    level: high
    tags:
      - attack.malicious
  name: ransomware-activity-detected
- action: isolate network
  metadata:
    description: Isolates the network when ransomware activity is
detected.
    level: critical
  name: isolate-ransomware-activity
```

Code Snippet

- **WMIC shadow copy deletion**

Objective: Detect attackers using wmic to delete shadow copies.

Why it's dangerous?

- WMIC (Windows Management Instrumentation Command-line) is another tool used to manage shadow copies.
- Attackers use it as an alternative if vssadmin is blocked.

LimaCharlie Detection Rule:

```
events:
  - NEW_PROCESS
  - EXISTING_PROCESS
op: and
rules:
  - op: is windows
  - op: or
    rules:
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: vssadmin delete shadows /all /quiet
      - case sensitive: false
        op: is
        path: event/HASH
        value:
8c1fabcc2196e4d096b7d155837c5f699ad7f55edbf84571e4f8e03500b7a8b0
      - case sensitive: false
        op: is
        path: event/HASH
        value:
bf4fa71c1495f95adbcf3f7c7d41837e2661622c2ee3b24cd9647676047578da
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: wmic shadowcopy delete
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: cipher /w
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE

        value: rundll32.exe cryptbase.dll
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  metadata:
    author: Terry VIT
    description: Ransomware activity detected.
    level: high
    tags:
      - attack.malicious
  name: ransomware-activity-detected
- action: isolate network
  metadata:
    description: Isolates the network when ransomware activity is
detected.
    level: critical
  name: isolate-ransomware-activity
```

Code Snippet

- **Cipher Activity**

Objective: Detect attackers using the Cipher tool to wipe free space, making file recovery impossible.

Why it's dangerous?

- *cipher.exe* is a built-in Windows tool used to securely delete files.
- Attackers use it to overwrite deleted files, preventing forensic recovery.

LimaCharlie Detection Rule:

```
events:
  - NEW_PROCESS
  - TERMINATE_PROCESS
op: or
rules:
  - op: is
    path: event/FILE_PATH
    value: C:\Windows\system32\cipher.exe
  - op: is
    path: event/COMMAND_LINE
    value: '"C:\Windows\system32\cipher.exe" /w:C:\'
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  name: Destroying deleted data using Cipher
- action: task
  command: terminate
  params:
    pid: '{{ .event.PARENT.PROCESS_ID }}'
```

Code Snippet

IV. Entertainment Site Blocking

- **Block list of 9 websites using LimaCharlie DNS rules.**

Objective

- Restrict access to entertainment websites (e.g., YouTube, Netflix, social media, etc.).
- Enhance security by blocking potential phishing or malware sites disguised as entertainment platforms.
- Improve workforce productivity by reducing distractions.

LimaCharlie DNS Filtering Configuration

To block a list of 9 entertainment websites, we define custom DNS rules in LimaCharlie:

```
events:
  - DNS_REQUEST
op: or
rules:
  - op: is
    path: event/DOMAIN_NAME
    value: www.youtube.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.instagram.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.facebook.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.netflix.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.tiktok.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.spotify.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.hotstar.com
  - op: is
    path: event/DOMAIN_NAME
    value: www.amazon.in
  - op: is
    path: event/DOMAIN_NAME
    value: www.flipkart.com
```

Code Snippet

Respond Rule for Entertainment Site Block:

```
- action: report
  metadata:
    author: Terry VIT
    description: Alert for DNS request to block youtube.com
    level: high
    tags:
      - entertainment.sites
  name: Entertainment-Site-Visited
- action: task
  command: block
- action: task
  command: drop_request
```

Code Snippet

V. Persistence Mechanisms

- **Detecting HKCU Registry Backdoors**

Objective: Attackers often use the HKEY_CURRENT_USER (HKCU) registry hive to establish persistence because it does not require administrative privileges. Malicious scripts or executables can add registry keys that trigger malware execution upon user login.

Threat:

Attackers often create registry-based backdoors under HKEY_CURRENT_USER (HKCU)\Software\Microsoft\Windows\CurrentVersion\Run to execute malicious payloads at startup.

Limacharlie Detection Rule:

```
event: NEW_PROCESS
op: and
rules:
  - op: is
    path: event/FILE_PATH
    value: C:\WINDOWS\system32\reg.exe
  - op: starts with
    path: event/COMMAND_LINE
    value: >-
      "C:\WINDOWS\system32\reg.exe" add
      HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v Backdoor /t
REG_SZ
```

Code Snippet

Limacharlie Respond Rule:

```
- action: report
  metadata:
    author: TerryVIT
    description: HKCU Activity Detected
    level: high
    tags:
      - admin-privilege
  name: HKCU Activity Detected
```

Code Snippet

- **Monitoring Scheduled Task Creation (schtasks.exe)**

Objective: Detection of the use of Scheduled Tasks (schtasks.exe) used to execute malicious scripts or programs at specific intervals, ensuring continued control over an infected system.

Threat:

Attackers abuse schtasks.exe to create scheduled tasks that run malicious scripts at predefined intervals, ensuring long-term persistence.

Limacharlie Detection Rule:

```
event: NEW_PROCESS
op: and
rules:
  - case sensitive: false
    op: is
    path: event/FILE_PATH
    value: C:\WINDOWS\system32\schtasks.exe
  - op: starts with
    path: event/COMMAND_LINE
    value: >-
      "C:\WINDOWS\system32\schtasks.exe" /create /tn BackdoorTask /tr
      C:\malware.exe /sc onlogon /f
```

Code Snippet

Limacharlie Respond Rule:

