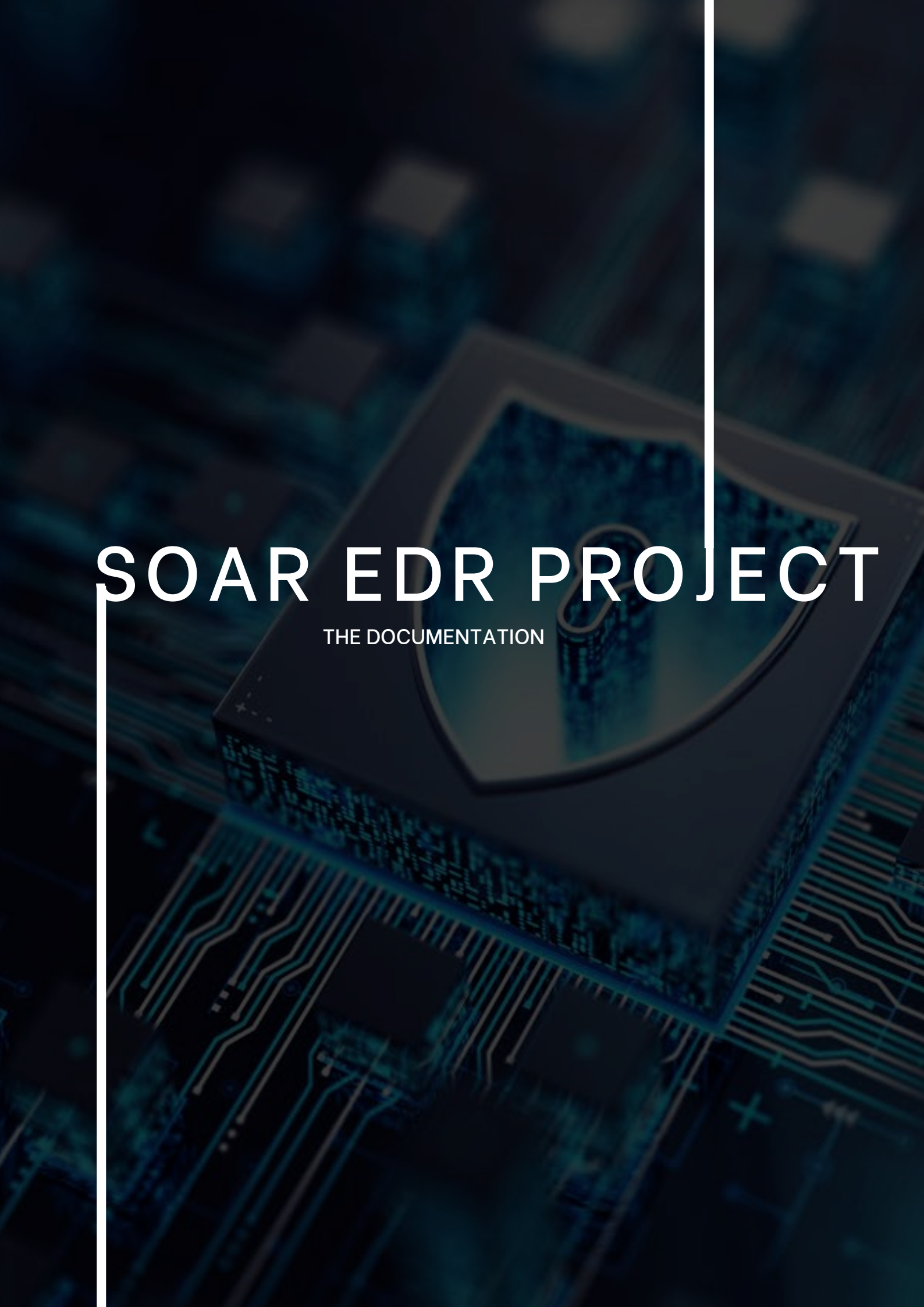
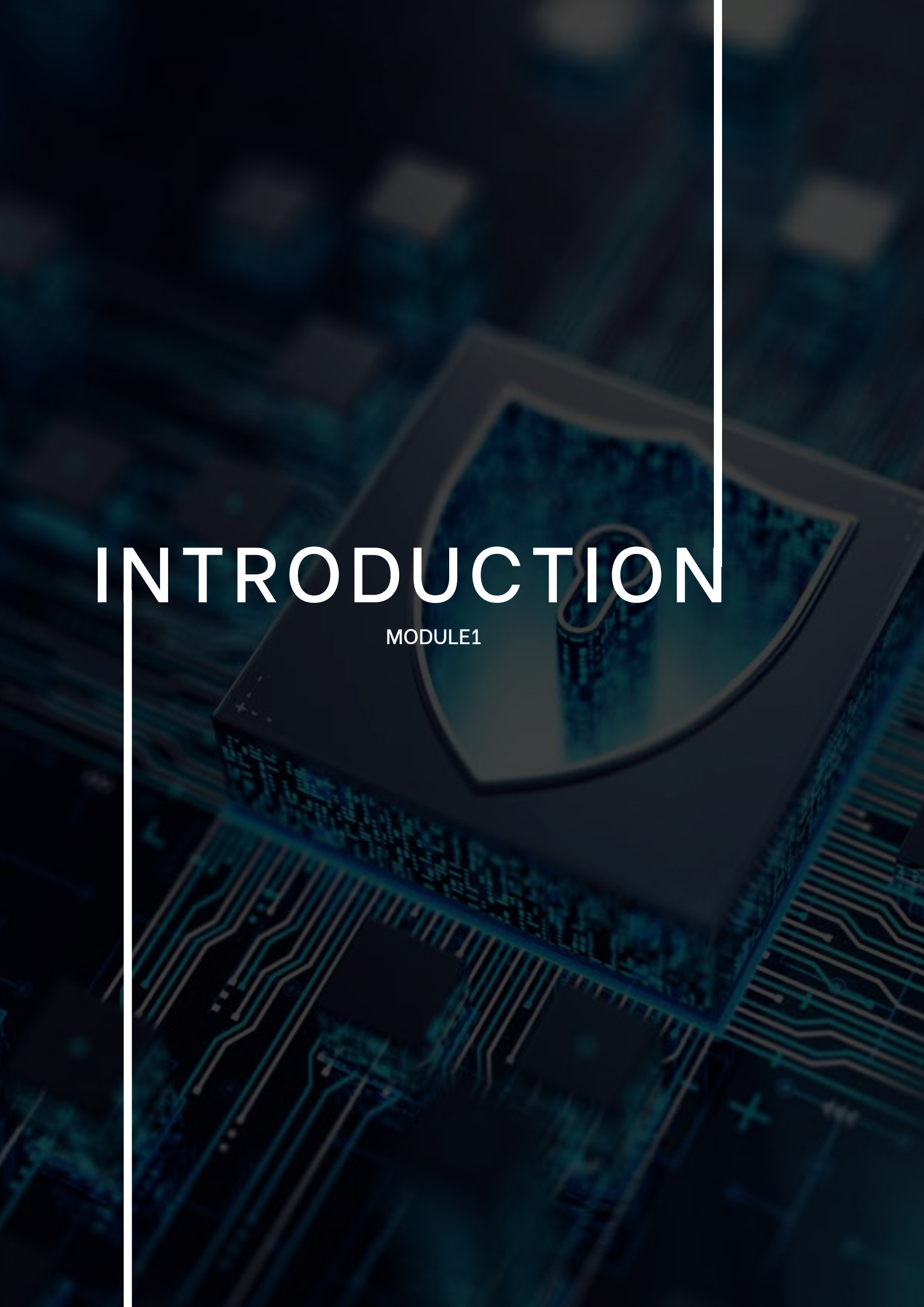


