

SOAR EDR Project Report :

Overview

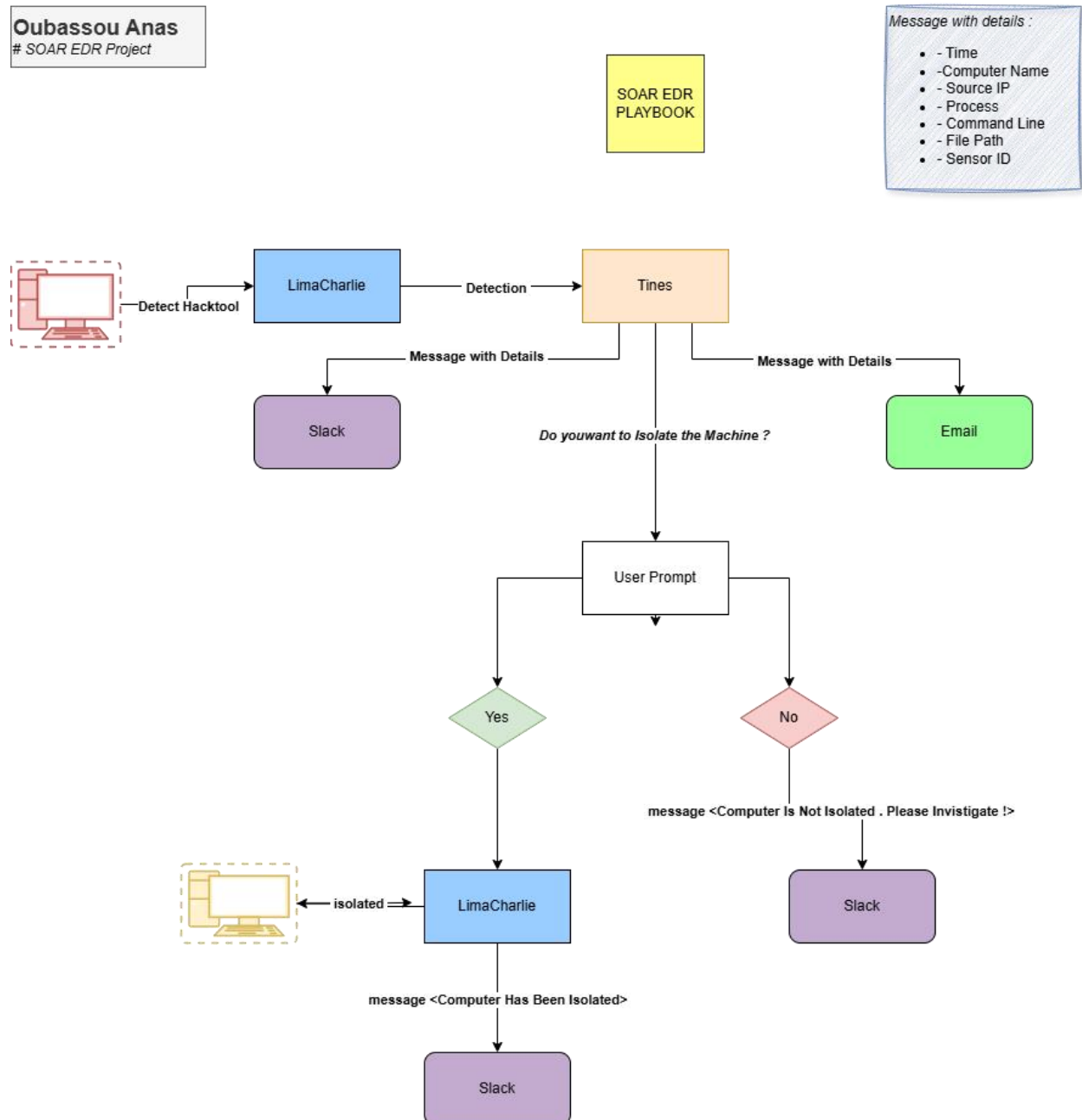
This project showcases the development of a Security Orchestration, Automation, and Response (SOAR) solution integrated with an Endpoint Detection and Response (EDR) system. The objective is to automate the detection, notification, and response process for a malicious hack tool on an infected host. The project utilizes LimaCharlie as the cloud-based EDR platform, a Windows Server 2019 instance hosted on Vultr, Tines as the SOAR platform for workflow automation, Slack for real-time communication, and Gmail for email notifications. Spanning six parts, the project covers the initial workflow design, environment setup, detection rule creation, integration of LimaCharlie and Tines with Slack and Gmail, playbook development, and validation of automated isolation using LimaCharlie's API. The solution aims to enhance incident response efficiency by combining detection capabilities with automated remediation, offering a practical framework for security operations.