```
- action: report
  metadata:
    author: TerryVIT
    description: schtasks Usage Detected
    level: high
    tags:
      - admin-privilege
  name: schtasks Usage Detected
```

Code Snippet

- **WMIC-Based Backdoors**

Objective: Detecting the usage of Windows Management Instrumentation Command-line (WMIC) tool which is abused by attackers to execute remote commands or scripts stealthily. This technique is often used in fileless malware attacks.

Threat:

Windows Management Instrumentation Command-line (WMIC) can be used to execute hidden persistence mechanisms. Attackers create scheduled events or execute scripts remotely.

Limacharlie Detection Rule:

```
event: NEW_PROCESS
op: and
rules:
  - op: is
    path: event/FILE_PATH
    value: C:\WINDOWS\System32\Wbem\WMIC.exe
  - op: is
    path: event/COMMAND_LINE
    value: >-
      "C:\WINDOWS\System32\Wbem\WMIC.exe"
/namespace:\\root\subscription Path
      __EventFilter Create Name=PersistenceTest
EventNamespace=root\\CimV2
      QueryLanguage=WQL "Query=SELECT * FROM __InstanceCreationEvent
WITHIN 10
      WHERE TargetInstance ISA 'Win32_Process'"
```

Code Snippet

Limacharlie Respond Rule:

```
- action: report
  metadata:
    author: Terry VIT
    description: WMIC Event filtering activity detected.
    level: high
    tags:
      - attack.malicious
  name: WMIC Event Filter enabling detected
- action: isolate network
  metadata:
    description: Isolates the network when ransomware activity is
detected.
    level: critical
  name: isolate-ransomware-activity
```

Code Snippet

VI. File Extension & Mass Encryption Detection

- **Detect rapid file modification indicative of ransomware.**

Objective: Detection of ransomware attacks which often involve mass encryption of files, where a malicious process rapidly modifies multiple files by changing the extensions or encrypting the contents.

Why it's dangerous?

- **Data Loss:** Ransomware encrypts files, making them inaccessible without a decryption key.
- **Business Disruption:** Encrypted critical files can halt operations, causing downtime.
- **Ransom Demands:** Attackers demand payment for decryption, leading to financial loss.
- **Spreading Threats:** Some ransomware variants spread laterally, affecting entire networks.

LimaCharlie Detection Rule:

```
events:
  - FILE_CREATE
  - NEW_DOCUMENT
  - NEW_PROCESS
  - SERVICE_CHANGE
  - EXISTING_PROCESS
  - CLOUD_NOTIFICATION
op: and
rules:
  - op: is windows
  - op: or
    rules:
      - op: contains
        path: event/FILE_PATH
        value: .locky
      - op: contains
        path: event/FILE_PATH
        value: .crypted
      - op: contains
        path: event/FILE_PATH
        value: .ransom
      - op: contains
        path: event/FILE_PATH
        value: .crypted000007
      - op: and
        rules:
          - op: contains
            path: event/FILE_PATH
            value: C:\Windows\system32\DllHost.exe
          - op: or
            rules:
              - op: contains
                path: event/COMMAND_LINE
                value: '{AB8902B4-09CA-4BB6-B78D-A8F59079A8D5}'
              - op: contains
                path: event/PARENT/FILE_PATH
                value: C:\Windows\system32\svchost.exe
              - op: contains
                path: event/PARENT/FILE_PATH
                value: C:\Windows\system32\powershell.exe
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: isolate network
  metadata:
    author: TerryVIT
    description: Detects and isolates ransomware behavior based on file
renaming
    falsepositives:
      - Legitimate user actions (should be verified)
    level: critical
    tags:
      - attack.impact
  name: Ransomware - File Renaming Detected
- action: report
  metadata:
    author: TerryVIT
    description: Report potential ransomware activity
    level: high
    tags:
      - ransomware
      - file_rename
  name: Renaming of files Detected
```

Code Snippet

VII. Detection and Termination of Malicious Tool

Objective: To identify and neutralise Mimikatz through real-time process and command line analysis

Threat:

➢ Threat Actors use Mimikatz to access sensitive data such as credentials in plain text, hashed passwords, PINs and Kerberos Tickets

➢ Mimikatz is often launched via disguised scripts, renamed binaries, or PowerShell to bypass security tools

➢ It endangers domain environments by exposing admin-level credentials.

Risk:

➢ Once credentials are stolen, attackers can leverage it to move laterally and access multiple systems without detection.

➢ The breach can extend to sensitive components, eventually lead to complete system takeover.

➢ Since it operates in-memory, it is very stealthy and offers limited forensic traces. This makes forensic investigation very challenging.

➢ Escalated privileges allow attackers to disable security controls and remove any digital traces of the attack.

Implementation:

➢ NEW_PROCESS and EXISTING_PROCESS events are tracked to cover both real-time processes and those that existed before LimaCharlie sensor loaded.

➢ Attackers often execute Mimikatz through powershell or other command-line interfaces.

➢ The LimaCharlie detection rule relies on matching file name and string patterns to detect Mimikatz usage.

➢ Logical operators(OR/AND) are used to combine various indicators which widens the detection scope.

➢ Upon a match, the reporting action logs an alert titled “Mimikatz Detected and Killed.”

➢ The “deny_tree” command kills the entire process tree.

➢ A follow up task uses the “file_del” command to initiate the deletion of Mimikatz Executable.

LimaCharlie Detection Rule:

```
events:
  - NEW_PROCESS
  - EXISTING_PROCESS
op: and
rules:
  - op: is windows
  - op: or
    rules:
      - case sensitive: false
        op: ends with
        path: event/FILE_PATH
        value: mimikatz.exe
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: mimikatz
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  name: Mimikatz Detected and Killed
- action: task
  command:
    - deny_tree
    - <<routing/this>>
- action: task
  command: file_del "{{ .event.FILE_PATH }}"
```

Code Snippet

VIII. Unauthorised Logon Detection

Objective: To detect interactive or remote login attempts during off-hours and trigger immediate shutdown.

Threat:

- ➢ Hackers prefer to log in when the activity is less, as it helps them evade human oversight.
- ➢ Most attackers sneak in through remote or network logins, as it is hard to detect them.
- ➢ These break-ins can easily slip right past the security team without live alerts.

Risk:

- ➢ Attackers can effortlessly deploy malicious payloads once they are in.
- ➢ They need only one exploited session to escalate privilege and quickly take over the entire system.
- ➢ Logins during off-hours often fly undetected.
- ➢ It becomes a lot easier for attackers to hide their tracks in absence of active monitoring.

Implementation:

- ➢ Ensure Lima Charlie is allowed to ingest Windows Event Logs(WEL) through its artifact pipeline.
- ➢ The detect block monitors for successful logins.
- ➢ It only checks for interactive and remote logons as they suggest guaranteed real user activity.

- The rule is set to catch logins happening outside of permitted hours.
- If a match is found, then the response block logs it as an “unauth login” alert.
- It also executes the command “shutdown --is-confirmed” to immediately power off the system.
- This prevents the attacker from further exploitation.

LimaCharlie Detection Rule:

```
events:
  - WEL
op: and
rules:
  - op: is
    path: event/EVENT/System/EventID
    value: '4624'
  - op: is
    path: event/EVENT/System/Channel
    value: Security
  - op: matches
    path: event/EVENT/System/TimeCreated/SystemTime
    re: T(08|09|10):[0-5][0-9]:[0-5][0-9]
  - op: matches
    path: event/EVENT/EventData/LogonType
    re: ^(2|10|11)$
```

Code Snippet

LimaCharlie Respond Rule:

```
- action: report
  name: unauth login
```

Code Snippet

IX. Sensitive System Process Access Detection

Objective: To detect interactive or remote login attempts during off-hours and trigger immediate shutdown.

Threat:

- Malicious actors prefer to target Local Security Authority Subsystem Service (Lsass.exe) to steal access tokens.
- Core processes like Client/Server Runtime Subsystem are often sought after to elevate control or hide actions.
- Malicious code mimics patterns of trusted processes to slip past security tools.
- In the post-exploitation step, these core processes are very valuable for sensitive credential dumping.

Implementation:

- The detection block listens for “SENSITIVE_PROCESS_ACCESS” events from the endpoint sensor.
- The rule flags access to any of the core processes specified in the detect block.
- Safe processes like MRT.exe are ignored to avoid triggering false alerts
- The report block then logs the executable that tried to access the sensitive process.

LimaCharlie Detection Rule