SOAR EDR PROJECT

THE DOCUMENTATION

TABLE OF CONTENTS



INTRODUCTION

MODULE1

What is EDR?

Endpoint Detection and Response (EDR) is a cybersecurity technology that continuously monitors and responds to mitigate cyber threats. EDR tools focus on detecting and investigating suspicious activities on hosts and endpoints.

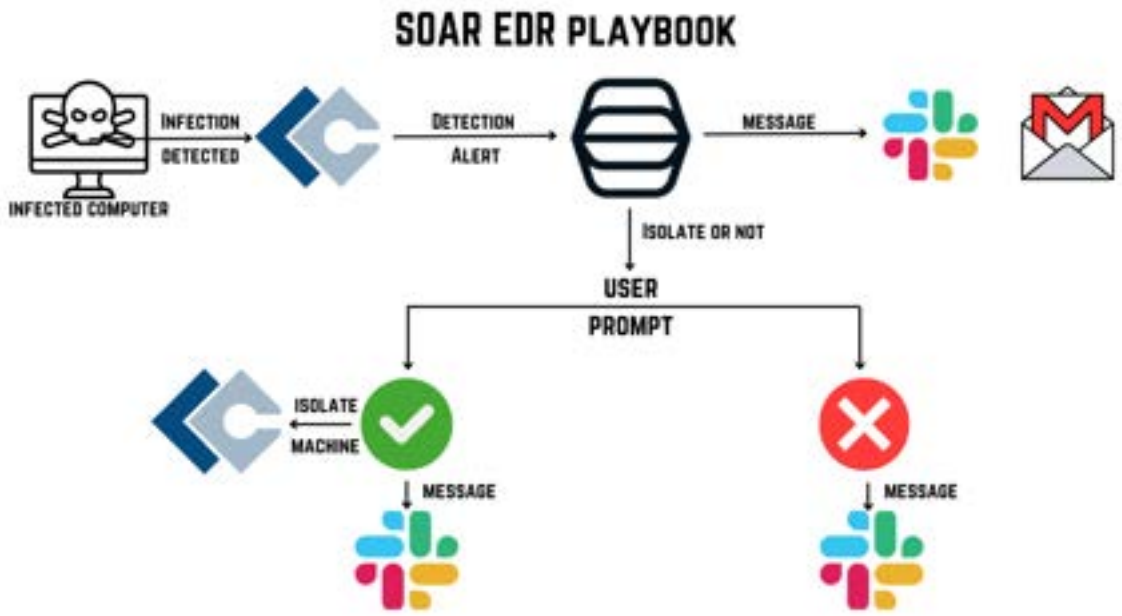
What is SOAR ?

Security Orchestration, Automation and Response (SOAR) refers to technologies that enable organizations to collect inputs monitored by the security operations team. SOAR allows companies to define incident analysis and response procedures in a digital workflow format.

Project Overview:

This project combines EDR and SOAR technologies to create an automated threat detection and response system. By integrating LimaCharlie (EDR) with Tines (SOAR), we've developed a workflow that detects potential threats, alerts security teams, and optionally isolates compromised machines with minimal human intervention.

Workflow:



Environment

This SOAR EDR project utilizes a combination of virtualization, endpoint detection and response (EDR) software, security orchestration and automated response (SOAR) platform, and simulated threat tools. Here's a detailed look at each component:

1. Windows Server:

- Operating System: Windows Server (version can be specified, e.g., Windows Server 2019)
- Purpose: Acts as the target endpoint for threat detection and response simulation
- Key Features: Supports running of enterprise applications, provides a realistic environment for testing security measures

2. VirtualBox:

- Type: Open-source hypervisor for x86 virtualization
- Purpose: Hosts the Windows Server virtual machine
- Benefits: Allows for isolated testing environment, easy snapshot and rollback capabilities

3. LimaCharlie:

- Type: Cloud-native EDR platform
- Purpose: Monitors the Windows Server for threats, provides real-time visibility into endpoint activities
- Key Features:
  - Sensor deployment on endpoints
  - Real-time process monitoring
  - Custom rule creation for threat detection
  - API for integration with other security tools

