***A Project Report on***

***SOAR\_EDR (Security Orchestration Automation & Response)-(Endpoint Detection & Response) Playbook***

***MASTER OF SCIENCE (CYBER SECURITY)***

***Submitted By***

***Kanika Moulekhi***

***Er. No 2########2***

***Under the Guidance of***

***Ms XXXXXXXXX***

***Assistant Professor***

***U.O.U. MODEL STUDY CENTER, $$$$$$$***

***CENTRE CODE: 16000***

******

***School of Computer Science and IT,***

***Uttarakhand Open University, $$$$$$***

***2025***

**U.O.U. Model Study Centre, $$$$$$$$ Centre Code: 16000**

**School of Computer Science and IT**

**CERTIFICATE**

This is to certify that the project report titled “***SOAR\_EDR (Security Orchestration Automation & Response) Project*”** submitted by **Kanika Moulekhi,** bearing **Er. No 2######2** in **Master of Science (Cyber Security) - Second Semester** is a record Bonafide work carried out by me. The results embodied in this report have not been submitted by me to any other University for the award of any degree.

**Student Signature**

**Supervisor Signature Co-ordinator Signature**

**ACKNOWLEDGEMENT**

Degrees are important, but in today’s world, we see how experience is demanded from the moment we step out of college and begin searching for jobs. Theory often takes a back seat, while the driver everyone is looking for is practical skill. This project has laid that foundation for me.

I would like to express my sincere thanks to **Dr. &&&&&&**, Coordinator of **U.O.U. Model Study Centre (Centre Code: 000)**, for overseeing the academic process and providing the official framework for project submission.

I am especially grateful to my project guide **Ms. $$$$$$**, whose insights and feedback greatly helped shape the direction of this work.

I also extend my thanks to the faculty of the **School of Computer Science & Information Technology, Uttarakhand Open University**, for their indirect support and academic resources.

A special mention goes to my **family and friends** for their continued motivation and emotional support throughout the duration of this project. My project was not only mine, but also became their priority for a while.

I also acknowledge the role of **online resources, tools, and platforms** that helped me research, develop, and complete this project successfully. These resources made the process simpler, and the fact that vast, ocean-level knowledge is now accessible at our fingertips — just a few clicks away — makes me genuinely thankful for the existence of the Internet.

**CONTENTS**

**Abstract**

**List of Figures List of Tables List of Screens**

**Symbols & Abbreviations**

**1. INTRODUCTION**

**1.1 Motivation**

**1.2 Problem definition**

**1.3 Objective of Project**

**1.4 Limitations of Project**

**1.5 Organization of Documentation**

**2. LITERATURE SURVEY**

**2.1 Introduction**

**2.2 Existing System**

**2.3 Disadvantages of Existing system**

**2.4 Proposed System**

**2.5 Conclusion**

**3. ANALYSIS**

**3.1 Introduction**

**3.2 Software Requirement Specification**

3.2.1 User requirement

3.2.2 Software requirement

3.2.3 Hardware requirement

**3.3 Content diagram of Project**

**3.4 Algorithms ad Flowcharts**

**3.5 Conclusion**

**4. DESIGN**

**4.1 Introduction**

**4.2 DFD / ER / UML diagram (any other project diagrams)**

**4.3 Module design and organization**

**4.4 Conclusion**

**5. IMPLEMENTATION & RESULTS**

**5.1 Introduction**

**5.2 Explanation of Key functions**

**5.3 Method of Implementation**

**5.2.1 Forms**

5.2.2 Output Screens

5.2.3 Result Analysis

**5.4 Conclusion**

**6. TESTING & VALIDATION**

**6.1 Introduction**

**6.2 Design of test cases and scenarios**

**6.3 Validation**

**6.4 Conclusion**

**7. CONCLUSION: First Paragraph - Project Conclusion Second Paragraph - Future enhancement REFERENCES**

**1. Author Name, Title of Paper/ Book, Publisher’s Name, Year of publication**

**2. Full URL Address**

**ABSTRACT**

Cyber Crimes are serious concern of Today’s world. At an average in India almost **5000 Cyber-Crime complaints** are registered every day. And on Organisational Level **460 Cyber-Attacks** every day.

On such huge slots the Cyber-Criminals are able to scam, destroy, change system that give them profits and loss to the common people. Such Crimes generally start when they somehow gain access to our credentials, Organisation’s important Credentials. This project deal with it By **Detecting threats** and then **isolating the affected machine**.

This project **SOAR-EDR (Security Orchestration Automation & Response)- (End point Detection & Response)** handles such attacks via **Sensors** installed the machine. The Sensors generate log files and give remote access of the machine to the **analyst** to deeply analyse any threats and if there are any, then **isolate and block access to internet.**

The technology I used here are **Lima Charlie-** for Endpoint Detection and access to the target machine remotely.  **LaZagne** – a password stealing application for testing and emitting an **Cyber-Attack** to steal **Credentials.** And **Tines,** it is a no-code / low-code SOAR platform (Security Orchestration, Automation, and Response) used by cybersecurity teams to automate repetitive security tasks without writing traditional code**.**  And **Slack** is a messaging platform for teams and workplaces, designed to improve communication, collaboration, and productivity. It’s widely used by companies, schools, and communities — especially in tech, cybersecurity, and remote team.

It works in flow like Lima Charlie detects and give access to logs and all data. We connect Lima Charlie to tines and in tines story board we create story that joins the data to the slack, to email, from there if we get detection, we see it all happening live in story as its directly connected to the Lima Charlie, we can decide after studying the detection if its harmful or not, isolated or not.

This system is fully automated where an analyst, cyber engineer can now see the detections, its details and can decide If system should be isolated or not. And it also sends automatic emails and message to keep record.

**LIST OF FIGURES / TABLE / SCREEN**

**DIAGRAMS**

Diagram1.1 flow chart of the application

Diagram 1.2 showing the beginning process of VM setup and lima charlie integration

Diagram 1.3 showing another step of connecting further with Tines and slack

**SCREEN SHOTS**

Screen shot 2.1 dashboard of limacharlie we can see its connected to the VM windows name [**desktop-1cacekm**](https://app.limacharlie.io/orgs/c61fe42e-d827-44c4-815b-5142ab9acd05/sensors/411b525e-2635-460a-8eaa-0d5af3518c7b/overview?from=sensors) as its online, can see its sidebar with all features.

Screen shot 2.2 running LaZagne in the VM we can see it extracted some passwords.

Screenshot 2.3 we can see that the detections are here and details also telling us about the threats we face.

Screen shot 2.4 we can see that the detection we saw in the limacharlie are visible here in tines ‘s webhook.

screen shot2.5 we can see that alerts are generated here giving full information about affected system we click on link and visit deeper.

screen shot 2.6 after clicking on link we reach here and we can do detailed and deep analysis to decide if we should isolate machine or not?

Screenshot 2.7we can a message in burner email as well and many more so we can send alerts via prebuilt message in email and in slack.

Screen shot 2.8 we now after studying the machine’s detail can decide if we want to isolate it or not, here we will choose **NO**

Screenshot 2.9 here we can clearly see message telling us that machine is not isolated hence, no is success.

Screenshot 2.10 we are going to say yes to isolate.

Screenshot 2.11 we can see last message here that confirm that machine is isolated.

Screen shot 2.12 here in limacharlie dashboard too we can see system is isolated.

Screen shot 2.13 we see here system is isolated

Screen shot 2.14 in the VM we can clearly see internet is connected despite that it’s not working.

Screen shot 2.15 we can now rejoin easily by simple click.

Screen shot 2.16 now after rejoining it works fine.

**TABLES**

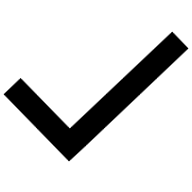
Table 3.1 testcases and detections

**SYMBOL AND ABBREVATIONS**

**EDR:** Endpoint Detection and Response

**SOAR:** Security Orchestration Automation and Response

**Silo:** A data silo is a situation where a set of data is isolated from the rest of the organization, SOAR EDR, STEM

 success

**VM:** Virtual Machine

1. **INTRODUCTION**
   1. **MOTIVATION**

Technological advancements are accelerating rapidly. There was a time when all data had to be recorded and stored manually, either handwritten or typed and filed in physical forms. Retrieving specific information from those records was time-consuming and inefficient.

With the rapid growth of technology, data management shifted to digital platforms. Information is now stored in digital files, external drives, and cloud systems, making access and retrieval significantly faster and more efficient. Not only organizational data but also personal data became easier to store and access. However, this digital transformation brought new challenges and rise to a new different kind of Crime- empire.

As more data moved online, the threat of cybercrime surged. Today, a single careless click or phone call can lead to a serious breach if proper safeguards are not in place. With cyber threats evolving rapidly and targeting individuals, organizations, and even governments, traditional security methods are no longer enough.

There is an increasing need for automated, scalable, remote systems capable of detecting, responding to, and neutralizing attacks in real-time. This has led to the integration of **SOAR (Security Orchestration, Automation, and Response) with EDR (Endpoint Detection and Response)**, streamlining complex incident response processes and strengthening overall cybersecurity posture.

This section explores the evolution of data handling and the growing need for modern cybersecurity solutions like **SOAR and EDR**. This project here will demonstrate a system that can be integrated in our systems so we can guard the systems even remotely.

**1.2 PROBLEM DEFINITION**

Security teams are often overwhelmed by the volume of alerts generated by multiple tools. If there are no-centralized automation, critical threats may be missed or delayed in response. Manual investigation is time-consuming and inconsistent. The challenge is to build a **unified automated solution** that can detect suspicious activities on endpoints and automatically take predefined actions to contain them, thus reducing response time and analyst fatigue**.**

The problem is that if a device is under attack and attacker is able to get into the system, they can access other systems as well especially if systems are interconnected (like generally in organisations happens). If attacker is able to get access to the passwords or other sensitive information. The whole system can be on the verge of collapse. This EDR\_SOAR setup help in resisting these types of attacks or make sure that affected device will do minimal damage.

* 1. **OBJECTIVES OF THE PROJECT**
* To create a basic working model of a SOAR (Security Orchestration Automation and Response)system integrated with an EDR (Endpoint Detection and Response) solution.
* To demonstrate automation in response to certain threat indicators (e.g., malicious IP, hash detection).
* To use open-source tools and platforms (e.g., Lima Charlie, Tines, Slack, Virtual Machine) for building a functional pipeline.
* To showcase alert triage, enrichment, and automated mitigation through workflows.
  1. **LIMITATIONS OF THE PROJECT**
* The project is for educational and demonstration purposes; it may not be suitable for production-level deployment.
* Limited to free-tier or trial APIs of platforms like Slack, and Lima Charlie and Tines.
  + Automation rules are basic and may not cover complex attack chains.
  + Endpoint coverage and real-time response speed may be affected by API rate limits or service delays.
  + Since the its not paid regularly updating JWT keys of Lima Charlie to the Tines.