```
event: SENSITIVE_PROCESS_ACCESS
op: and
rules:
  - case sensitive: false
    op: ends with
    path: event/*/event/TARGET/FILE_PATH
    value: lsass.exe
  - case sensitive: false
    not: true
    op: ends with
    path: event/*/event/SOURCE/FILE_PATH
    value: system32\MRT.exe
  - case sensitive: false
    not: true
    op: ends with
    path: event/*/event/SOURCE/FILE_PATH
    value: system32\csrss.exe
  - case sensitive: false
    not: true
    op: ends with
    path: event/*/event/SOURCE/FILE_PATH
    value: windows\sysmon.exe
```

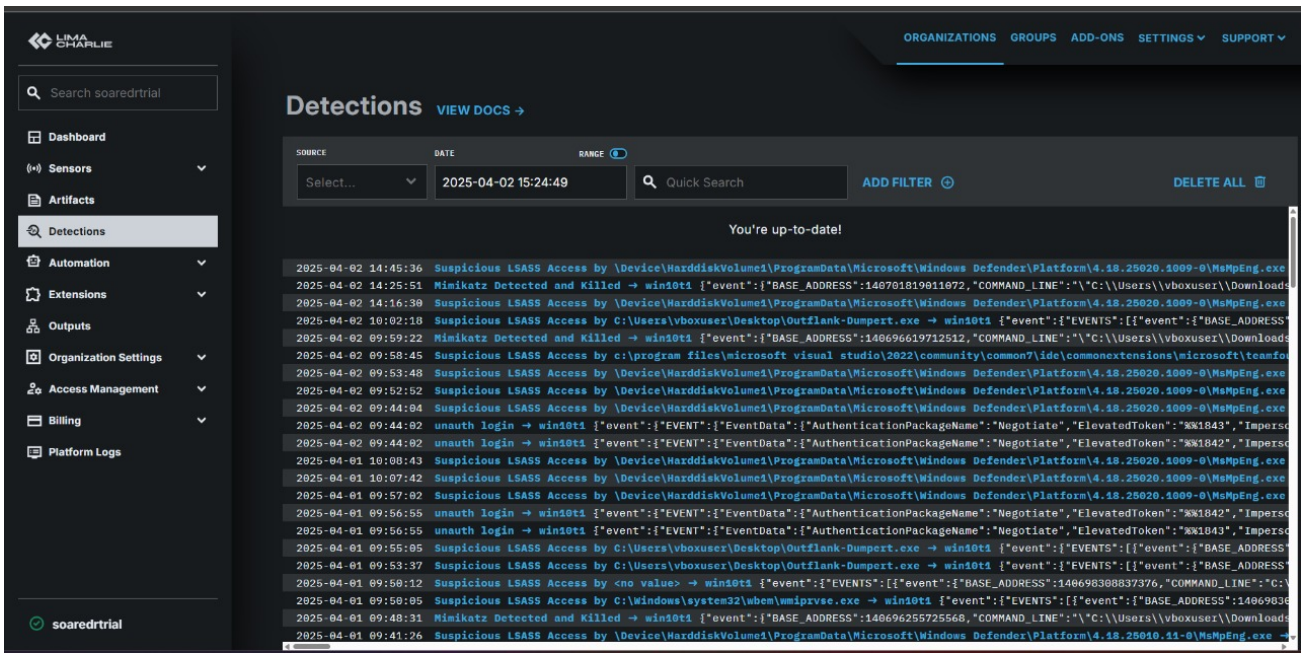
Code Snippet

LimaCharlie Respond Rule:

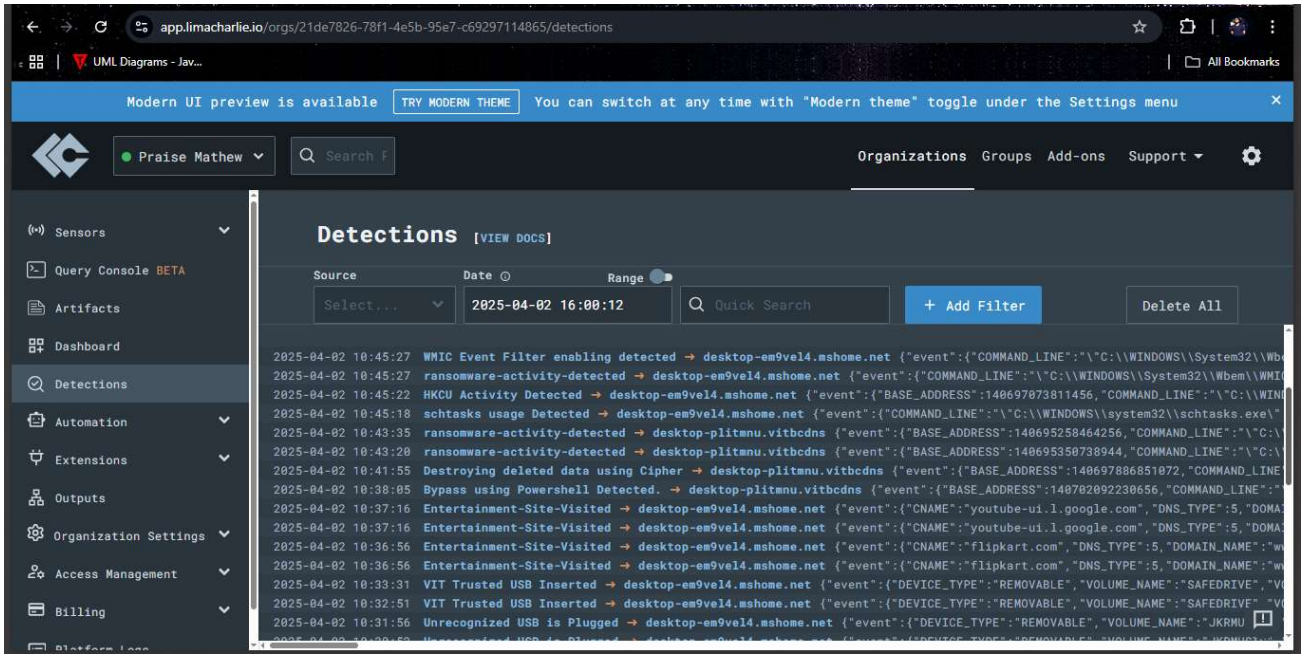
```
Response Block:
- action: report
  name: >-
    Suspicious LSASS Access by {{ index (index .event.EVENTS 1) "event"
"SOURCE"
    "FILE_PATH" }
```

Code Snippet

LimaCharlie Detection



LimaCharlie Detection Dashboard 1



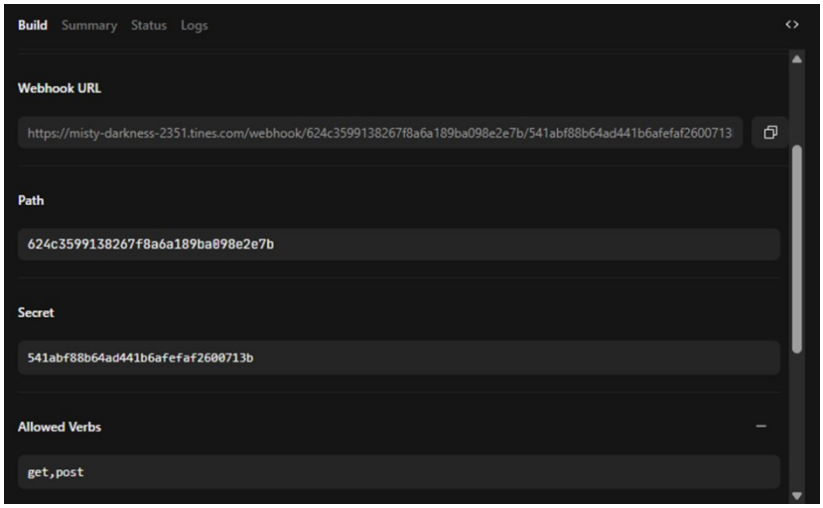
LimaCharlie Detection Dashboard 2

Automation through Tines Playbook

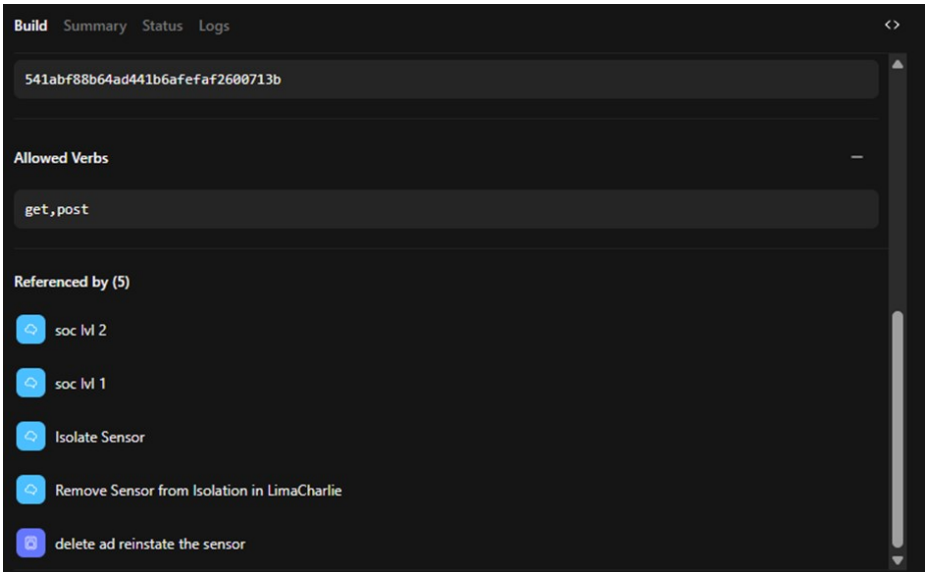
I. Tines Webhook Configuration for receiving alerts

Objective: Alerts which are generated by Lima Charlie are automatically sent to the Tines Webhook for security automation.

Mechanism:
The webhook's URL is configured within Lima Charlie from where alerts are directly forwarded. This marks the beginning of Tines Security Automation Playbook.



Tines Playbook Config



Tines Playbook METHODS Aloowed with refrences

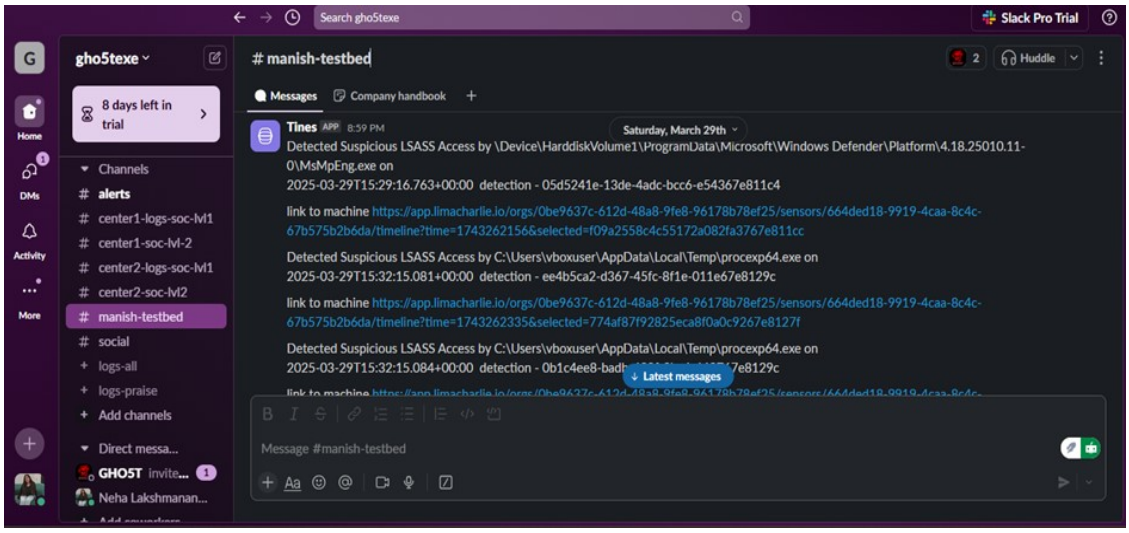
II. Email and Slack Configuration

Objective: Once the alerts are in the webhook it has to be notified. For that we have used Slack (collaboration platform) and Email so that potential threats can be informed immediately.

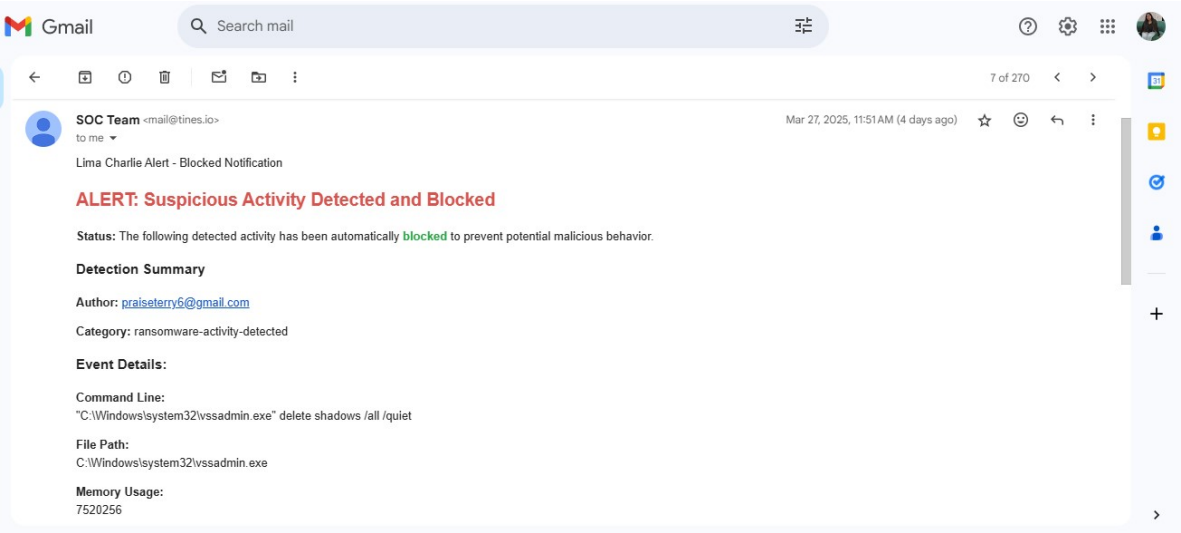
Mechanism:
Slack has to be integrated withTines through sharing the Channel/User ID(from slack). In this project we have deployed 4 channels which are :

- Center 1 - SOC Lvl 1
- Center 1- SOC Lvl 2
- Center 2 - SOC Lvl 1
- Center 2 - SOC Lvl 2

SOC Lvl 1- Will have basic alert details such as the name/category of the alert, its generation time and the detection ID.
SOC Lvl 2 - Will have basic details same as in Lvl 1 additionallyit will have a link to the lima charlie machine.



Playbook Configuration to initiate Practical



Email with threat detection

III. A response automation page

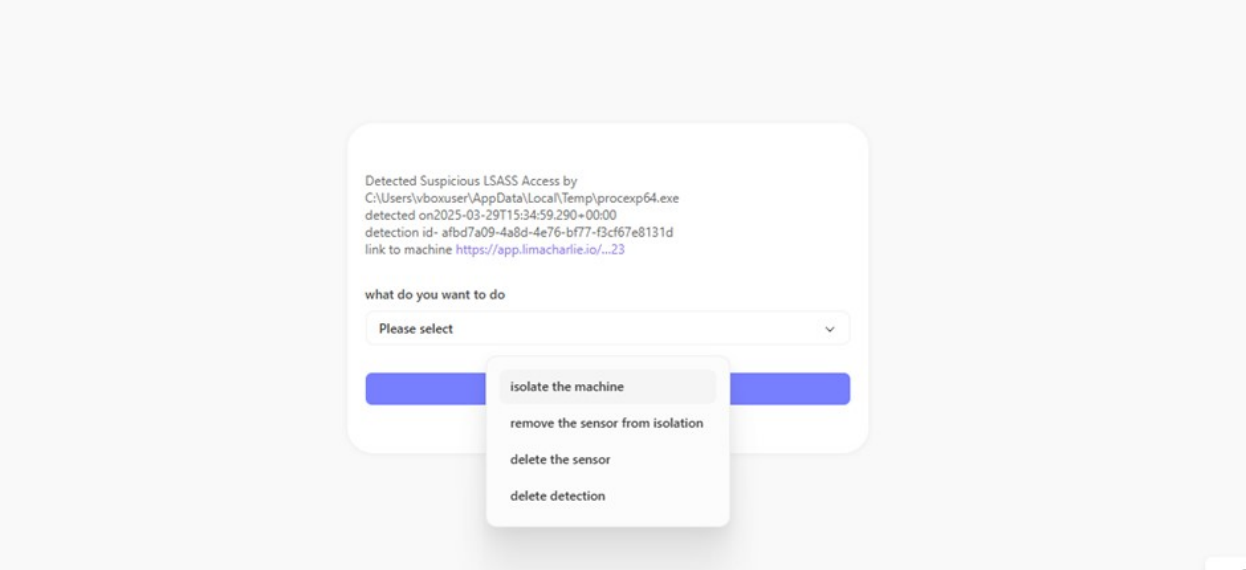
Objective: The objective of this automation page is for implementing quicker actions.

Mechanism:

We have created 4 options according to the project requirement namely:

- Isolate the sensor
- Delete Detection
- Delete Sensor
- Remove Sensor from Isolation