4. Tines:

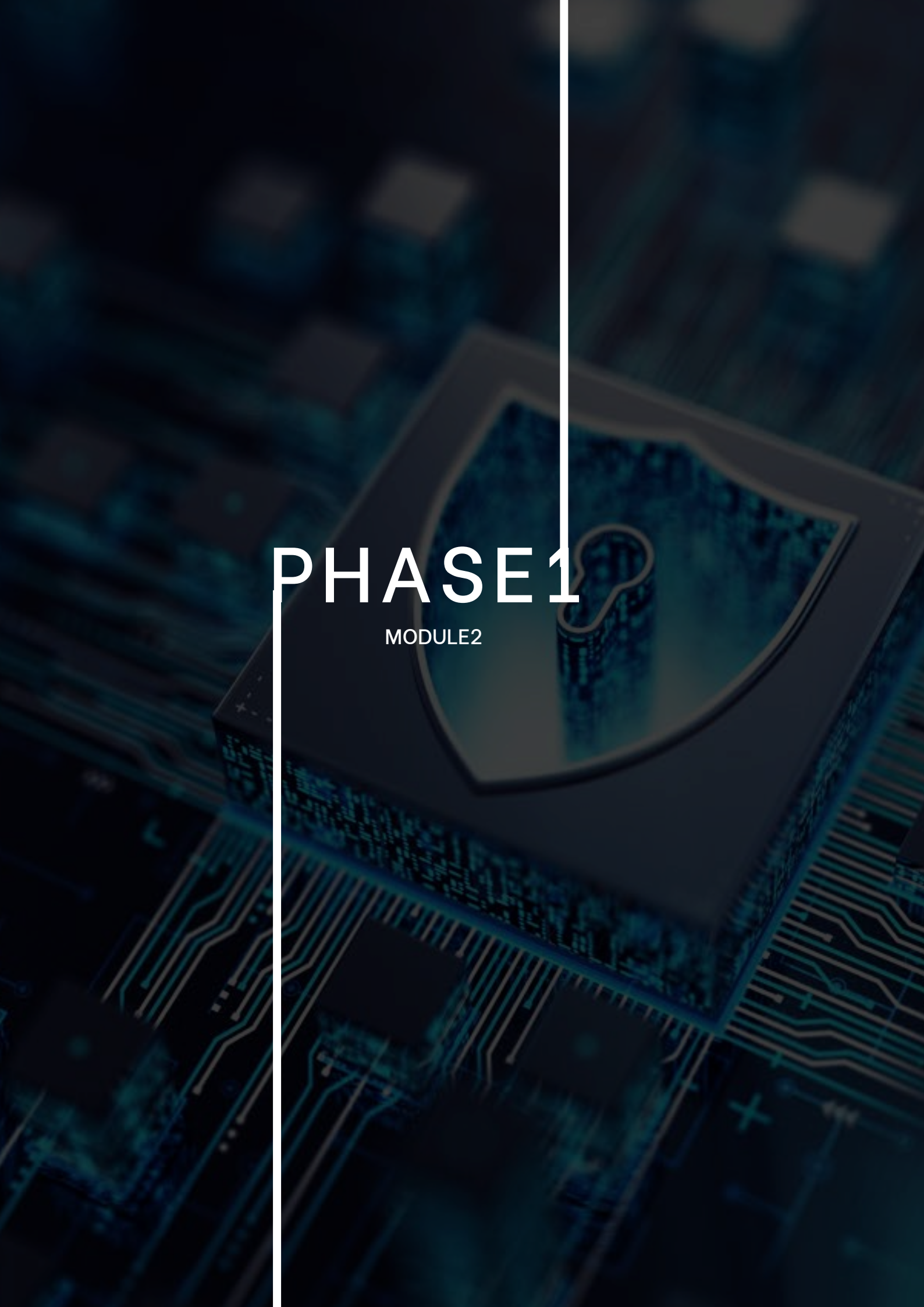
- Type: No-code automation platform for security operations
- Purpose: Orchestrates the response to threats detected by LimaCharlie
- Key Features:
  - Visual workflow creation ("Stories")
  - Integration with various security tools and communication platforms
  - Automated decision-making based on predefined criteria

5.LaZagne:
-Type: Open-source password recovery tool
-Purpose: Simulates a credential harvesting attack
-Usage: Deployed on the Windows Server to trigger LimaCharlie's detection capabilities

6.Email Integration:
-Purpose: Provides an additional notification channel for security alerts
-Implementation: Configured in Tines to send out notifications when threats are detected

7.Slack Integration:
-Type: Team collaboration and messaging platform
-Purpose: Offers real-time notifications and potential for team coordination on threat response
-Implementation: Integrated with Tines for immediate alert delivery to security teams

Conclusion
This environment creates a comprehensive ecosystem for testing and implementing automated threat detection and response. The Windows Server on VirtualBox provides a controlled testing ground, LimaCharlie offers robust EDR capabilities, Tines enables automated workflow execution, LaZagne simulates a realistic threat, and the email and Slack integrations ensure rapid communication of security events to relevant team members.

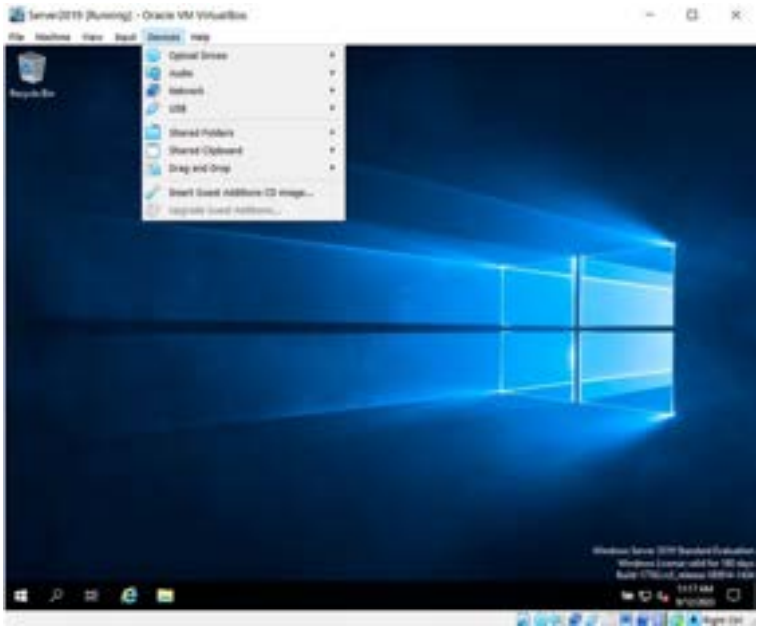

PHASE1
MODULE2

Preparing windows server

- Install Windows Server on VirtualBox (https://www.microsoft.com/en-us/evalcenter/download-windows-server-2019)
- Configure basic settings and network connectivity
- Ensure the system is updated and ready for sensor deployment