Project Workflow (Part 1)

The project begins with a detailed workflow design to address the detection and response to a hack tool, specifically "lazagne," a credential-dumping tool. The workflow outlines the following steps:

1. LimaCharlie detects the execution of lazagne.exe and forwards the event to Tines.
2. Tines sends a notification to a Slack "alerts" channel and an email via Gmail, including critical details such as detection title, event time (converted from epoch to human-readable format), hostname, source IP, username, file path, command line, sensor ID, and a link to the detection in LimaCharlie.
3. A user prompt is presented within Tines asking, "Do you want to isolate the machine?" with Yes and No options, displaying the detection details for context.
4. If the user selects No, Tines sends a Slack message stating, "Computer [hostname] was not isolated, please investigate."
5. If the user selects Yes, Tines initiates an isolation request to LimaCharlie via its API, isolates the machine, retrieves the isolation status, and sends a Slack message confirming, "Computer [hostname] has been isolated, isolation status: [true/false]."

This workflow serves as the blueprint for the subsequent setup and automation phases.



Environment Setup (Part 2)

The technical foundation was established by setting up a Windows Server 2019 instance on Vultr. LimaCharlie was configured by creating a new organization, generating an installation key, and deploying a Windows sensor to the server. The installation process involved downloading the LimaCharlie agent installer, executing it with the installation key, and verifying the sensor's online status in the LimaCharlie dashboard. A free Slack workspace was created, and an "alerts" channel was set up to receive notifications. A Gmail account was configured for email alerts. Additionally, a Tines account was established, and a new story was created to house the automation logic. The environment was tested by ensuring the LimaCharlie sensor was active, the Slack channel was accessible, and Gmail was ready to send emails, laying the groundwork for integration and automation.

The screenshot shows the LimaCharlie dashboard with the 'Sensors' page selected. The left sidebar contains navigation links: Sensors, Sensors List, Event Collection, Payloads, Sensor Cull, Deployed Versions, Installation Keys (highlighted), Artifact Collection, Query Console BETA, Artifacts, Dashboard, Detections, Automation, and Extensions. The main content area is titled 'Sensors [View Docs]' and includes a '+ Add Sensor' button. Below this is a 'Quick Search' bar and a '+ Add Filter' button. A filter is applied: 'online is true'. The sensor list table shows the following data:

Hostname	Tags	Last Seen/Alive	Online	Isolated	Sealed
ext-atomic-red-team	ext:ext-atomic-red-team; lc:system	2024-05-03 20:16:49	✓	🔒	🛡️
ext-reliable-tasking	ext:reliable-tasking; lc:system	2024-05-03 20:16:56	✓	🔒	🛡️
demo1	lc-demo-sensor	2024-05-03 20:17:01	✓	🔒	🛡️
IT-01.corp.net		2024-05-03 20:17:14	✓	🔒	🛡️
ext-yara	ext:ext-yara; lc:system	2024-05-03 20:17:12	✓	🔒	🛡️

Summary statistics: Sensors: 5, Billed on Usage: 5, Billed on Quota: 0 (max: 2).