**1.5 ORGANIZATION OF DOCUMENTATION**

This project documentation is structured into seven chapters, each focusing on a specific phase of the project development:

* **Chapter1: Introduction** Provides an overview of the topic, the motivation behind the project, the problem definition, and the primary objectives.
* **Chapter2: Literature Survey** Covers background research on SOAR, EDR, and related cybersecurity frameworks and tools.
* **Chapter3: Analysis** Discusses the requirements, feasibility, and challenges. It outlines the system scope and examines various components needed for implementation.
* **Chapter4: Design** Presents the system architecture, design decisions, flow diagrams, and the planned structure of tool integration.
* **Chapter5: Implementation** Describes the actual development phase, including tools used, workflow automation, code snippets, and integration of components like Tines, Slack, Smart Draw, LaZagne and Lima Charlie.
* **Chapter6: Testing** Explains the testing strategies used, scenarios considered, and outcomes. It also highlights the difficulties encountered and how they were resolved.
* **Chapter7: Conclusion** Summarizes the work done, key findings, and proposes areas for future enhancement or further exploration in the SOAR-EDR domain.

. **2. LITERATURE SURVEY**

**2.1 INTRODUCTION**

This chapter reviews the background and previous work related to cybersecurity automation, specifically SOAR (Security Orchestration, Automation, and Response) and EDR (Endpoint Detection and Response). It explores existing tools, their limitations, and the need for a more integrated and efficient solution.

**2.2 EXISTING SYSTEM**

Traditional cybersecurity operations rely on separate tools for monitoring, detection, and response. SOC **(Security Operations Centre) analysts** often use **SIEMs (like Splunk, ELK)** for log collection and alerting, while EDR tools like **CrowdStrike or Sentinel-One** are used for endpoint visibility. Although these systems are powerful, they often operate in silos, requiring manual correlation and delayed response to threats.

**2.3 DISADVANTAGES OF EXISTING SYSTEM**

* **Lack of Integration**: Tools often do not communicate with each other, leading to data **silos**. And if problem is isolated or hidden. It’ll be hard to track until the irreplaceable damage had been done.
* **Manual Workload**: Analysts spend time switching between platforms and investigating alerts manually. Multiple tabs, then checking them serial wise or one can miss any message or log, can make things difficult to catch and response to it.
* **Delayed Response Time**: Slow triage and investigation may allow threats to spread. With constant internet connection with other devices with the threat is not caught on time, whole network can be affected.
* **High False Positives**: Existing systems generate a large number of alerts, overwhelming analysts. And sometimes full access of the alerts is not given, increasing manual workarounds
* **Limited Automation**: Most systems lack the ability to automatically respond based on threat intelligence**.** In many systems manual coding is very much required, and between analysing threat, its effect it can delay in response time

**2.4 PROPOSED SYSTEM**

The proposed system leverages **SOAR** tools and open-source **EDR** frameworks to build an automated pipeline that can detect, analyse, and respond to threats in near real-time.

Integration of tools such as **Tines** for real time trigger, testing and linking different platform together like **slack, automatic mailing** systems and joining system via **credentials** lack for getting the system information inform of alerts and direct link to the Lima Charlie aims to create a unified workflow that reduces analyst burden and improves response time.

**2.5 CONCLUSION**

The literature survey shows a clear need for tighter integration and automation in cybersecurity operations. While existing tools serve their purpose individually, a coordinated system using SOAR and EDR concepts can significantly improve threat handling and reduce fatigue in security teams.

This system is light weight, considering it uses online platforms and one has option to run them in browser or download application. Both works fine.

**3.ANALYSIS**

**3.1 INTRODUCTION**

This chapter focuses on the analysis stage of the project. It outlines the software and hardware requirements, details the key components of the system, and presents a clear understanding of how the solution will work in practice through diagrams and flowcharts.

**3.2 SOFTWARE REQUIREMENT SPECIFICATION**

**3.2.1 User Requirements**

* Fully working internet connection.
* The system must automate incident detection and response.
* The user must receive alerts in real time.
* The interface should be simple and accessible (e.g., Slack notifications).
* Automation workflows must require minimal manual effort once configured**.**

**3.2.2 Software Requirements**

* Tines: For building custom SOAR workflows and automation sequences. Connecting Lima Charlie and slack to its story board and burner email for testing.
* Lima Charlie: Serves as the EDR tool to monitor endpoints and generate telemetry.
* Slack: Used to receive alert notifications and workflow results.
* Burner Email Services (e.g., temp-mail): Used to register dummy accounts safely during testing.
* Windows OS (VM): Hosted via Oracle Cloud Free Tier to simulate the endpoint.

**3.2.3 Hardware Requirements**

* A system with stable internet access to access cloud services.
* Oracle Cloud infrastructure (Free Tier) for hosting the Windows virtual machine.
* Minimum of 4 GB RAM and dual-core CPU for running VM smoothly.
* Personal laptop/PC to configure and monitor all services.
* At least 25-30 Giga byte free space in C drive as even though Oracle machine is of less Gb it feels pretty quickly with so many alerts generation and working. Also, it’s not possible to store the VM machine in D or any other drive it does work.

**3.3 Component Diagram of the Project**

**Below is a textual description –**

ORACLE VIRTUAL BOX

STARTING VIRTUAL MACHINE (HAS LIMACHARLIE SENSOR INTEGRATED IN IT).

RUNNING Lazagne IN THE VM (A PASSWORD STEALING/RECOVERY TOOL).

LIMACHARLIE DETECTING IT IN THE DASHBOARD TIMELINE AND SEND IT TO THE TINES (WE JOINED THEM VIA CREDENTIALS).

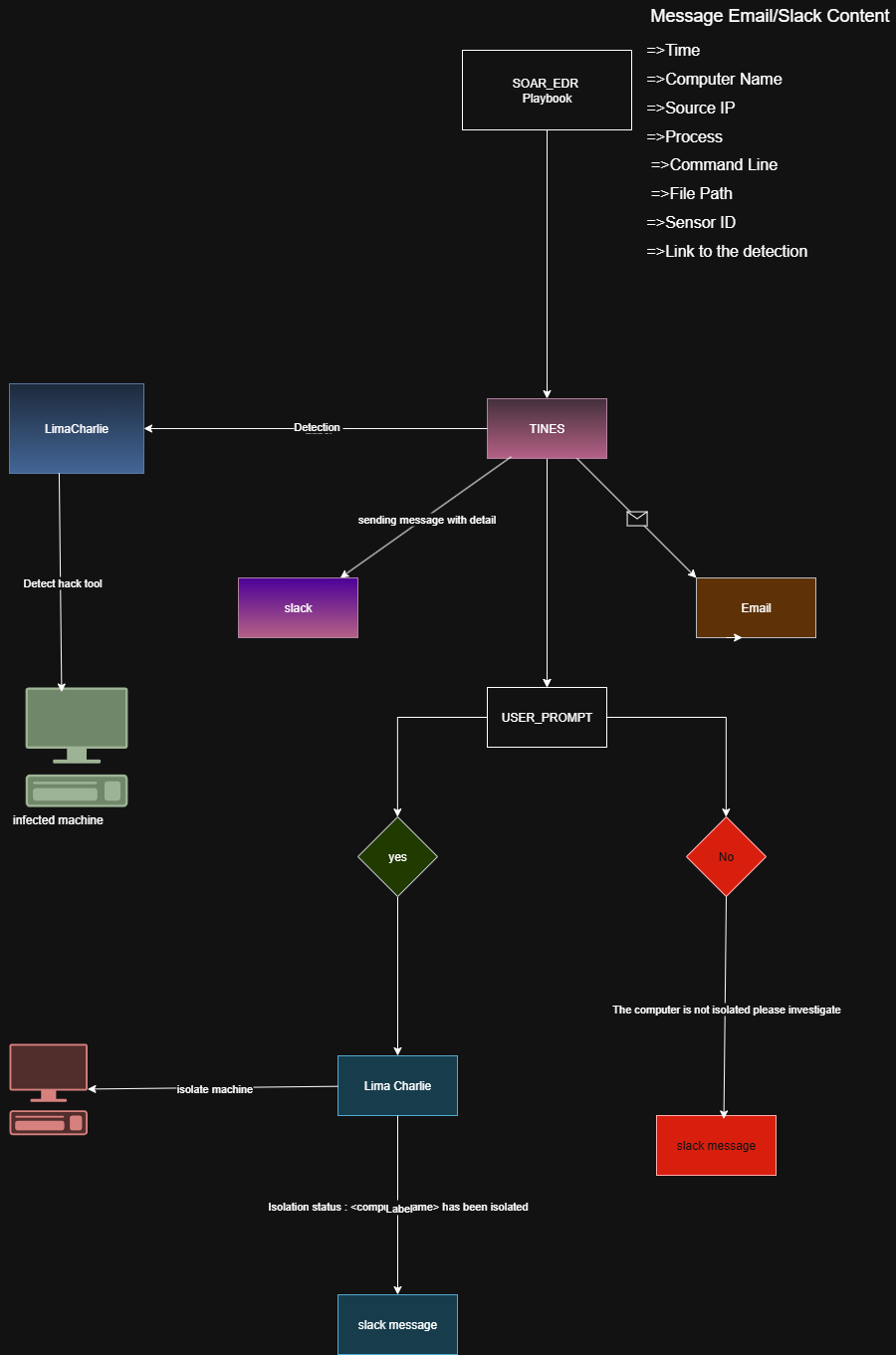
THROUGH TINES STORY BOARD AN EMAIL AND A MESSAGE IS SENT TO SLACK (THE LINK TO DEVICE ID IN THE MESSAGE ONE CAN STUDY IF THE DETECTED **PROCESS** IS THREAT OR NOT AND CAN DECIDE TO ISOLATE MACHINE OR NOT).

THE MESSAGE WILL BE GENERATED IN SLACK AS WELL AS IN BURNER EMAIL, IF THE MACHINE IS ISOLATED OR NOT,

**3.4 ALGORITHMS AND FLOWCHARTS**

Example Flowchart (text version you can draw later):

1. Start
2. Configure Lima Charlie on Oracle Cloud VM
3. Set up the sensors in the VM
4. Tines receives the EDR alerts from LimaCharlie
5. Tines performs actions (e.g., generate Tines-process and then slack alert)
6. Slack notifies the user in real-time
7. Optional: Tines triggers automated response actions
8. End



**Diagram1.1 flow chart of the application**

**3.5 CONCLUSION**

The analysis phase helped define all critical components, from software to user-facing interfaces. By using LimaCharlie, Tines, and Slack in combination with a Windows VM hosted on Oracle Cloud, the system is designed to automate cybersecurity workflows effectively and affordably. We’ll also be using the burner emails to check the automatic mailing systems. This approach avoids the complexity of heavier platforms, focusing on real-world, accessible tooling.