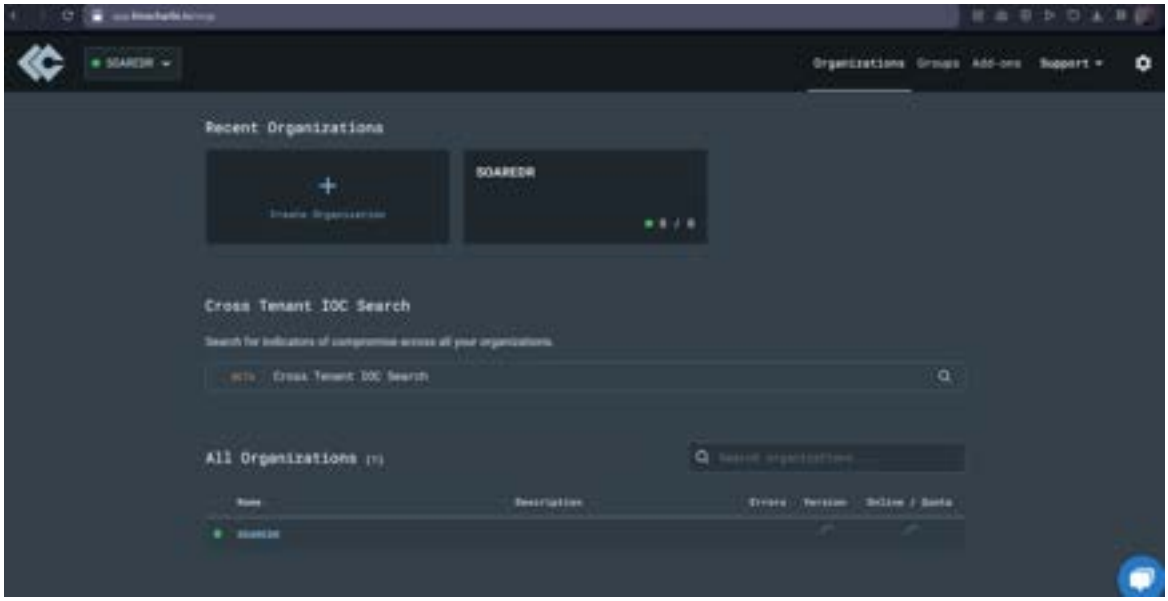
NB: you can use this blog for guidance

https://medium.com/@brianmwambia3/a-step-by-step-guide-setting-up-windows-server-2019-on-oracle-virtualbox-1a7b39090589



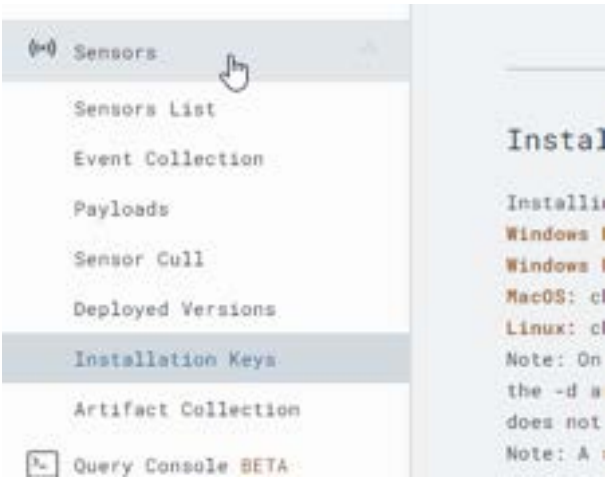
Setting up LimaCharlie:

- Create an account on LimaCharlie.io
- Set up a new organization named "SOAREDR"
- Generate an installation key for sensor deployment

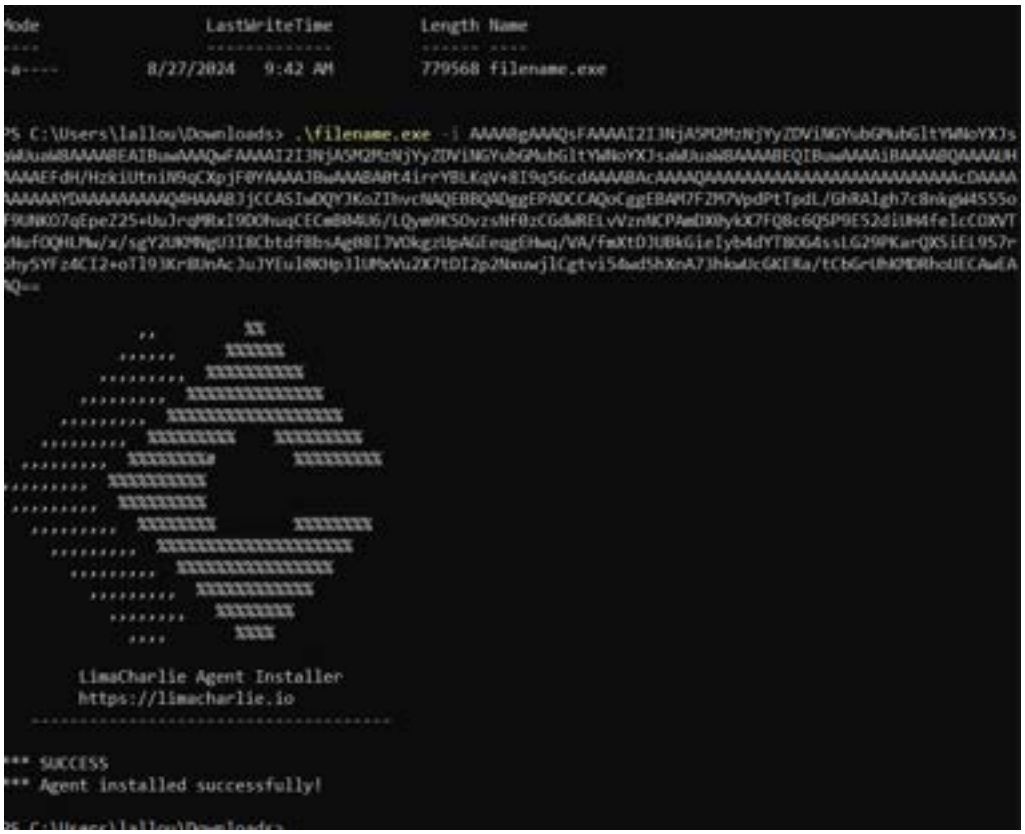


Deploying sensors

- Download the LimaCharlie sensor installation package
- Install the sensor on the Windows Server using the generated key
- Verify sensor connectivity in LimaCharlie dashboard



Run this command on your windows server:
bash <(curl -s https://raw.githubusercontent.com/refractionPOINT/lc-installer/master/installer.sh) \ -i YOUR_INSTALLATION_KEY_HERE