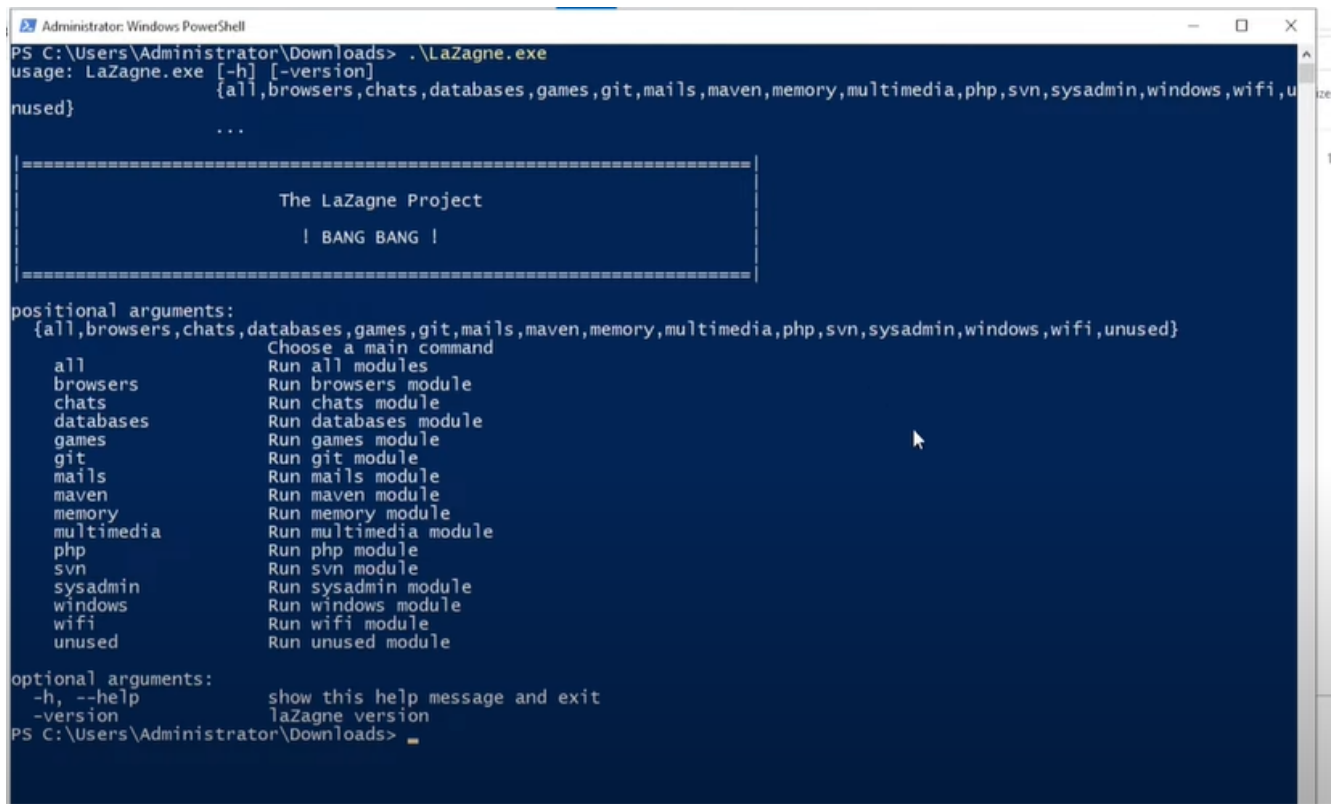
The screenshot shows the 'Installation Keys' page in the LimaCharlie dashboard. The left sidebar is the same as the previous screenshot, with 'Installation Keys' highlighted. The main content area is titled 'Installation Keys [View Docs]' and includes a 'Create Installation Key' button. Below this is a table with one entry:

Description	Tags	Sensor Key	Chrome Key	Adapter Key	Actions
MyDFIR-SOAR-EDR-Project					

Below the table, there are two sections: 'Installation [View Docs]' and 'Connectivity'. The 'Installation' section provides instructions for installing the sensor on various operating systems (Windows, macOS, Linux, Chrome OS, Docker) and mentions a sample installer script. The 'Connectivity' section provides information about agent-to-cloud and Chrome agent-to-cloud connections, including required ports and SSL certificates.