**4. DESIGN**

**4.1 INTRODUCTION**

This chapter outlines the design aspects of the project. It defines the structure, modular design, and logical flow that enable the system to perform its functions efficiently. The focus is on how different components interact and how the automation is structured internally. DIFFERENT WEB-APPLICATIONS, OPEN APIs, and PASSWORD RECOVERY TOOLS to stimulate real-time attacks. This system built and then test the components to check its authenticity

**4.2 PROJECT DIAGRAMS (DFD / ER / UML)**

The core flowchart already presented in the Analysis chapter. The primary system logic and flow were visualized through that single flowchart. Now this the first step—

****

Diagram 1.2 showing the beginning process of VM setup and lima charlie integration

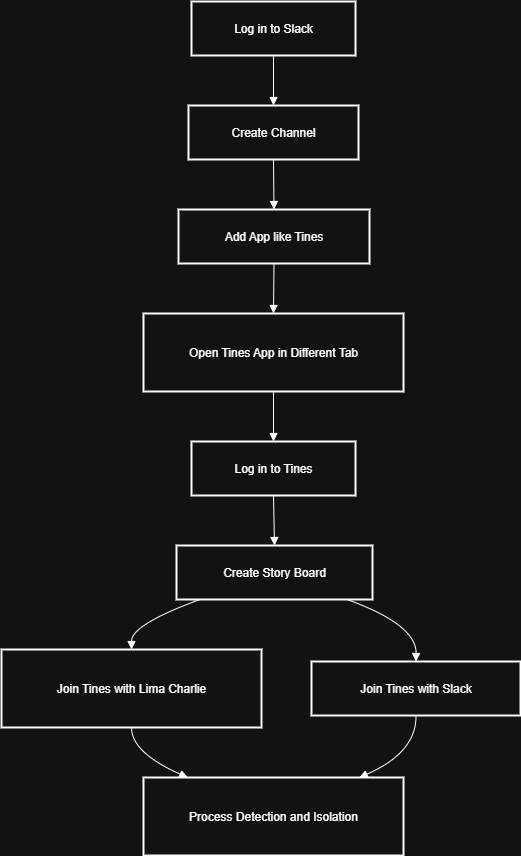


Diagram 1.3 showing another step of connecting further with Tines and slack

**4.3 MODULE DESIGN AND ORGANIZATION**

The system is broken down into four main functional modules:

1. Monitoring Module – LimaCharlie
   * Installed on the Windows VM hosted via Oracle Cloud.
   * Captures endpoint telemetry, monitors behaviour, and sends data.
2. Automation Module – Tines
   * Receives input from LimaCharlie (e.g., detection signals).
   * Builds and executes workflows based on predefined triggers (e.g., if suspicious behaviour → send alert).
3. Alerting Module – Slack
   * Tines pushes notifications and workflow results to Slack channels.
   * Used for human visibility into automated processes.
4. Testing & Deployment Support – Burner Emails
   * Temporary email accounts used to simulate user activity and account creation.
   * Ensures safe and isolated testing without real-world data risks.

Each module is lightweight, cloud-friendly, and chosen to maintain simplicity while achieving core SOAR functionalities.

**4.4 CONCLUSION**

The design approach of the project emphasizes simplicity, modularity, and real-time response. Even without complex diagrams, the project structure remains easy to understand and replicate. The modular design using Tines, LimaCharlie, and Slack enables rapid automation with minimal configuration and low-cost deployment.

1. **IMPLEMENTATION & RESULTS**

**5.1 INTRODUCTION**

This section outlines how the system was implemented using cloud-based tools to automate threat detection and response workflows. The goal was to simulate a basic SOAR (Security Orchestration, Automation and Response) system using publicly available resources without coding or commercial software**.**

**5.2 EXPLANATION OF KEY FUNCTIONS**

* LimaCharlie was used to simulate real-time event detection on a Windows virtual machine. It acted as the EDR (Endpoint Detection and Response) sensor, forwarding events such as process creation or file modifications.
* Tines served as the automation platform, where workflows (called “Stories”) were designed to respond to specific events by filtering, enriching, or acting upon the data received.
* Slack was integrated into the workflow to receive alerts and send automated notifications in real-time.
* Burner email was used temporarily for setting up test accounts without using personal credentials.

**5.3 METHOD OF IMPLEMENTATION**

The system was implemented using the following components:

* Oracle Cloud Free Tier was used to create a free Windows virtual machine where LimaCharlie’s sensor was deployed.
* LimaCharlie was installed on the VM to start sending telemetry data to its dashboard.
* Tines workflows were configured to:
  + Receive webhook alerts from LimaCharlie.
  + Parse and filter suspicious activity.
  + Send a notification message to a Slack channel using incoming webhooks.
* Slack was set up to receive formatted alerts (e.g., “New process launched: cmd.exe”).

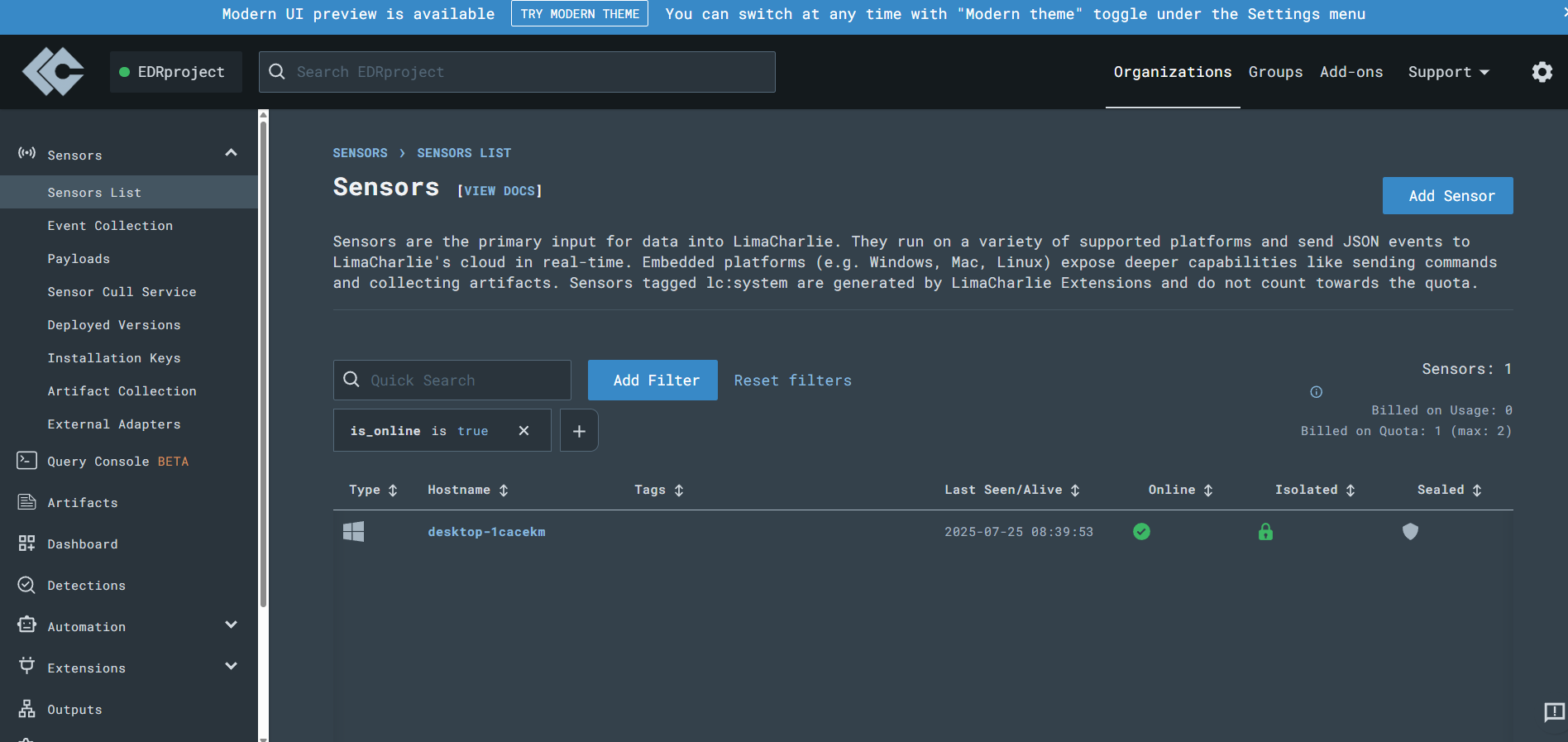
**5.3.1 FORMS**

There were no traditional “forms” in the interface, as this was not a web application. However, Tines Agents acted as forms — accepting structured data and triggering appropriate responses.