This helps with rapid incident response and centralized management. Any one of the 4 options should be selected and submitted upon which the next step will start its implementation.



Automation Panel

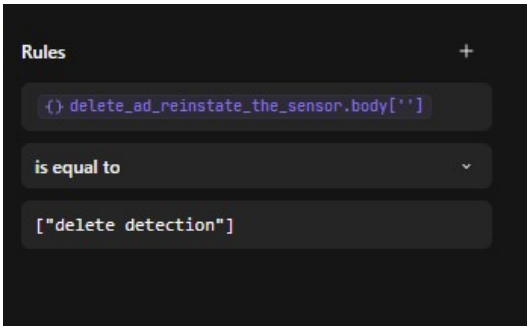
Each of these option will trigger an HTTP request which will be explained in the next section.

IV. Configuring Trigger Events

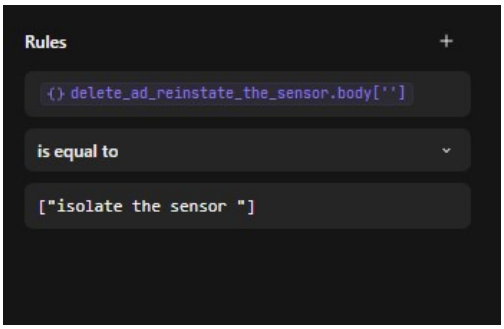
Objective: Upon selection of any option mentioned in the page a corresponding trigger event will be activated that is the selection from the page is marked as the input for the trigger event.

Mechanism:

Pull up a trigger event and setup the rules corresponding to the input from the sheet to get it triggered.



Trigger events[delete detection]



Trigger events[isolate the sensor]

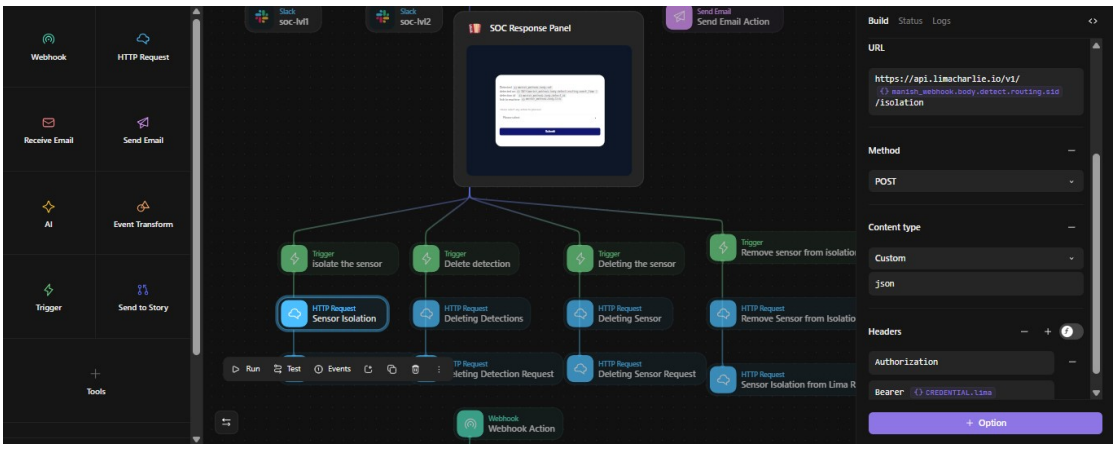
Trigger events[delete Sensor]

V.Executing Action

Objective: Here comes the main thing once a trigger is activated let it be any of them, a HTTP request if formed with the Lima Charlie's API as the URL which directs what Lima Charlie has to do like isolating, deleting the sensor etc.

Mechanism:

Get the API from Lima Charlie put it up in the URL set the method to POST and fill in the Lima Credentials. This will direct Lima Charlie to take respective action.



System Design with Wireframes

Finally a message is sent to slack that the selected action has been performed. Therefore the blocking/Deleting / Isolation of the sensor can be done automatically through the Tines Security Automated Playbook.

5.3 Working Layout of Forms

The system consists of:

- **LimaCharlie Dashboard:** Displays alerts, rules, and logs.
- **Tines Workflow Builder:** Configured automation flows.
- **Incident Reporting Panel:** Logs detected threats.

5.4 Prototype Submission

A working prototype was deployed with:

- LimaCharlie agents installed on test endpoints.
- Predefined detection rules actively monitoring security events.
- Automated Tines workflows responding to incidents.

5.5 Test and Validation

Test Cases and Scenarios

Test Case	Expected Outcome	Result
Insert trusted USB	Alert triggered with specific note	Passed
Insert unrecognized USB	Alert triggered	Passed
Run PowerShell with ExecutionPolicy Bypass	Alert triggered	Passed
Bypass False Positive (safe_script)	No alert	Passed
Ransomware (VSS Admin, WMIC, Cipher)	Alert triggered	Passed
Entertainment Site Block	Alert triggered	Passed
Persistence Mechanisms (HKCU Backdoor Detection, schtasks Usage Detection,WMIC Backdoor)	Alert triggered	Passed
File Extension & Mass Encryption Detection	Alert triggered	Passed

Detection and Termination of Mimikatz	Alert triggered	Passed
Unauthorised Logon Detection	Alert triggered	Passed
Lsass.exe Detection	Alert triggered	Passed