Detection Rule Creation (Part 3)

A detection rule was developed in LimaCharlie to identify the "lazagne" hack tool. The rule was crafted using a JSON-based configuration, targeting events where a process named "lazagne.exe" is executed. Key conditions included matching the file path (e.g., "C:\Users\Administrator\Downloads\lazagne.exe"), command line arguments, and event type (e.g., NEW_PROCESS). The rule was tested by downloading and running lazagne.exe on the Windows Server, triggering a detection event logged in LimaCharlie. The event payload included metadata such as base address, command line, file signature status, hash, memory usage, parent process ID, process ID, thread count, username, and routing information. This successful detection confirmed LimaCharlie's capability to identify the target threat, serving as the trigger for the Tines automation.



```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Downloads> .\LaZagne.exe
usage: LaZagne.exe [-h] [-version]
                  {all,browsers,chats,databases,games,git,mails,maven,memory,multimedia,php,svn,sysadmin,windows,wifi,unused}
...

=====
                        The LaZagne Project
                        ! BANG BANG !
=====

positional arguments:
  {all,browsers,chats,databases,games,git,mails,maven,memory,multimedia,php,svn,sysadmin,windows,wifi,unused}
    all                  Choose a main command
    browsers             Run all modules
    chats                Run browsers module
    databases            Run chats module
    games                Run databases module
    git                  Run games module
    mails                Run git module
    maven                Run mails module
    memory               Run maven module
    multimedia           Run memory module
    php                  Run multimedia module
    svn                  Run php module
    sysadmin             Run svn module
    windows              Run sysadmin module
    wifi                 Run windows module
    unused               Run wifi module

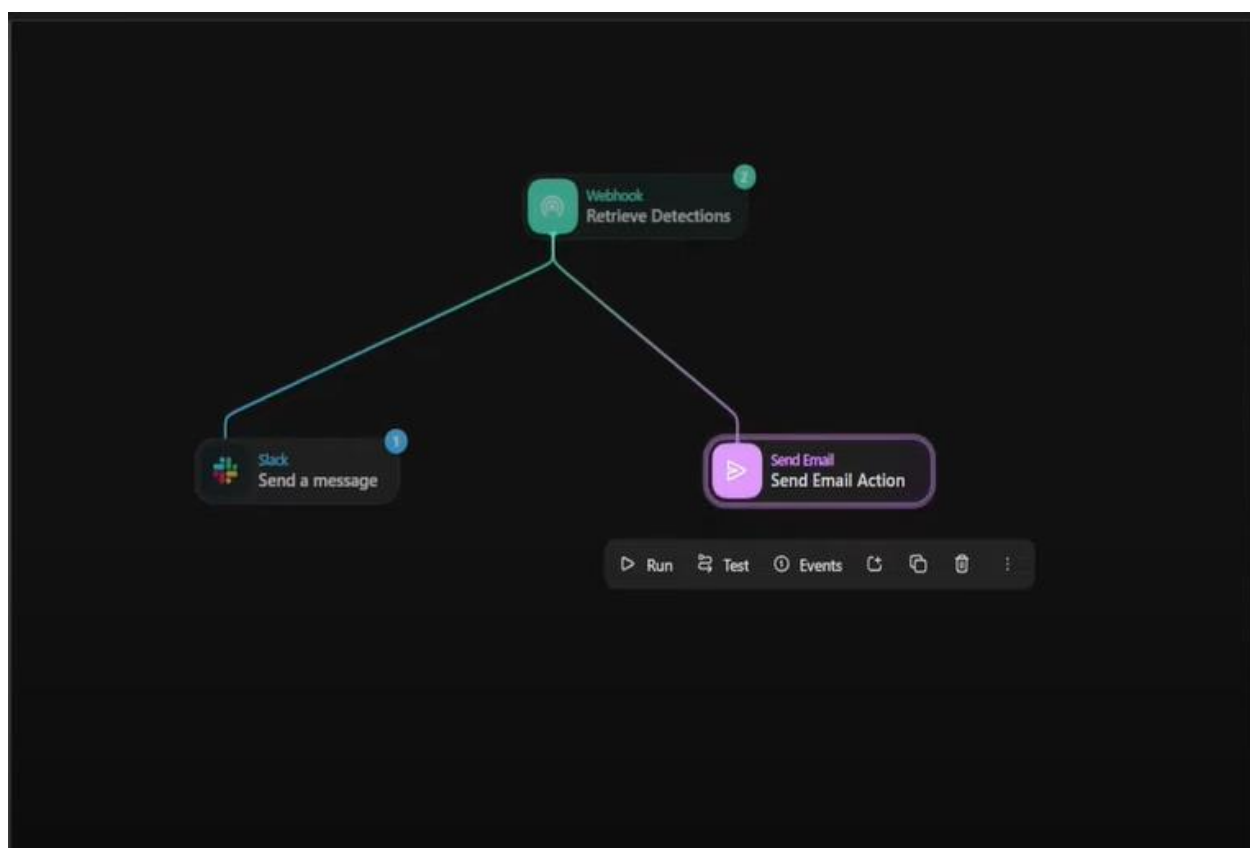
optional arguments:
  -h, --help            show this help message and exit
  -version               lazagne version
PS C:\Users\Administrator\Downloads>
```

```
1 events:
2 - NEW_PROCESS
3 - EXISTING_PROCESS
4 op: and
5 rules:
6 - op: is windows
7 - op: or
8   rules:
9   - case sensitive: false
10     op: ends with
11     path: event/FILE_PATH
12     value: lazagne.exe
13   - case sensitive: false
14     op: ends with
15     path: event/COMMAND_LINE
16     value: all
17   - case sensitive: false
18     op: contains
19     path: event/COMMAND_LINE
20     value: lazagne
21   - case sensitive: false
22     op: is
23     path: event/HASH
24   value: '3cc5ee93a9ba1fc57389705283b760c8bd61f35e9398bbfa3210e2becf6d4b05'
```

```
Respond
1 - action: report
2 metadata:
3   author: MyDFIR
4   description: Detects Lazagne (SOAR-EDR Tool)
5   falsepositives:
6     - To the moon
```