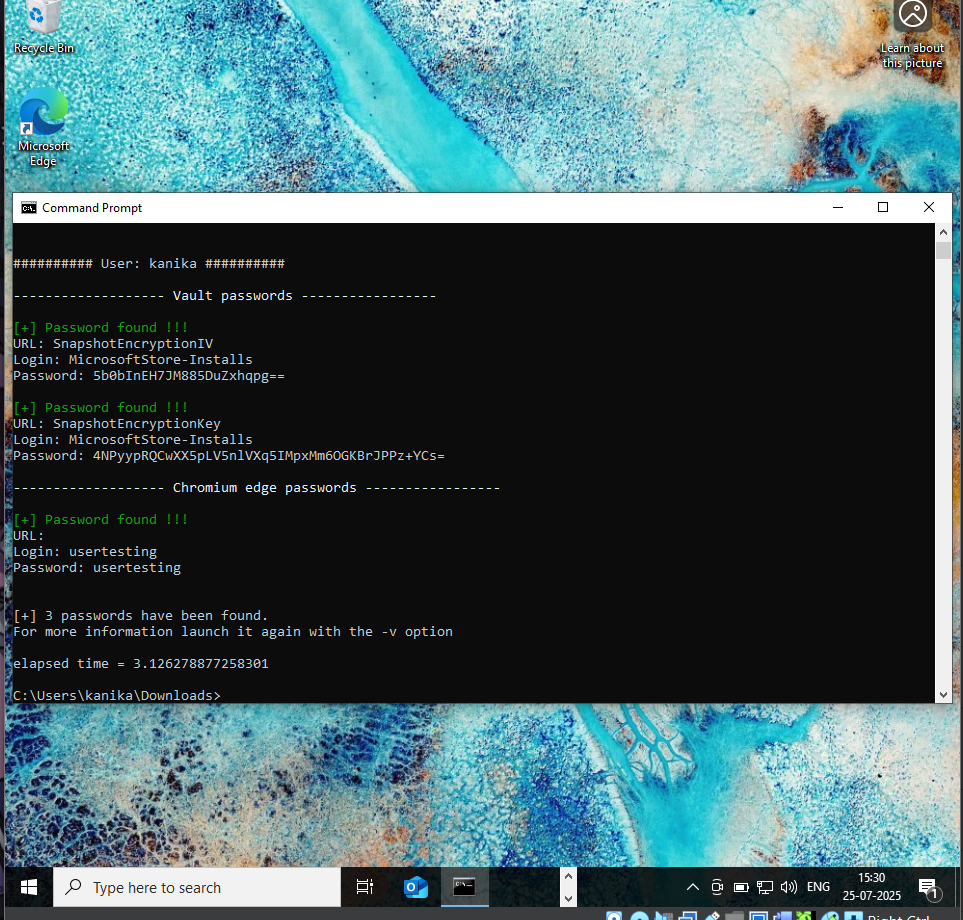
**5.3.2 OUTPUT SCREENS**

The output can be described:

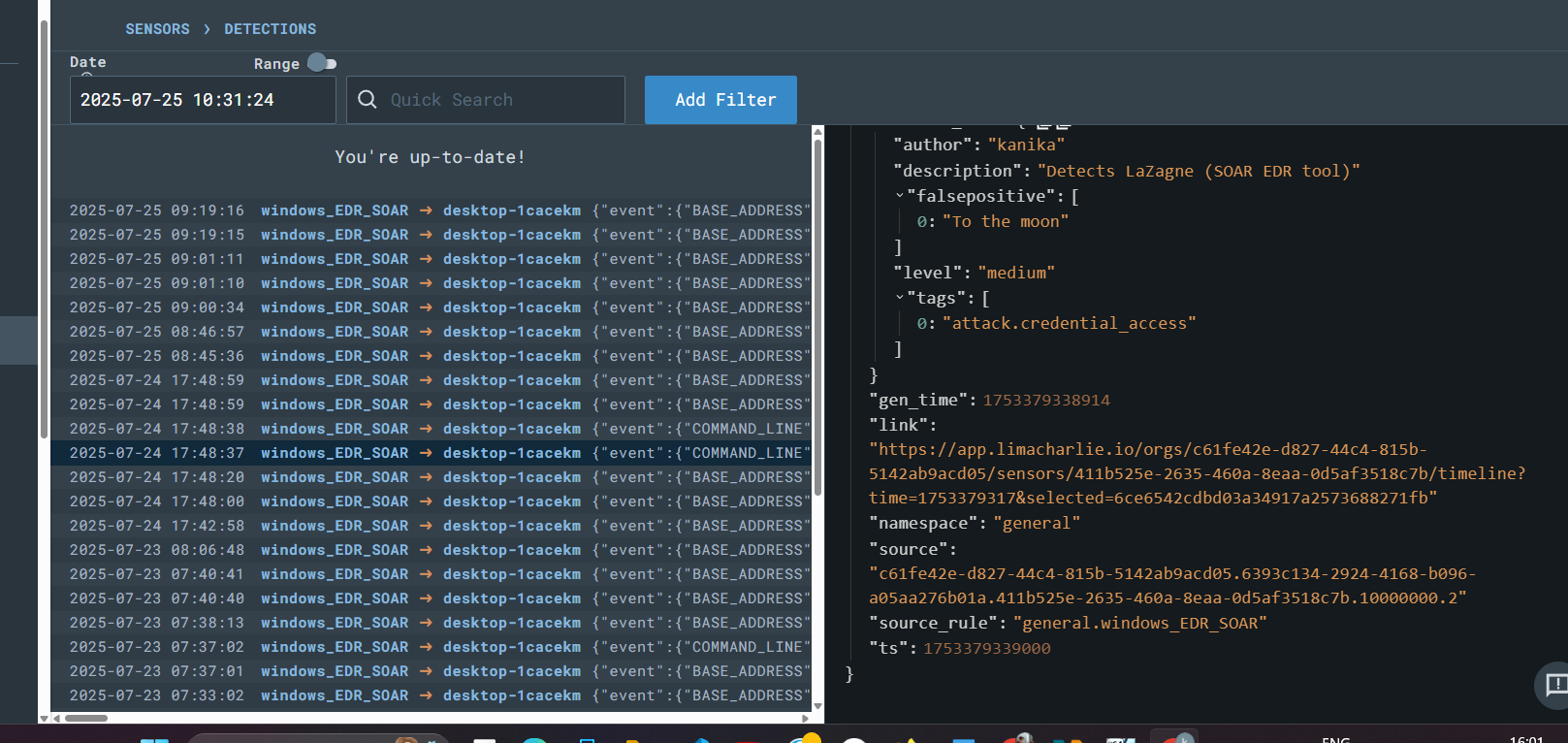
* + - * VM set up , and connecting sensor of lima-charlie.
      * Limacharlie dashboard, output screen, time-line screens
      * Tines Dashboard: Displayed execution status of each automation step.
      * Slack Channel: Displayed real-time alerts with metadata like time, process name, user, etc.



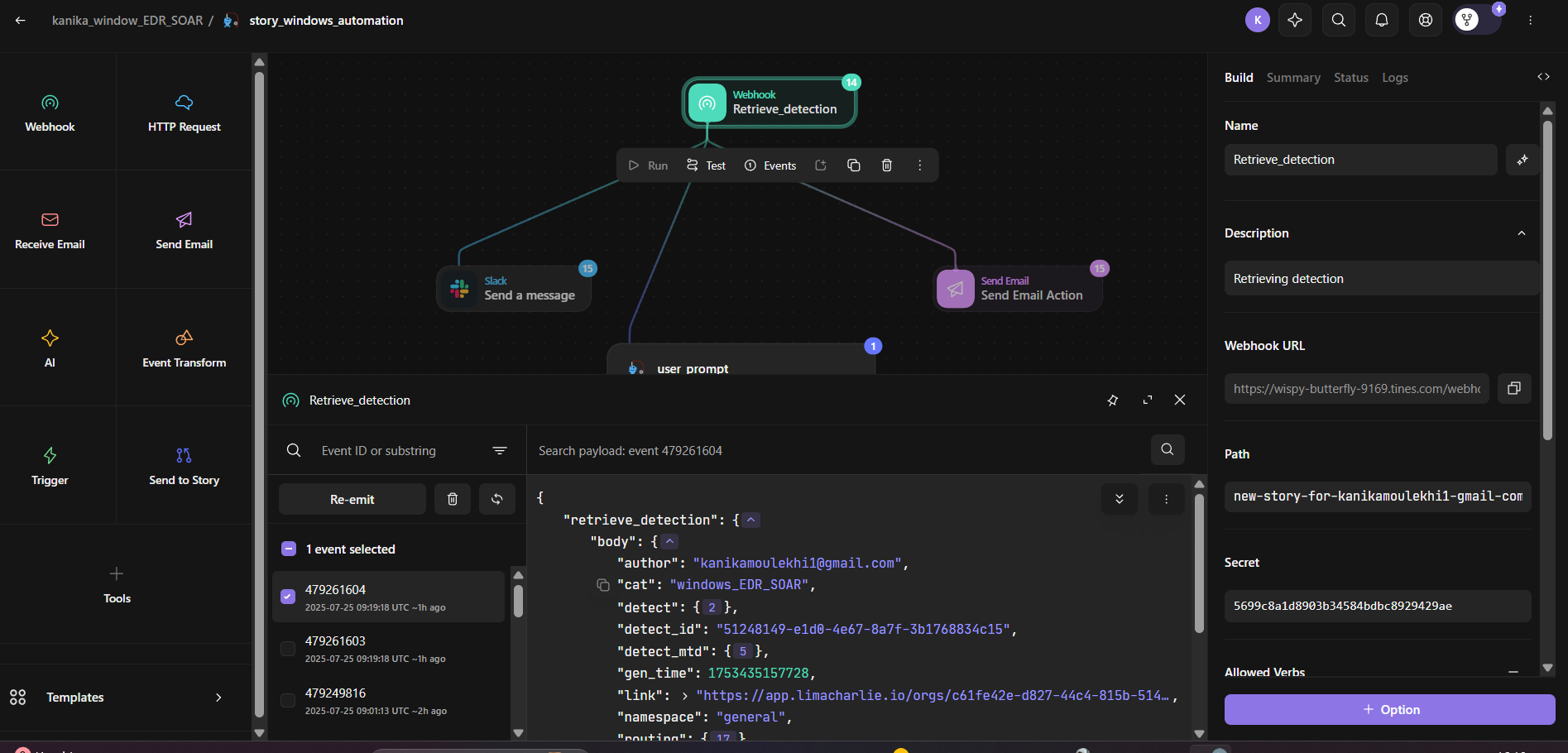
Screen shot 2.1 dashboard of limacharlie we can see its connected to the VM windows name [**desktop-1cacekm**](https://app.limacharlie.io/orgs/c61fe42e-d827-44c4-815b-5142ab9acd05/sensors/411b525e-2635-460a-8eaa-0d5af3518c7b/overview?from=sensors) as its online, can see its sidebar with all features.