Sensor running


PHASE2
MODULE3

Introducing Lazagne

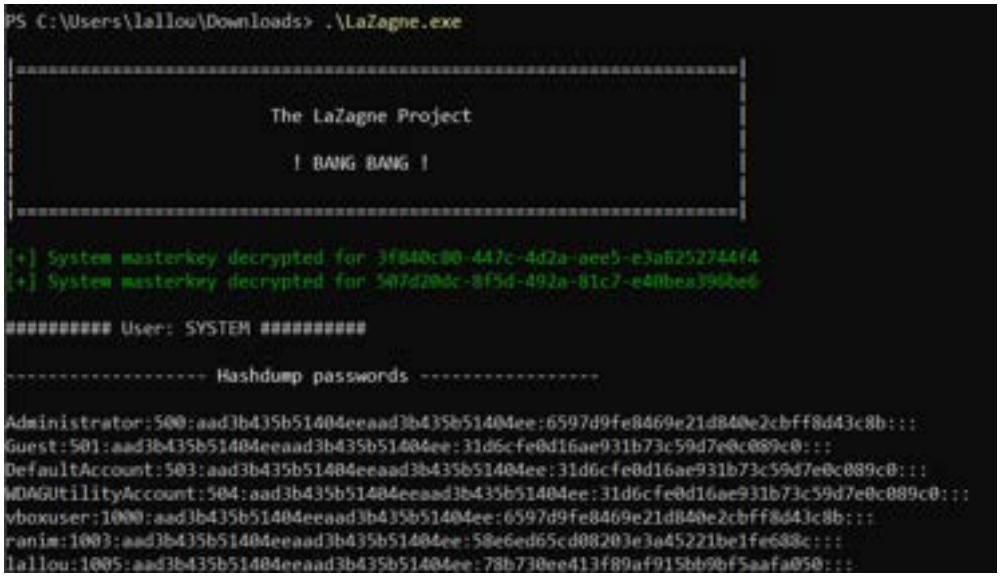
- LaZagne is an open-source application used to retrieve lots of passwords stored on a local computer. It's often used by attackers to harvest credentials, making it an ideal tool for simulating a security threat.

Setting up Lazagne

- Download LaZagne from its official GitHub repository
- Place the LaZagne executable in a directory on the Windows Server

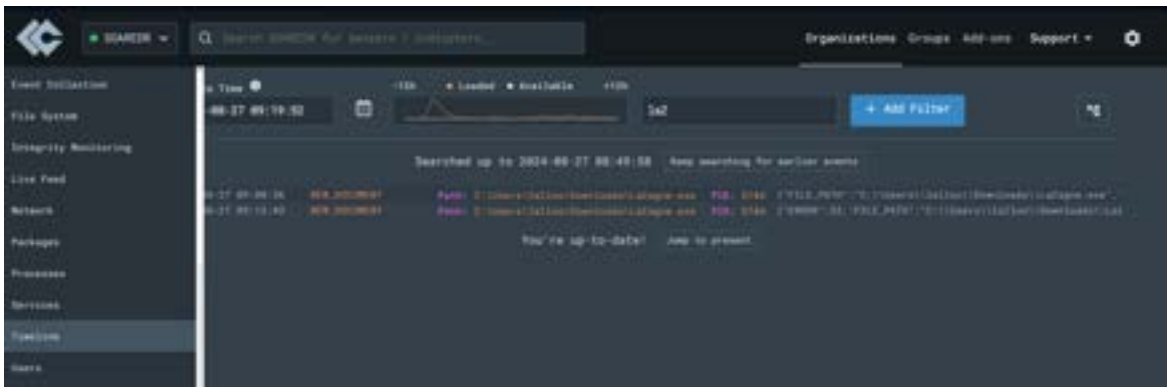
(https://github.com/AlessandroZ/LaZagne)

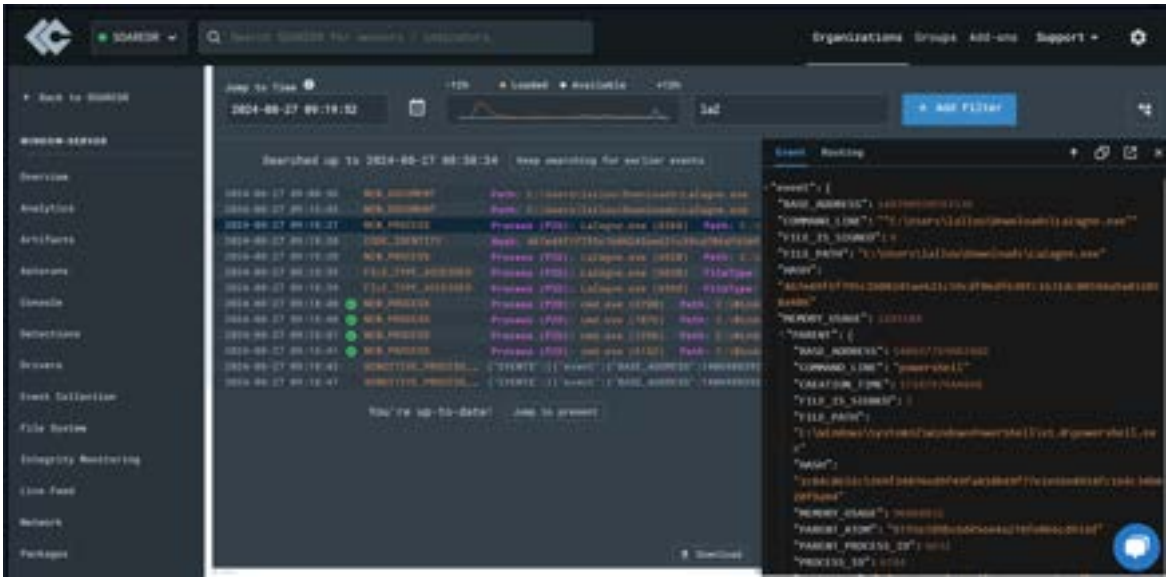
- Run LaZagne



Visualization in LimaCharlie

- Navigate to Sensor list -> Windows Server -> Timeline
- Observe the LaZagne process execution
- Verify that the rule triggers as expected





Creating a rule

- In LimaCharlie, navigate to the Automation section
- Create a new rule for credential access detection
- Set the rule to trigger when LaZagne process is detected

```
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: LaZagne.exe
    - case sensitive: false
      op: contains
      path: event/COMMAND_LINE
      value: LaZagne
    - case sensitive: false
      op: is
      path: event/HASH
      value: '3cc5ee93a9ba1fc57389705283b760c8bd61f35e9398bbfa3210e2becf6d4b05'

- action: report
  metadata:
    author: MyDFIR
    description: TEST - Detects Lazagne Usage
    falsepositives:
    - ToTheMoon
    level: high
    tags:
    - attack.credential_access
  name: MyDFIR - HackTool - Lazagne
```