Integration Setup (Part 4)

Tines was integrated with LimaCharlie using a webhook to receive detection events in real-time. A webhook URL was configured in LimaCharlie, and Tines was set up to listen for incoming events. A Slack app was created and integrated into Tines by adding the app credentials (client ID, client secret, and verification token) and linking it to the "alerts" channel. The Gmail integration was set up in Tines using the Gmail account credentials to send email notifications. The integration was tested by regenerating the lazagne detection on the Windows Server, verifying that Tines received the event payload from LimaCharlie and successfully posted a message to the Slack channel and an email via Gmail. Both communication channels were operational.



Search payload: event 19513533

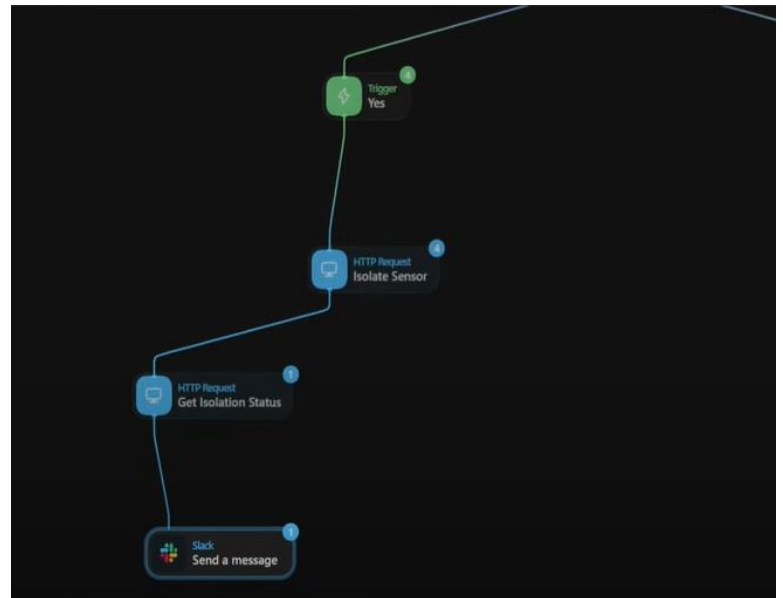
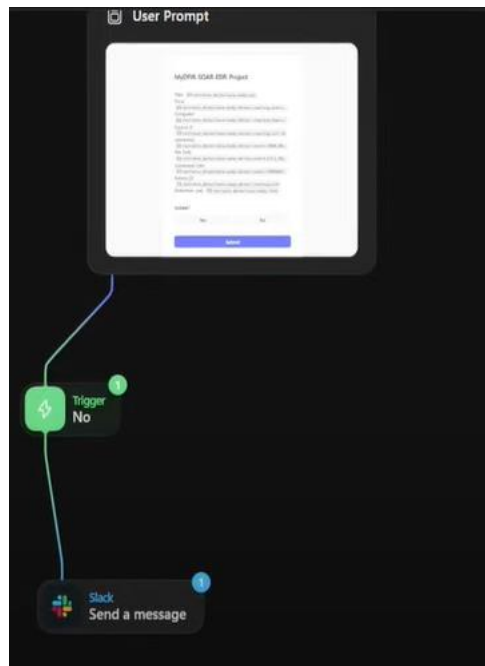
```
{
  "event": {
    "BASE_ADDRESS": 140700236972032,
    "COMMAND_LINE": > "\"C:\\Users\\Administrator\\Downloads\\LaZagne.exe\"",
    "FILE_IS_SIGNED": 0,
    "FILE_PATH": "C:\\Users\\Administrator\\Downloads\\LaZagne.exe",
    "HASH": > "3cc5ee93a9ba1fc57389705283b760c8bd61f35e9398bbfa32...",
    "MEMORY_USAGE": 15106048,
    "PARENT": > { ... },
    "PARENT_PROCESS_ID": 3340,
    "PROCESS_ID": 5676,
    "THREADS": 3,
    "USER_NAME": "MYDFIR-SOAR-EDR\\Administrator"
  },
  "routing": > { ... }
},
"detect_id": "1726b54b-1252-468f-9f98-3fb366356282",
"detect_mtd": > { ... },
```

Playbook Creation (Part 5)

A detailed playbook was constructed in Tines to automate the workflow. The playbook consisted of the following components:

- **Slack and Email Notifications:** An event transform action in Tines parsed the detection data from LimaCharlie, extracting fields like title, time, hostname, IP, username, file path, command line, sensor ID, and detection link. A Slack action posted this information to the "alerts" channel using the channel ID, while an email action sent the details to the Gmail address with HTML formatting for clarity.
- **User Prompt:** A user prompt page was added in Tines, displaying the detection details and asking, "Do you want to isolate the machine?" with Yes and No options. The page was styled for readability and included a submit button to record the user's decision.
- **Conditional Logic:** Triggers were implemented in Tines to handle the Yes/No responses. A No trigger sent a Slack message, "Computer [hostname] was not isolated, please investigate," while a Yes trigger proceeded to the isolation phase. The playbook was tested by triggering the lazagne detection, confirming that notifications were sent to Slack and Gmail and the prompt appeared as expected.

This phase solidified the automation logic, preparing it for the final response step.