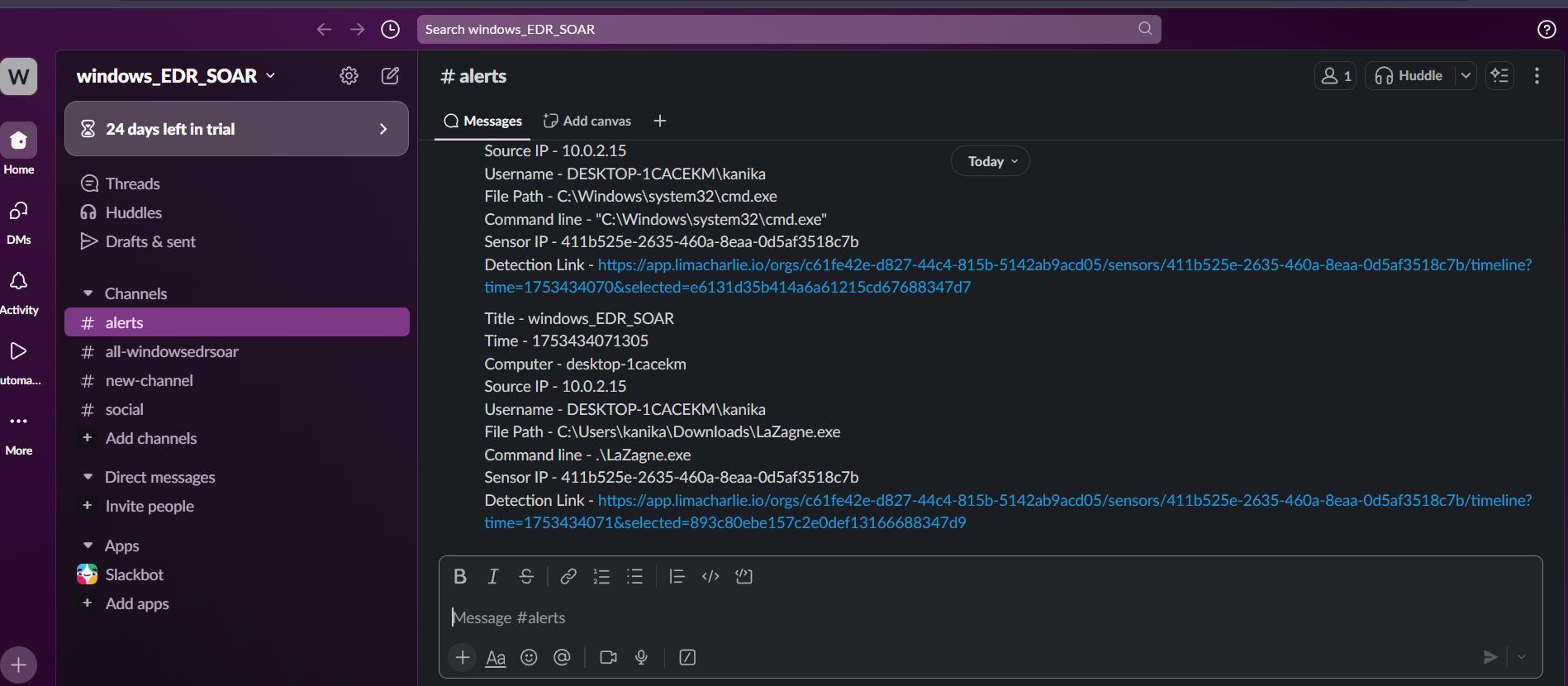
Screen shot 2.2 running LaZagne in the VM we can see it extracted some passwords.

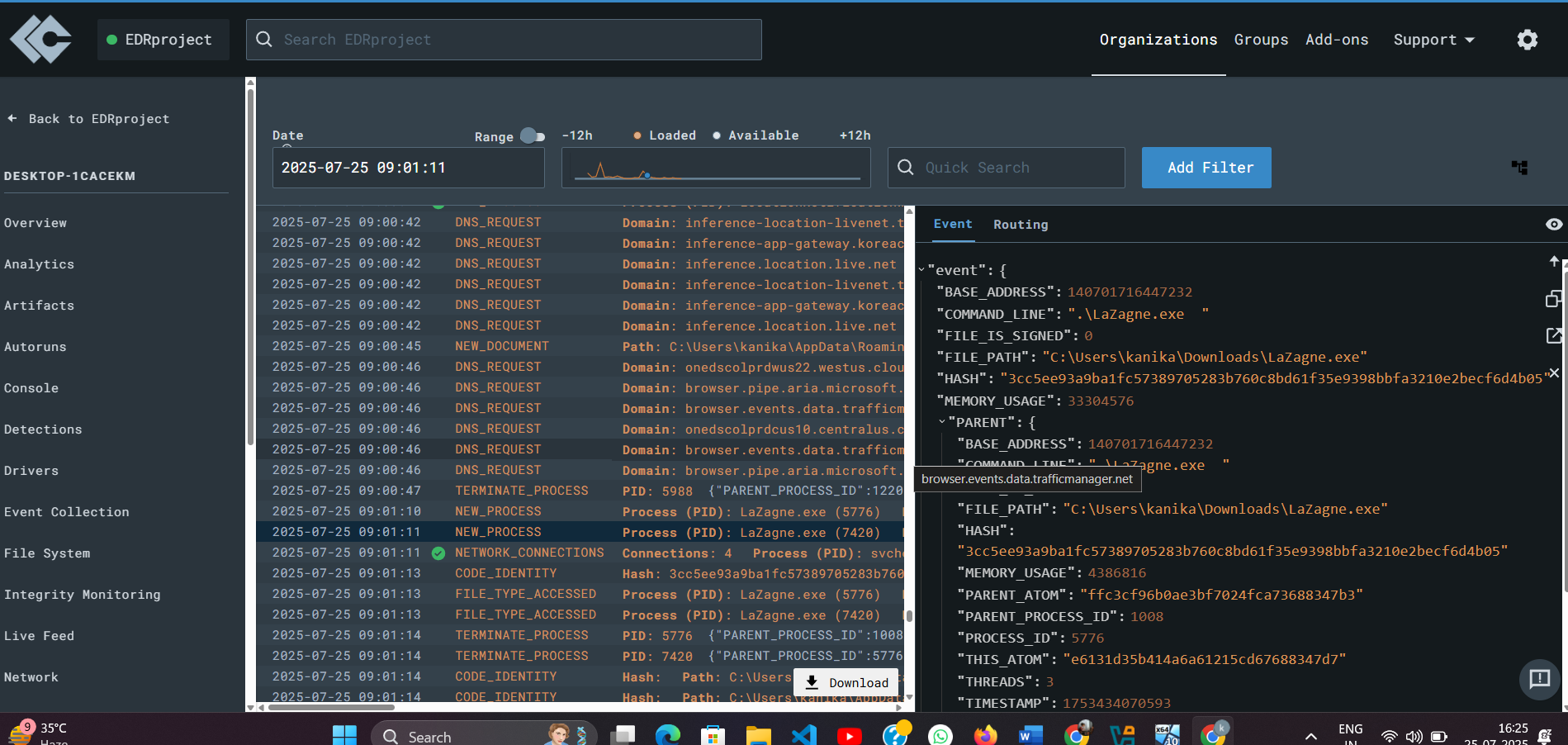


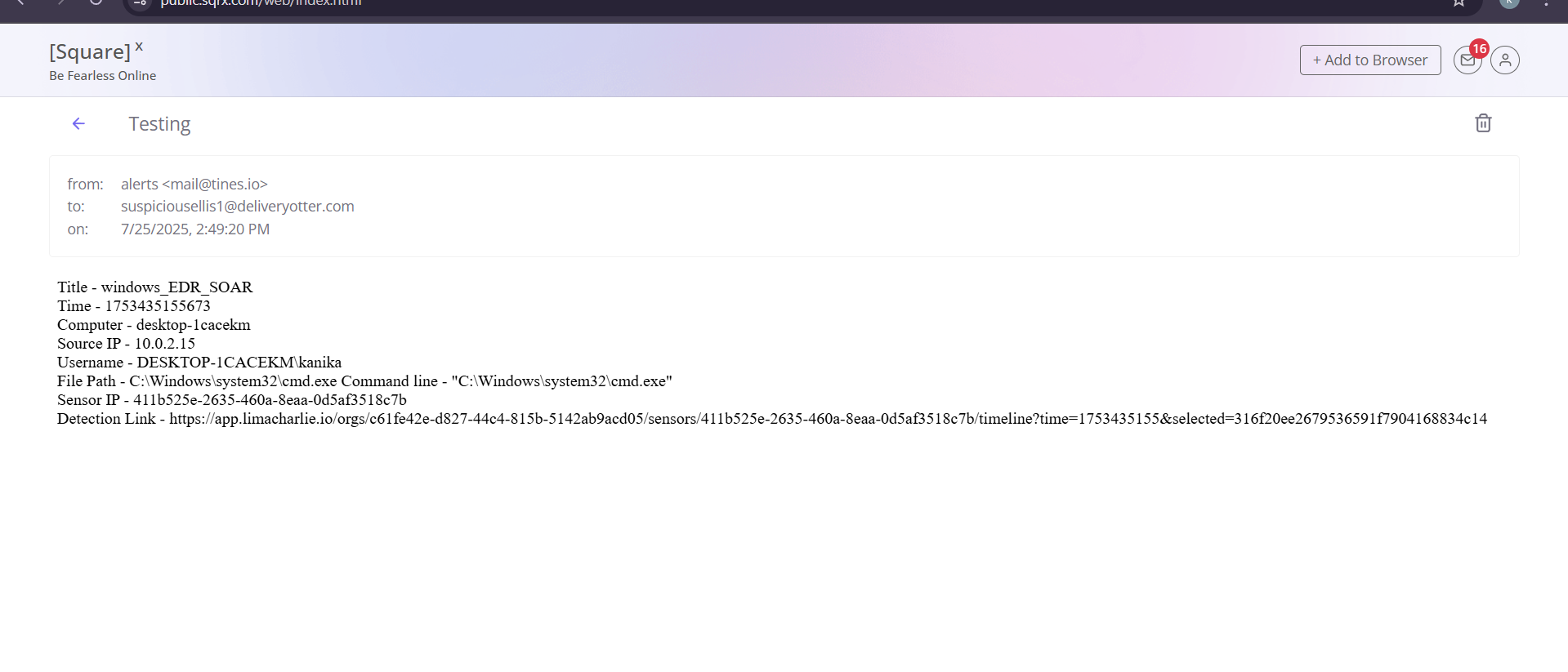
Screenshot 2.3 we can see that the detections are here and details also telling us about the threats we face.



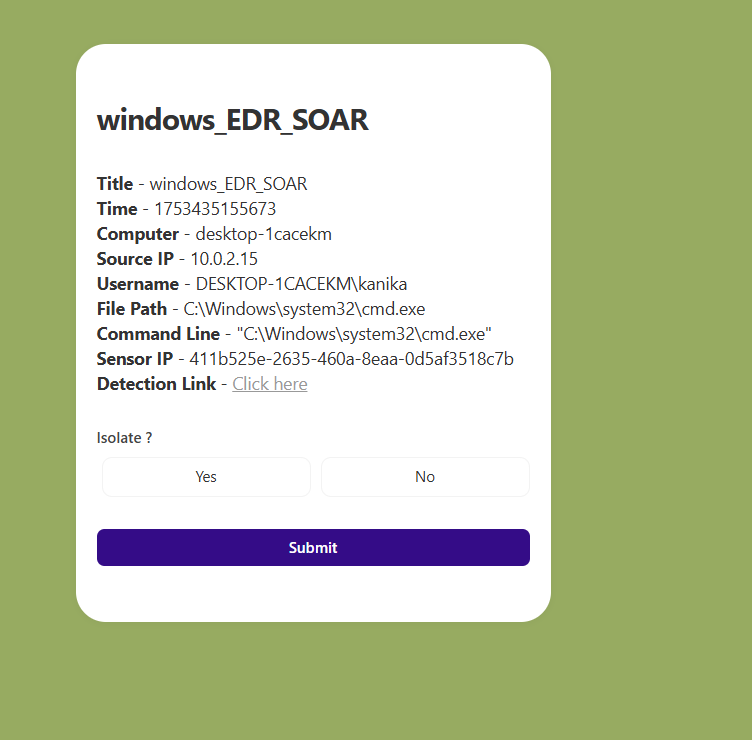
Screen shot 2.4 we can see that the detection we saw in the limacharlie are visible here in tines ‘s webhook.