Table 2: Test Cases and Scenarios

5.6 Performance Analysis (Graphs/Charts)

- Threat Detection Accuracy (Graph of rule effectiveness)
- Incident Response Time (Comparison of manual vs. automated response time)
- False Positives vs. Actual Threats (Data visualization of rule effectiveness)

5.7 Summary

This chapter covered the technical implementation, detection rule configurations, UI layouts, testing results, and performance analysis. The next chapter will discuss deployment and real-world impact.

CHAPTER-6: PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

This chapter summarizes the key implementations, significant outcomes, and real-world applicability of the EDR-SOAR Integration using LimaCharlie and Tines. It highlights how the project enhances cybersecurity defenses and provides insights into its practical use cases.

6.2 Key Implementations of the System

The core implementations of this project include:

1. **LimaCharlie-based Threat Detection**

○ Developed custom detection rules for USB monitoring, Execution Policy Bypass, Ransomware behavior, and Persistence Mechanisms.

○ Implemented real-time alerting and logging for suspicious activities.

2. **SOAR Automation using Tines**

○ Configured automated response workflows to handle security incidents efficiently.

○ Integrated incident escalation processes to notify administrators in case of critical threats.

3. **Incident Response and Analysis**

○ Established automated threat analysis using LimaCharlie's telemetry.

○ Implemented blocking mechanisms for entertainment websites to enhance security in enterprise environments.

4. **Performance Monitoring & Optimization**

○ Optimized false positive handling by refining detection rules.

○ Implemented real-time logging and alerting dashboards for security monitoring.

6.3 Significant Project Outcomes

The following are the major achievements of the project:

Improved Threat Detection Accuracy

- Successfully detected ransomware activities, USB threats, and execution bypass attempts.

Automated Incident Response

- Reduced response time through Tines-powered SOAR workflows.

Reduced False Positives

- Implemented exception rules to prevent unnecessary alerts.

Enhanced Endpoint Security

- Blocked malicious persistence techniques such as HKCU backdoors and scheduled task abuses.

Scalability and Flexibility

- LimaCharlie's cloud-based architecture allows easy scaling and deployment across multiple endpoints.

6.4 Project Applicability in Real-World Applications

The project is highly applicable in various cybersecurity domains, including:

1. Enterprise Security

- Automated threat detection for corporate networks.
- Prevention of unauthorized USB usage to protect against insider threats.