Detect

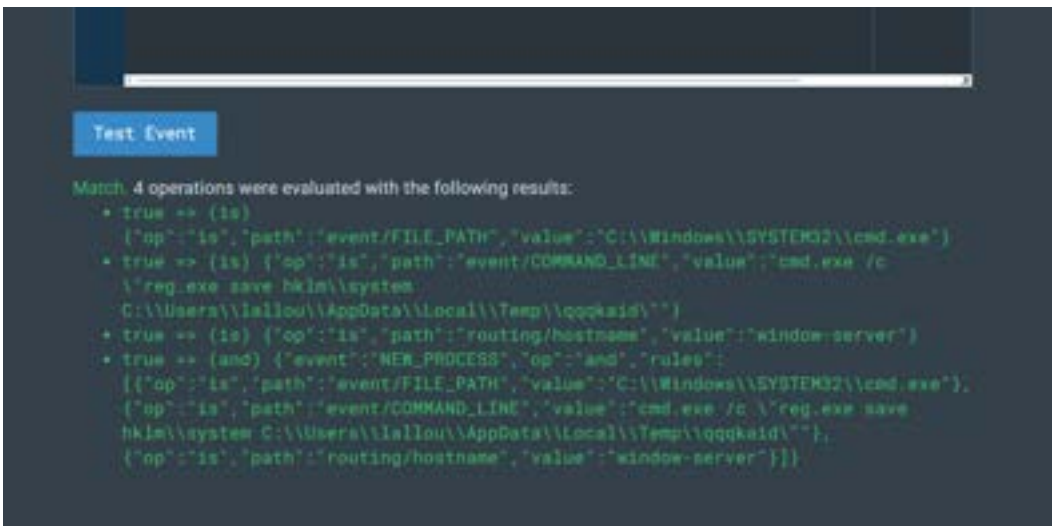
Restore

```
events:
- NEW_PROCESS
- EXISTING_PROCESS
op: and
rules:
- op: is windows
- op: or
  rules:
  -case sensetive false
   op: ends with
   path: event/FILE_PATH
   value: lazagne.exe
  -case sensetive: false
   op: ends with
   path: event/COMMAND_LINE
   value: all
  -case sensetive: false
   op: contains
   path: event/FILE_PATH
   value: \\LaZagne.exe
  - case sensitive: false
    op: is
    path: event/HASH
    value: '467e49f1f795c1b08245ae621c59cdf86df630fc1631
```

Respond

```
- action: report
  metadata:
    author: Ranim
    description: Detects LaZagne (SOAR-EDR tool)
      from view
    falsepositives:
    - To the moon
    level: medium
    tags:
    - attack.credential_access
  name: Ranim-HackTool-Lazagne [SOAR-EDR]
```