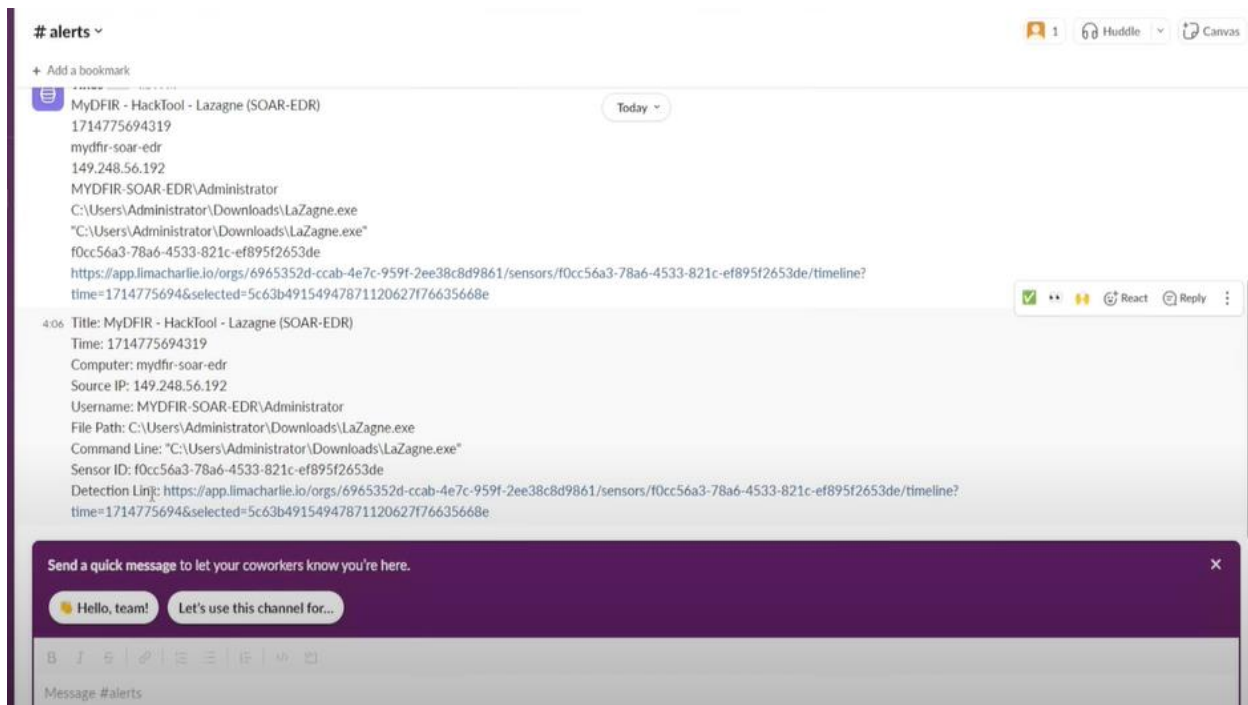


Automated Response and Validation (Part 6)

The playbook was extended in Tines to include automated machine isolation and validation using LimaCharlie's API. The process involved:

- **Isolation:** Upon a Yes response, Tines used LimaCharlie's API to isolate the machine. An organization API key was added to Tines to authenticate HTTP requests. The sensor ID from the detection payload identified the target machine, and an HTTP request was sent to the LimaCharlie API endpoint to initiate isolation.
- **Status Confirmation:** A second HTTP request in Tines retrieved the isolation status, which was then included in a Slack message, "Computer [hostname] has been isolated, isolation status: [true/false]." The status was verified to ensure the isolation command was executed successfully.
- **Validation:** The isolation was validated by pinging an external site (e.g., youtube.com) from the Windows Server on Vultr. Pre-isolation pings succeeded, while post-isolation pings failed, confirming network isolation. The LimaCharlie sensor timeline was checked to confirm the isolated status, and the Slack message was reviewed to ensure accurate reporting.

This final step demonstrated the end-to-end automation, from detection to remediation, with successful validation of the isolation process.



Outcomes

The project achieved its objectives with a fully functional SOAR-EDR integration:

- LimaCharlie accurately detected the lazagne hack tool and triggered the Tines playbook.
- Notifications were delivered via Slack and Gmail, containing all specified details, enhancing visibility for the security team.
- The user prompt in Tines provided a manual decision point, ensuring human oversight before isolation.
- Automated isolation was executed seamlessly via LimaCharlie's API, with status confirmation sent to Slack.
- Network connectivity tests post-isolation validated the effectiveness of the response, with the LimaCharlie sensor timeline corroborating the isolated state.

This implementation proves the viability of automating incident response, reducing manual effort and response time.

Conclusion

This SOAR-EDR project delivers a robust framework for detecting and responding to security incidents. By integrating LimaCharlie, Tines, Slack, Gmail, Windows Server on Vultr, it automates the identification of threats like lazagne, notifies the team, and isolates affected machines efficiently. The step-by-step process, from workflow design to validation, provides a comprehensive learning experience, applicable to real-world security operations. The solution's success highlights the potential of automation in enhancing cybersecurity resilience.