2. Financial Sector Protection

- Mitigates ransomware attacks that target financial institutions.
- Protects against unauthorized execution policy changes to prevent privilege escalation.

3. Government & Defense

- Detection of advanced persistent threats (APTs) through LimaCharlie telemetry.
- Incident response automation for high-security environments.

4. Small and Medium Businesses (SMBs)

- Cost-effective cybersecurity solution for organizations with limited security teams.
- Prevents unauthorized access to sensitive data.

6.5 Inference

This project demonstrates the effectiveness of EDR-SOAR integration using LimaCharlie and Tines. It provides a scalable and automated security framework that enhances threat detection and incident response. The real-world applications validate the practicality of the system, making it suitable for enterprise, government, and SMB security.

With its automated workflows, real-time detection, and minimal false positives, this solution contributes to a proactive cybersecurity defense strategy against evolving threats.

CHAPTER-7: CONCLUSIONS AND RECOMMENDATIONS

7.1 Outline

This chapter provides a summary of the project, its limitations, potential improvements, and key inferences. It highlights the effectiveness of integrating LimaCharlie (EDR) with Tines (SOAR) and suggests future enhancements to improve the system's efficiency and adaptability.

7.2 Limitations/Constraints of the System

While the EDR-SOAR Integration using LimaCharlie and Tines has significantly improved threat detection and automated response, certain limitations exist:

1. Dependence on Cloud Services

- LimaCharlie operates on a cloud-based infrastructure, which may introduce latency or availability risks in regions with poor connectivity.

2. False Positives in Threat Detection

- Although optimized, some detection rules may still trigger false positives, requiring manual intervention for fine-tuning.

3. Limited Customization in Tines SOAR Workflows

- While Tines provides flexibility, some advanced automation features require custom scripting, adding complexity to implementation.

4. Resource Consumption

- The continuous monitoring and logging processes may consume high system resources, potentially impacting performance on low-end devices.

7.3 Future Enhancements

To further strengthen the system, the following enhancements are recommended:

1. AI-Driven Threat Intelligence

- Integrating machine learning algorithms to automatically refine detection rules and minimize false positives.

2. Offline Detection and Response

- Implementing local threat intelligence storage to enable offline security monitoring in case of network disruptions.