****screen shot2.5 we can see that alerts are generated here giving full information about affected system we click on link and visit deeper.

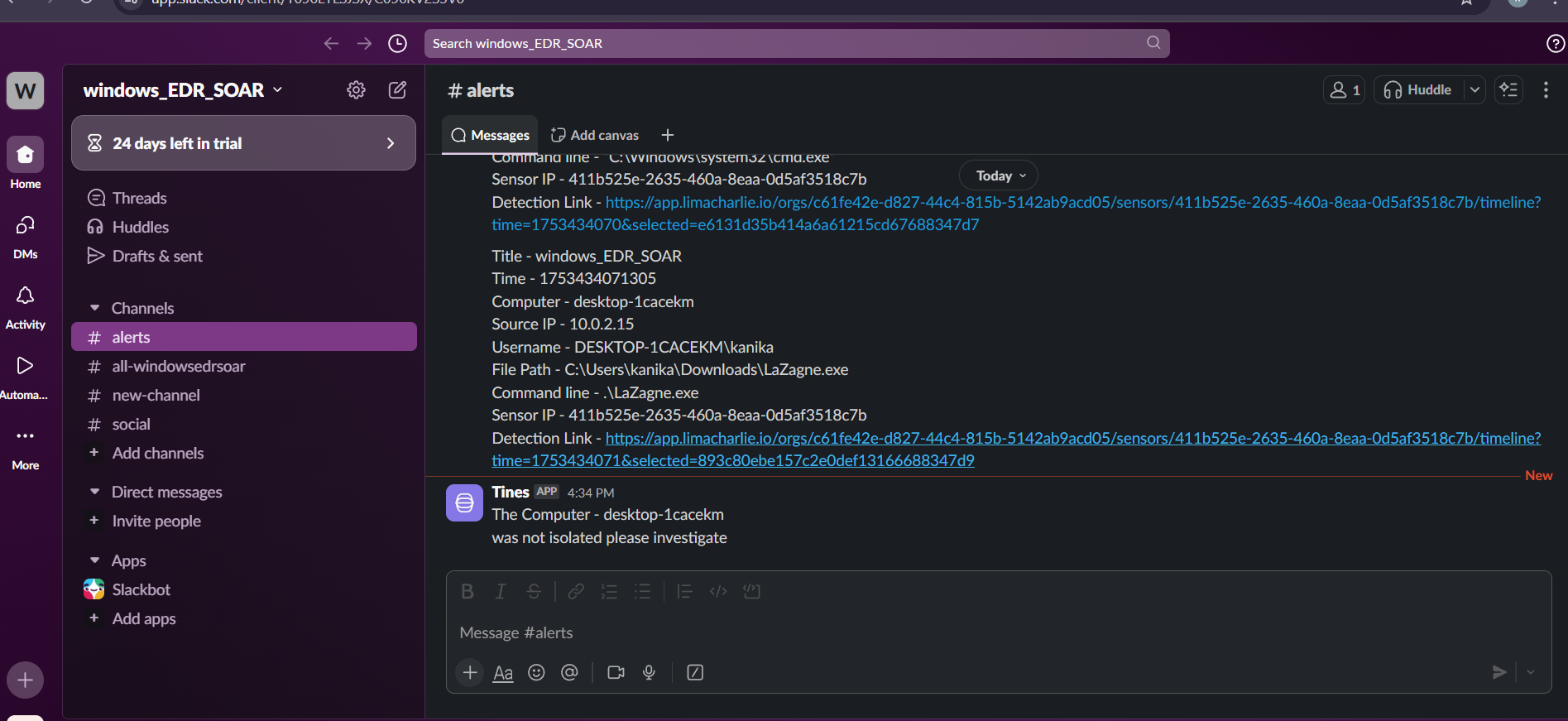
**** screen shot 2.6 after clicking on link we reach here and we can do detailed and deep analysis to decide if we should isolate machine or not?

****

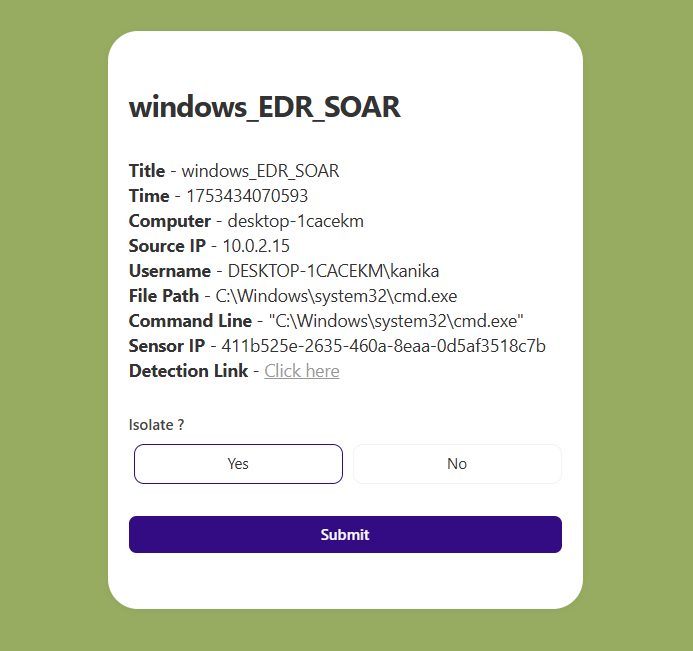
Screenshot 2.7we can a message in burner email as well and many more so we can send alerts via prebuilt message in email and in slack.



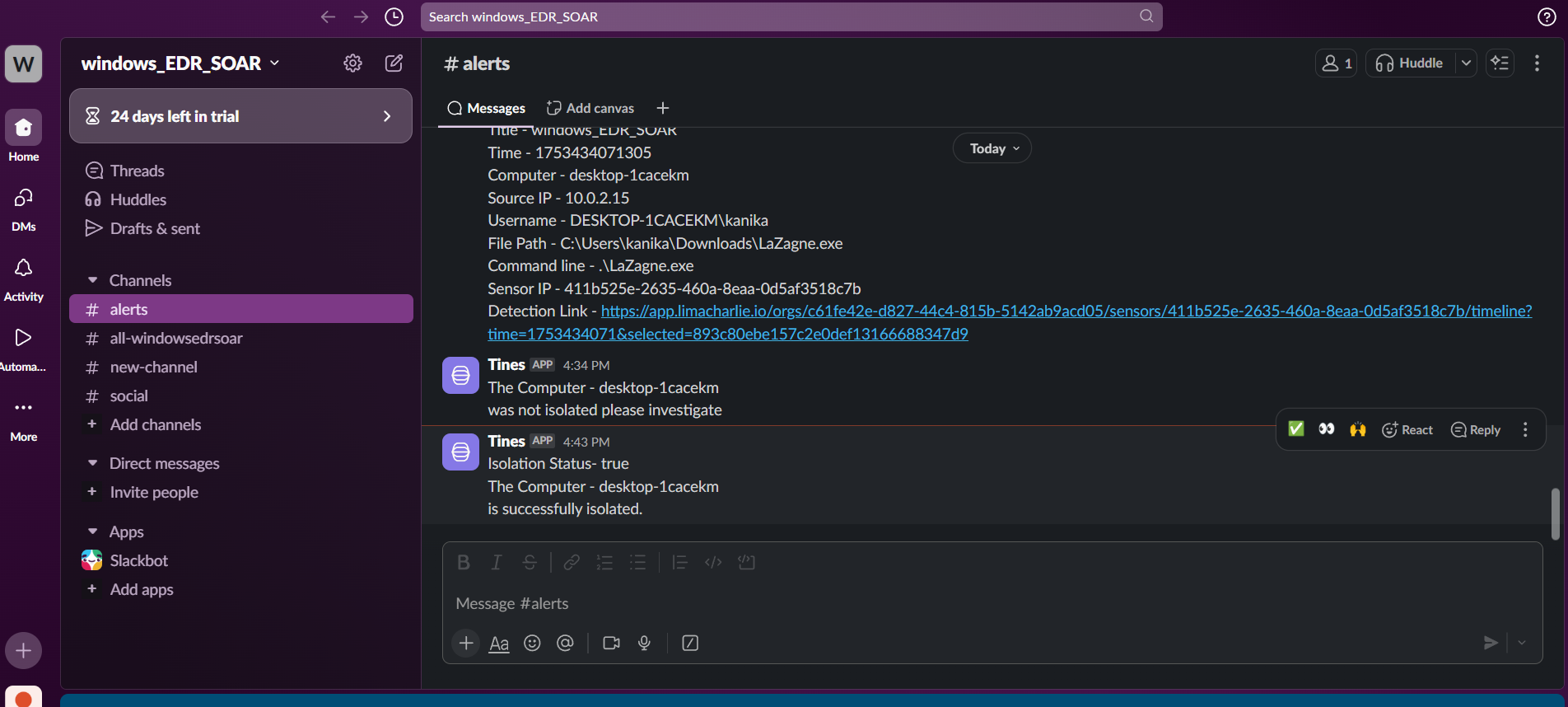
Screen shot 2.8 we now after studying the machine’s detail can decide if we want to isolate it or not, here we will choose **NO**