Simulating the rule



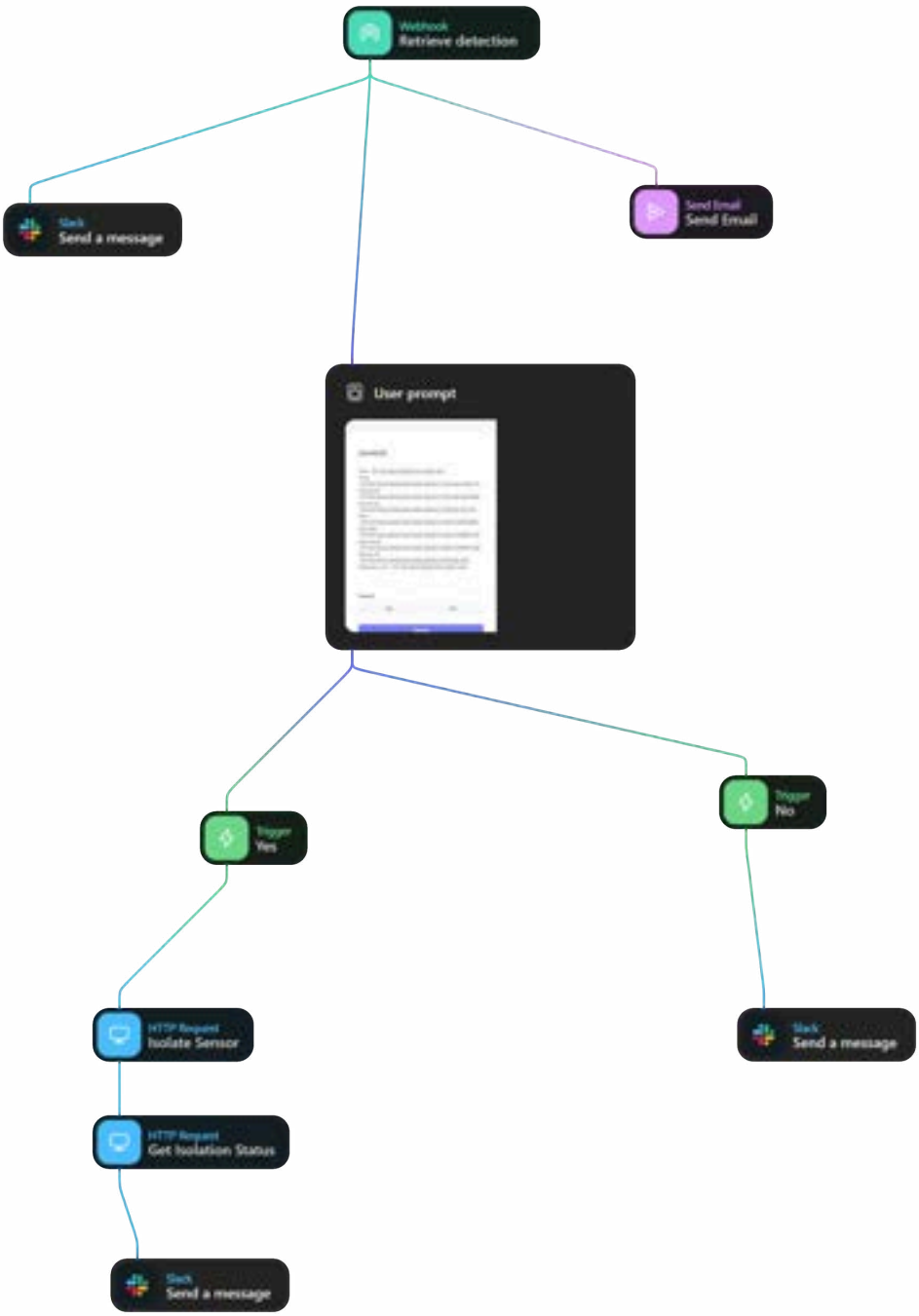

PHASE3
MODULE4

Setting up Tines

- Create an account on Tines
- Set up a new project for the SOAR EDR integration

Preparing the story

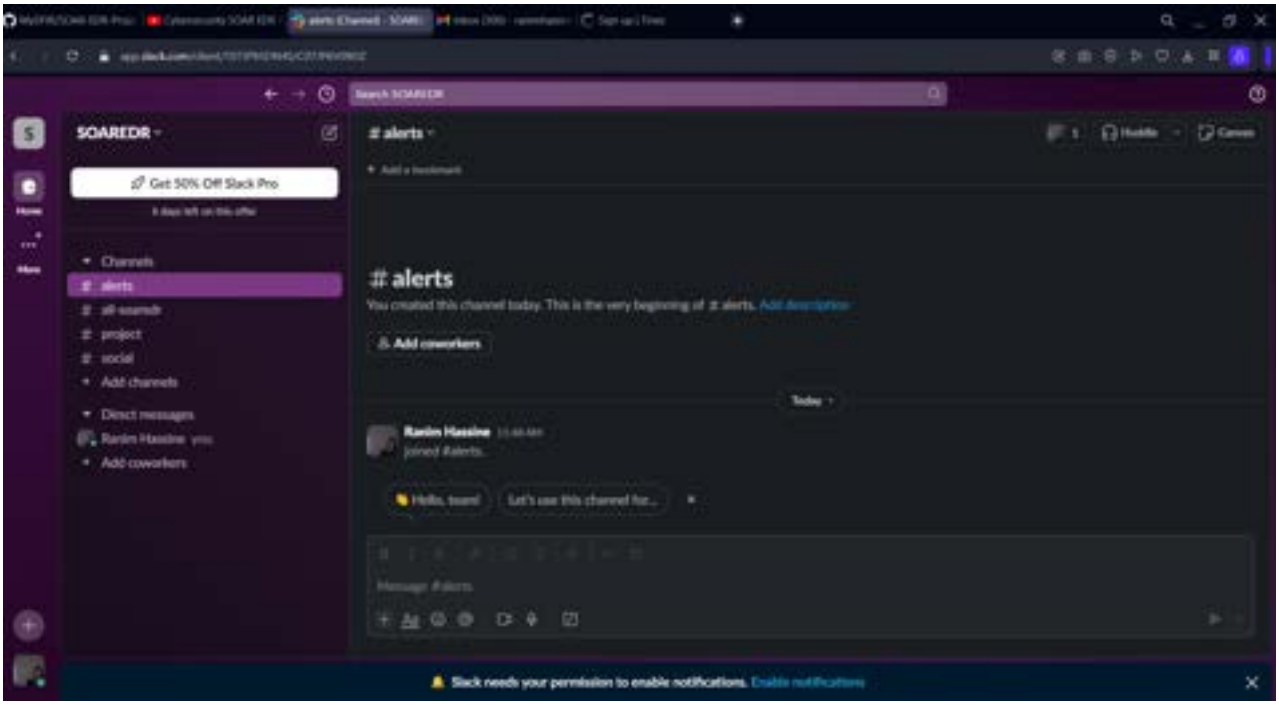
- Create a new story in Tines
- Design the workflow to receive alerts from LimaCharlie
- Implement logic for user prompts and machine isolation decisions



tines

Getting notified on slack and email

- set up you slack account
- Configure Slack integration in Tines
- Set up email notifications
- Test the notification system with a simulated alert



SUMMARY

This project demonstrates the integration of Security Orchestration, Automation and Response (SOAR) with Endpoint Detection and Response (EDR) technologies to create a robust, automated cybersecurity solution. The key components and workflow are as follows:

1. Environment:

- Windows Server running on VirtualBox, simulating a target endpoint
- LimaCharlie as the EDR solution
- Tines as the SOAR platform
- LaZagne for threat simulation

2. Workflow:

- LimaCharlie monitors the Windows Server for suspicious activities
- LaZagne is used to simulate a credential harvesting attack
- LimaCharlie detects the threat and triggers an alert
- Tines receives the alert and initiates an automated response
- The system notifies security personnel via email and Slack
- Tines prompts for a decision on whether to isolate the affected machine
- Based on the decision, Tines either instructs LimaCharlie to isolate the machine or simply logs the event

3. Key Achievements:

- Successful integration of EDR (LimaCharlie) and SOAR (Tines) platforms
- Automated threat detection and response capabilities
- Reduced response time to potential security incidents
- Improved visibility into endpoint activities
- Enhanced team communication through multi-channel alerts

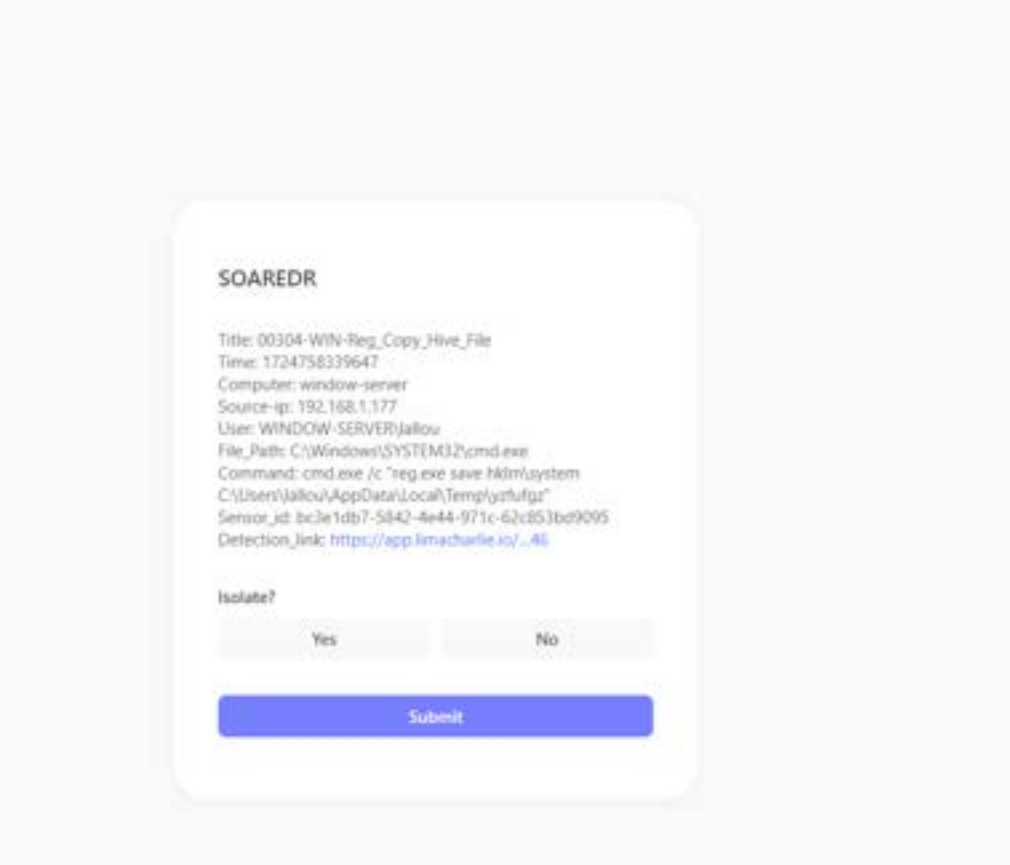
4. Benefits:

- Minimized human intervention in initial threat response
- Standardized and repeatable incident response procedures
- Increased efficiency in handling security events
- Potential for scaling across larger networks

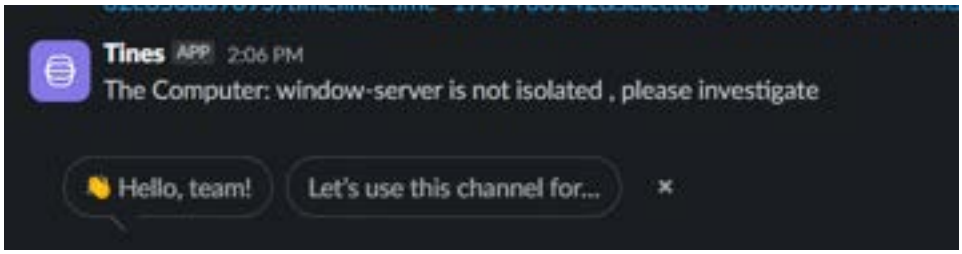
This project showcases the power of combining EDR and SOAR technologies to create a more responsive, efficient, and robust cybersecurity infrastructure. By automating key processes and providing clear workflows, it enhances an organization's ability to detect, analyze, and respond to threats quickly and consistently.

ANNEX

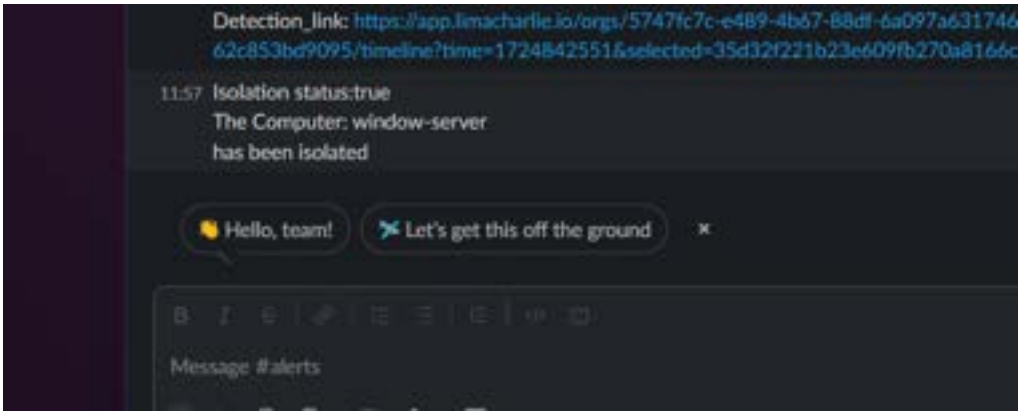
The user prompt when infection detected

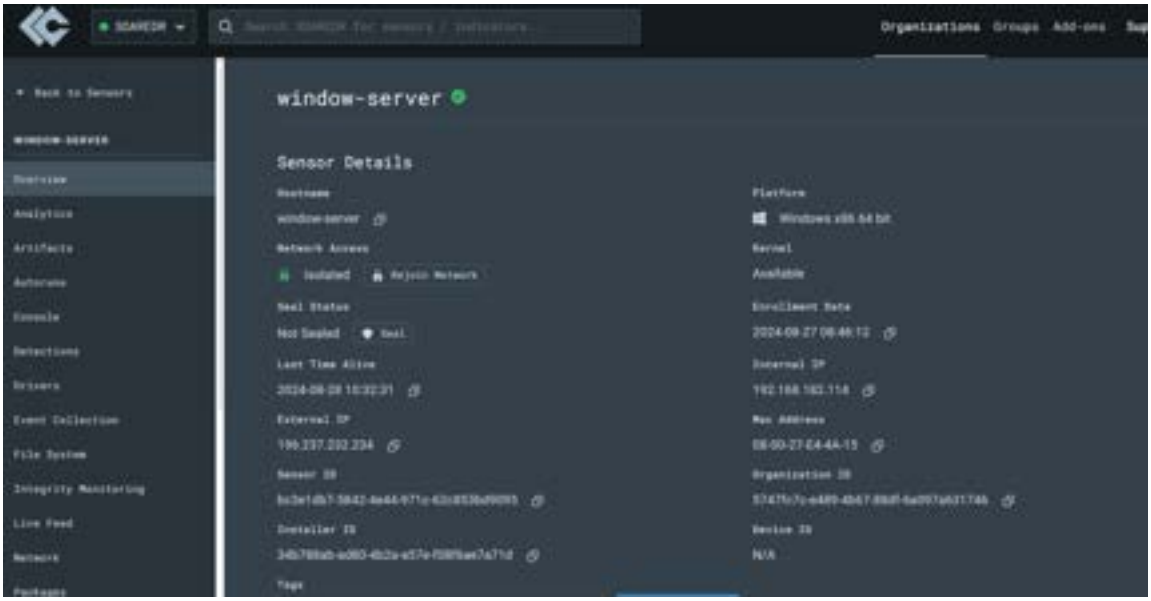


If the answer is no



If the answer is yes






HAPPY LEARNING