3. Expanded SOAR Integration

- Adding more third-party security tools (e.g., ELK Stack, Cortex XSOAR) to enhance threat intelligence sharing and response automation.

4. Advanced Behavioral Analysis

- Incorporating user behavior analytics (UBA) to detect anomalous activities beyond predefined rules.

5. Enhanced Dashboard and Reporting

- Developing a custom security dashboard for better incident visualization and reporting to security teams.

7.4 Inference

The EDR-SOAR Integration using LimaCharlie and Tines has successfully demonstrated a scalable and automated approach to cybersecurity threat detection and response. This system:

- Effectively detects and mitigates threats such as ransomware, unauthorized USB insertions, and execution bypass attempts.
- Automates security operations using Tines SOAR workflows, reducing manual intervention.
- Provides a structured incident response mechanism, improving organizational cybersecurity posture.

Despite its limitations, the system proves to be a valuable cybersecurity solution for enterprises and SMBs. Future enhancements, such as AI-driven threat detection and expanded SOAR integrations, will further optimize its performance and adaptability.

With continuous improvements, this solution has the potential to significantly enhance modern cybersecurity defense strategies against evolving threats.

REFERENCES

1. Tines, "Deploying and using the Tines Tunnel," *Tines Explained*, 2024. https://explained.tines.com/en/articles/6883952-deploying-and-using-the-tines-tunnel. [Accessed: March, 2025].

2. LimaCharlie, "LimaCharlie Core Concepts," *LimaCharlie Documentation*, 2024. https://docs.limacharlie.io/docs/limacharlie-core-concepts. [Accessed: March, 2025].

3. Microsoft, "What is a Security Operations Center (SOC)?", *Microsoft Security*, 2024. https://www.microsoft.com/en-us/security/business/security-101/what-is-a-security-operations-center-soc. [Accessed: March, 2025].

4. CrowdStrike, "Endpoint Detection and Response (EDR)," *CrowdStrike Cybersecurity 101*, 2024. https://www.crowdstrike.com/en-us/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/. [Accessed: March, 2025].

5. Fortinet, "Cyber Threat Intelligence," *Fortinet CyberGlossary*, 2024. https://www.fortinet.com/resources/cyberglossary/cyber-threat-intelligence. [Accessed:March, 2025].

6. Palo Alto Networks, "What is SOAR?", *Palo Alto Networks Cyberpedia*, 2024. http://paloaltonetworks.com/cyberpedia/what-is-soar. [Accessed: March, 2025].

7. EC-Council, "Cybersecurity Security Operations Center (SOC) Analyst," *GitHub Repository*, 2024. https://github.com/ec-council-learning/Cybersecurity-Security-Operations-Center-SOC-Analyst/tree/main. [Accessed: March, 2025].