****

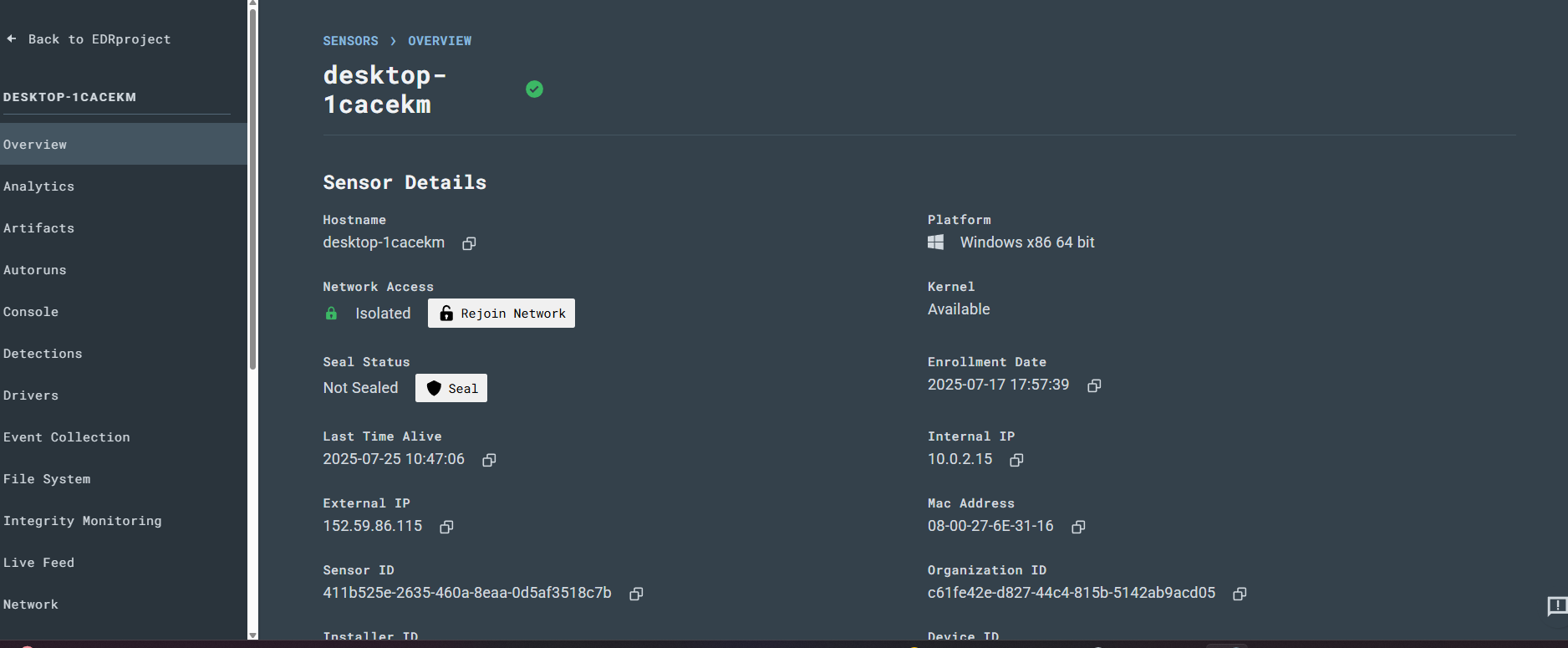
Screenshot 2.9 here we can clearly see message telling us that machine is not isolated hence, no is success.

****

Screenshot 2.10 we are going to say yes to isolate.

****

Screenshot 2.11 we can see last message here that confirm that machine is isolated.



Screen shot 2.12 here in limacharlie dashboard too we can see system is isolated.

**5.3.3 RESULT ANALYSIS**

* The system was able to detect test events like LaZagne and also able to isolate the machine.
* Alerts were delivered instantly to Slack with the correct context.
* Also the message was delivered to the set email as well.
* The solution demonstrated a low-cost, beginner-friendly SOAR workflow that worked entirely on free-tier services.

**5.4 Conclusion**

This implementation successfully proved that even without high-end enterprise tools, a functioning SOAR system can be created using free cloud platforms and open security tools. The system was able to detect endpoint activity and trigger meaningful alerts, confirming the feasibility of the design.

1. **TESTING & VALIDATION**

**6.1 INTRODUCTION**

Testing was performed to ensure that each component of the SOAR workflow, from endpoint activity to Slack alerts was working as expected. The purpose of this phase was to validate the functionality, accuracy, and responsiveness of the automated detection and response system.

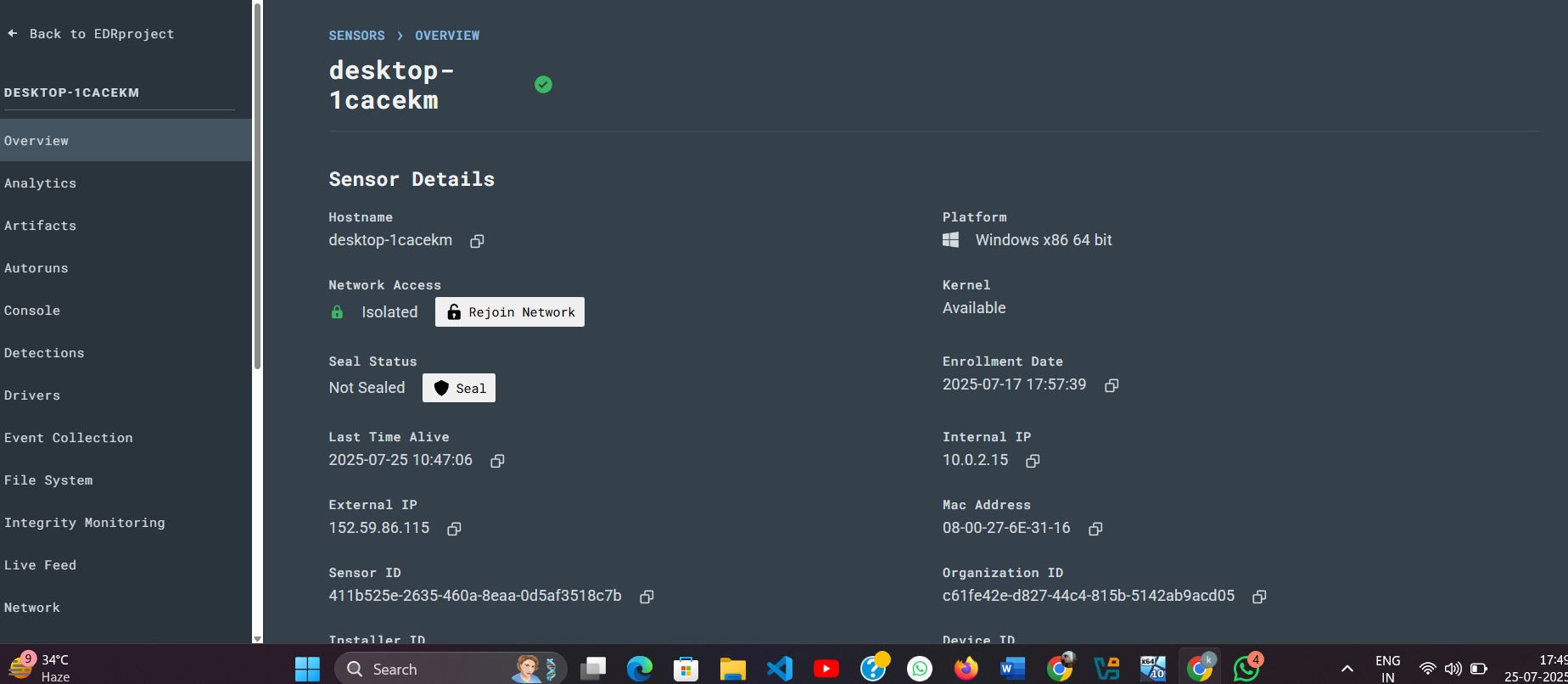
From the start by triggering fake alerts in VM to limacharlie detection and tines able to respond to it fully and can take action easily we see, the automated interconnected system work as aimed.

**6.2 DESIGN OF TEST CASES AND SCENARIOS**

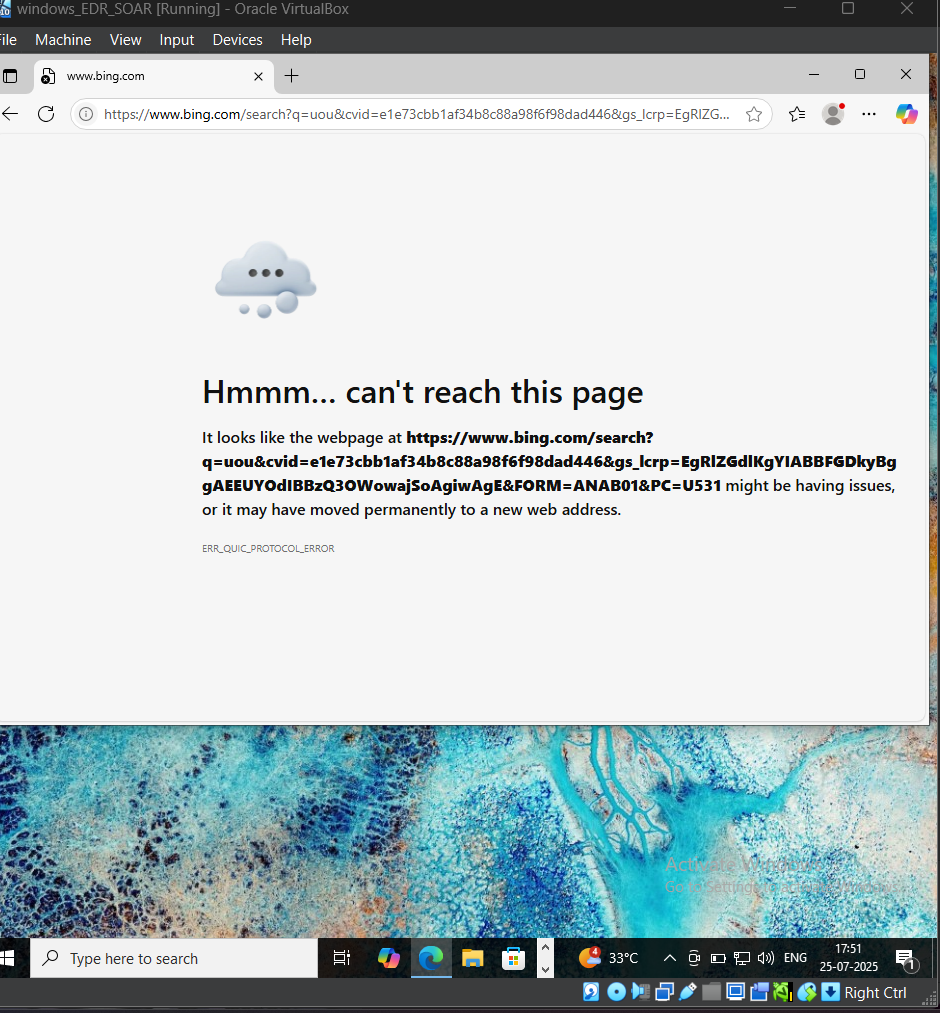
Testing focused on simulating typical endpoint activity and verifying if the SOAR system responded correctly. The test scenarios followed exactly what was demonstrated in the tutorial playlist.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Cases | Description | Expected result | status |
| TC01 | Launch LaZagne on VM by [.\LaZagne.exe] (after loading from GitHub) | It should be launched on the cmd prompt with “bang bang !!” | Checkmark with solid fill |
| TC02 | After connecting LimaCharlie to the VM machine via sensor key | We should be able to success link in LimaCharlie dashboard with name of virtual machine | Checkmark with solid fill |
| TC03 | After setting up detection and new rule to detect NEW\_PROCESSES | In the sidebar of LimaCharlie in detect component we should see a list of detections | Checkmark with solid fill |
| TC04 | Triggering fake malicious behaviour by running lazagne | It should be visible in timeline and detections and one can analyse it. | Checkmark with solid fill |
| TC05 | In the playbook after connecting to the limacharlie via credentials | In the webhook events should be visible with full details. | Checkmark with solid fill |
| TC06 | In user prompt all information is visible to analyse and decide | Choose yes to isolated, machine should be isolated(not even internet conncetion) | Checkmark with solid fill |
| TC07 | After isolated we can reconnect it to system | Going to limacharlie and to the VM, click on “rejoined” it should be rejoined. | Checkmark with solid fill |

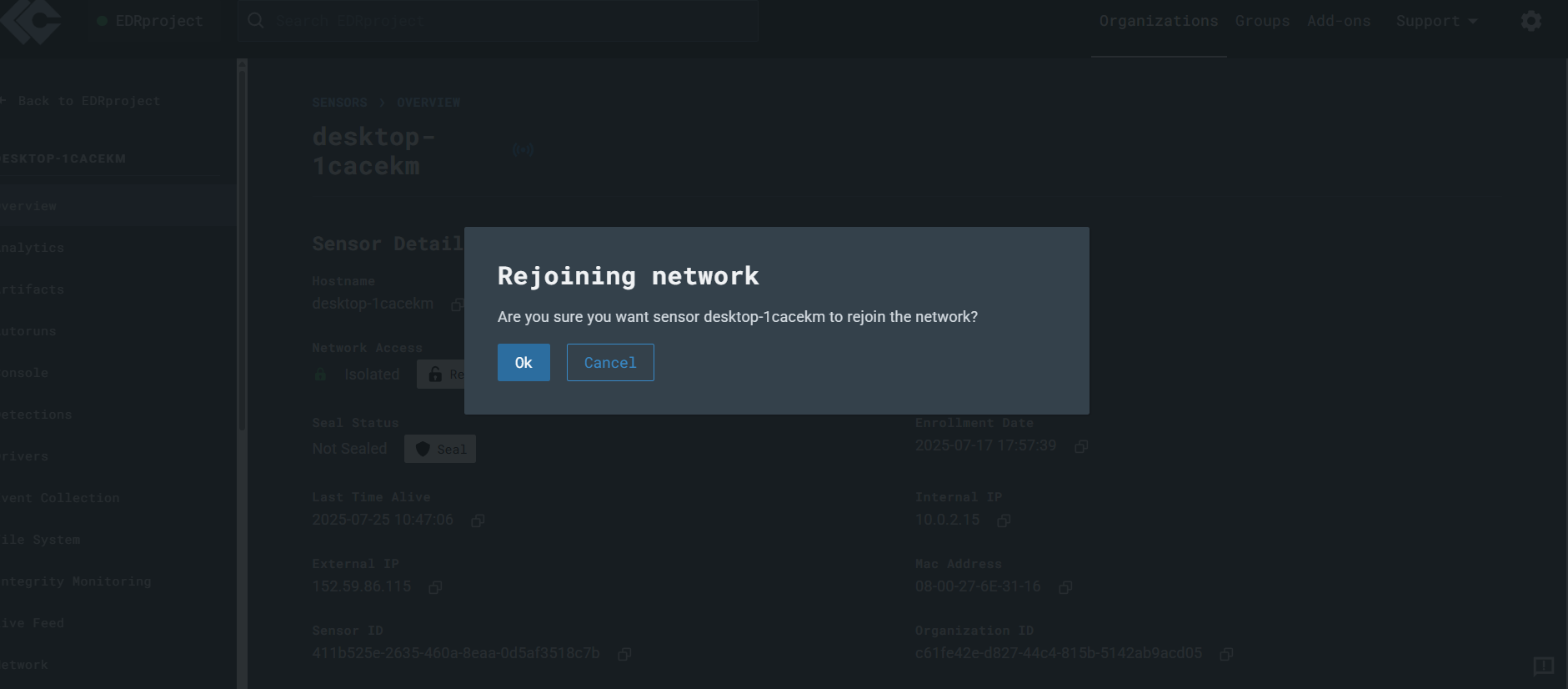
Table 3.1 testcases and their results

****

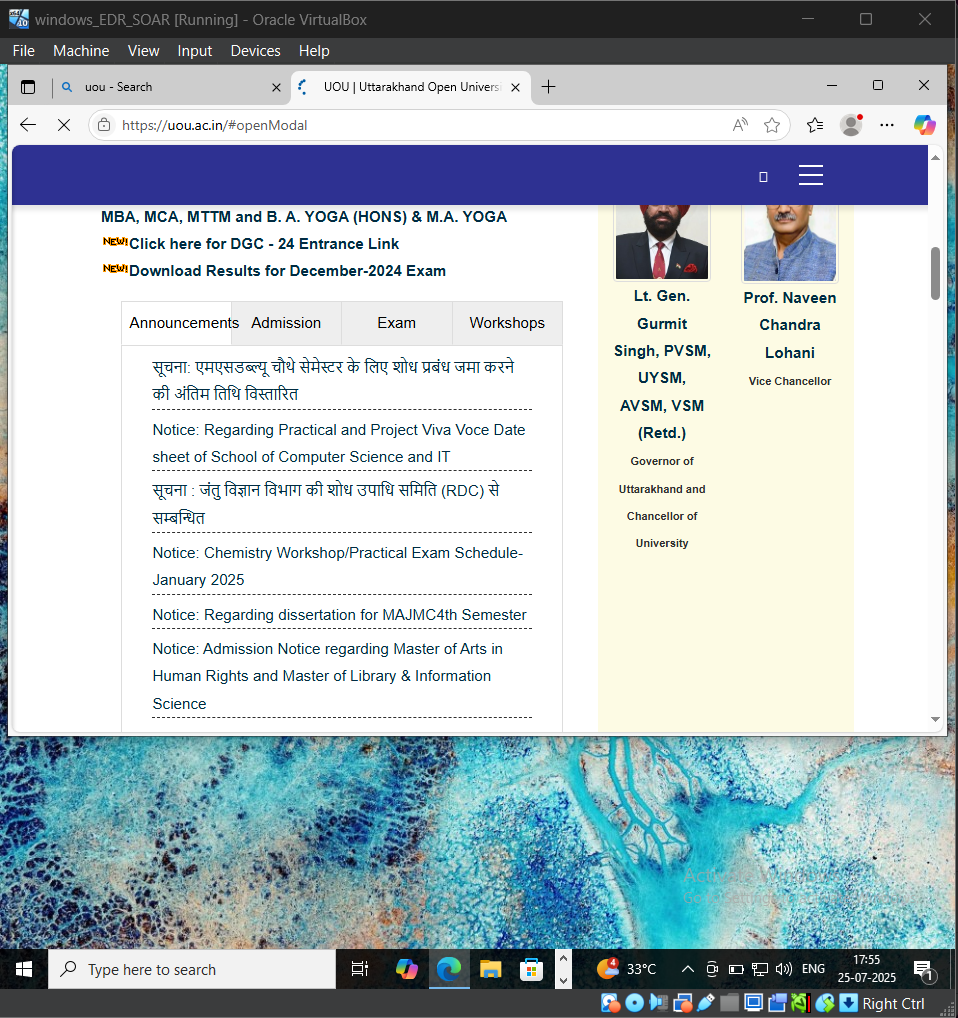
Screen shot 2.13 we see here system is isolated



Screen shot 2.14 in the VM we can clearly see internet is connected despite that it’s not working.

****

Screen shot 2.15 we can now rejoin easily by simple click.



Screen shot 2.16 now after rejoining it works fine.

**6.3 VALIDATION**

* Functional Validation: Each test input produced the expected outcome. Tine’s workflows executed exactly as designed and triggered correct Slack messages.
* Integration Validation: All components (VM → LimaCharlie → Tines → Slack) communicated successfully without delays or dropped data.
* Negative Testing: Invalid inputs or failed webhooks were properly logged and flagged for review, showing system resilience.
* Manual Cross-Check: Events were checked on the LimaCharlie dashboard to ensure the exact data was passed through the pipeline.

**6.4 CONCLUSION**

The testing phase confirmed that the system could accurately detect simulated activities and respond with automated alerts. The validation of key workflows verified that the SOAR system is functioning as intended. Test cases shows success and with minor problem that are due to not having paid membership of tines can be managed easily by pasting JWT key every few hours.

1. **CONCLUSION**

**PROJECT CONCLUSION**

This project successfully demonstrated a lightweight SOAR (Security Orchestration, Automation, and Response) system using LimaCharlie for endpoint monitoring, Tines for workflow automation, and Slack for alert delivery. The setup was deployed on a Windows VM running on Oracle Cloud Free Tier, with burner emails used for safe and temporary account creation during testing. The workflows followed in this project were directly inspired by the YouTube tutorial playlist, and they functioned exactly as intended, providing automated detection and notification with minimal configuration.

**FUTURE ENHANCEMENTS**

In future iterations, this system can be enhanced by integrating additional tools such as Sigma rule detection, The-Hive for case management, or Velociraptor for live response. Integration with threat intelligence feeds, multi-endpoint coverage, or advanced analytics within Tines could further improve the system’s capabilities. Moreover, securing the infrastructure with identity and access management controls, or containerizing the solution for scalable deployment, would increase robustness and professionalism.

Paid membership of Tines, Slack and Limacharlie can open more viable components for extra tight security and building your own filters etc.

**REFERENCES**

1. YouTube Playlist:  
   <https://www.youtube.com/playlist?list=PLMyDFIRz4TtdRg1MGvOn9y28h-Lvy5wKh>
2. LimaCharlie (Official Docs)  
   https://docs.limacharlie.io
3. Tines Automation Platform  
   https://www.tines.com
4. Oracle Cloud Free Tier  
   <https://www.oracle.com/cloud/free/>
5. Slack Incoming Webhooks  
   <https://api.slack.com/messaging/webhooks>
6. Burner email systems

https://public.sqrx.com/web