# SOC Project: SOAR Implementation Using Tines and LimaCharlie

**Introduction:**

In today's rapidly evolving cybersecurity landscape, **Security Orchestration, Automation, and Response (SOAR)** solutions are becoming essential components of Security Operations Centers (SOCs). The primary goal of SOAR is to automate repetitive tasks, enabling SOC teams to focus on more complex incidents, streamline workflows, and enhance incident response through structured playbooks.

This report details the process of integrating **Tines** as the SOAR platform with **LimaCharlie** as an Endpoint Detection and Response (EDR) system. The project demonstrates how SOAR can be effectively used in a SOC environment to automate responses to detections, such as password recovery tool activity, and escalate actions based on user input. This project also emphasizes the importance of understanding the functionality of tools, irrespective of the specific vendor, by focusing on core SOAR concepts rather than proprietary features.

**Project Objectives:**

1. **Automate Detection and Response**: Automate the detection of password recovery tools on Windows machines using LimaCharlie and respond to incidents using Tines.
2. **Design and Implement Playbooks**: Create a custom playbook (referred to as "story" in Tines) that integrates alerts with communication tools such as Slack and email for incident notification and escalation.
3. **User Interaction in Incident Response**: Provide a mechanism for user input to determine further actions, such as isolating a machine upon detecting unauthorized activity.
4. **Develop Hands-On SOAR and EDR Experience**: Gain practical experience with SOAR and EDR platforms by setting up detection rules, generating events, and automating responses.

**Project Workflow Overview:**

1. **Preparation**:
    - Install **Tines** as the SOAR platform and **LimaCharlie** as the EDR solution.
    - Deploy a Windows Virtual Machine (VM) to act as the test environment.
    - Ensure the VM has an active internet connection to communicate with the SOAR and EDR platforms.
2. **Detection Setup**:

    - **Custom Detection Rules in LimaCharlie**: Implement a detection rule to identify the usage of password recovery tools on the Windows VM.
    - **Data Collection**: Start collecting event data from LimaCharlie, which will act as the EDR solution for the project.

3. **Playbook Creation (SOAR Story in Tines)**:
    - Design the **playbook** (story) in Tines, which will automate the response to detections.
    - The playbook will:
        - Send an alert via **email** and **Slack** when a detection is made.
        - Query the user to ask whether the detected machine should be isolated.
        - **Isolate the machine** using LimaCharlie's capabilities if the user confirms the action.
4. **Slack and Email Integration**:
    - Integrate **Slack** and **email** with the playbook to enable real-time incident notifications and escalations.
    - Test the integration to ensure that notifications are triggered upon detection.
5. **User Interaction**:
    - The playbook will prompt the user with a **confirmation** to isolate the machine if suspicious activity is detected. Based on the user's response, LimaCharlie will execute an automated isolation process.

**Challenges Encountered:**

1. **Playbook Design Iterations**: The creation of a functional playbook (story) required several iterations. As is common in SOC environments, some planned functions didn't work as expected, and adjustments were made during the development process.
2. **Integration and Event Handling**: Ensuring smooth integration between LimaCharlie and Tines involved troubleshooting compatibility issues, particularly in passing alerts and data between the platforms. Documenting these integration steps was key to resolving problems efficiently.
3. **Handling User Inputs**: Allowing user interaction in automated processes posed some challenges. The logic required to handle user responses within the playbook was adjusted to ensure accuracy, as incorrectly configured logic could lead to improper incident response actions.

**Results:**

By the end of the project, we have successfully:

- Created a **detection rule** in LimaCharlie to monitor and alert on the use of password recovery tools.
- Integrated **Slack** and **email** notifications into the SOAR playbook, enabling real-time communication of incidents to the SOC team.
- Developed a **fully functional playbook** that provided users with options for manual intervention in incident handling, such as isolating compromised machines.

The project showcased the value of SOAR platforms like Tines in automating SOC workflows and allowed for hands-on experience in managing and responding to security threats in real-time.

**Lessons Learned:**

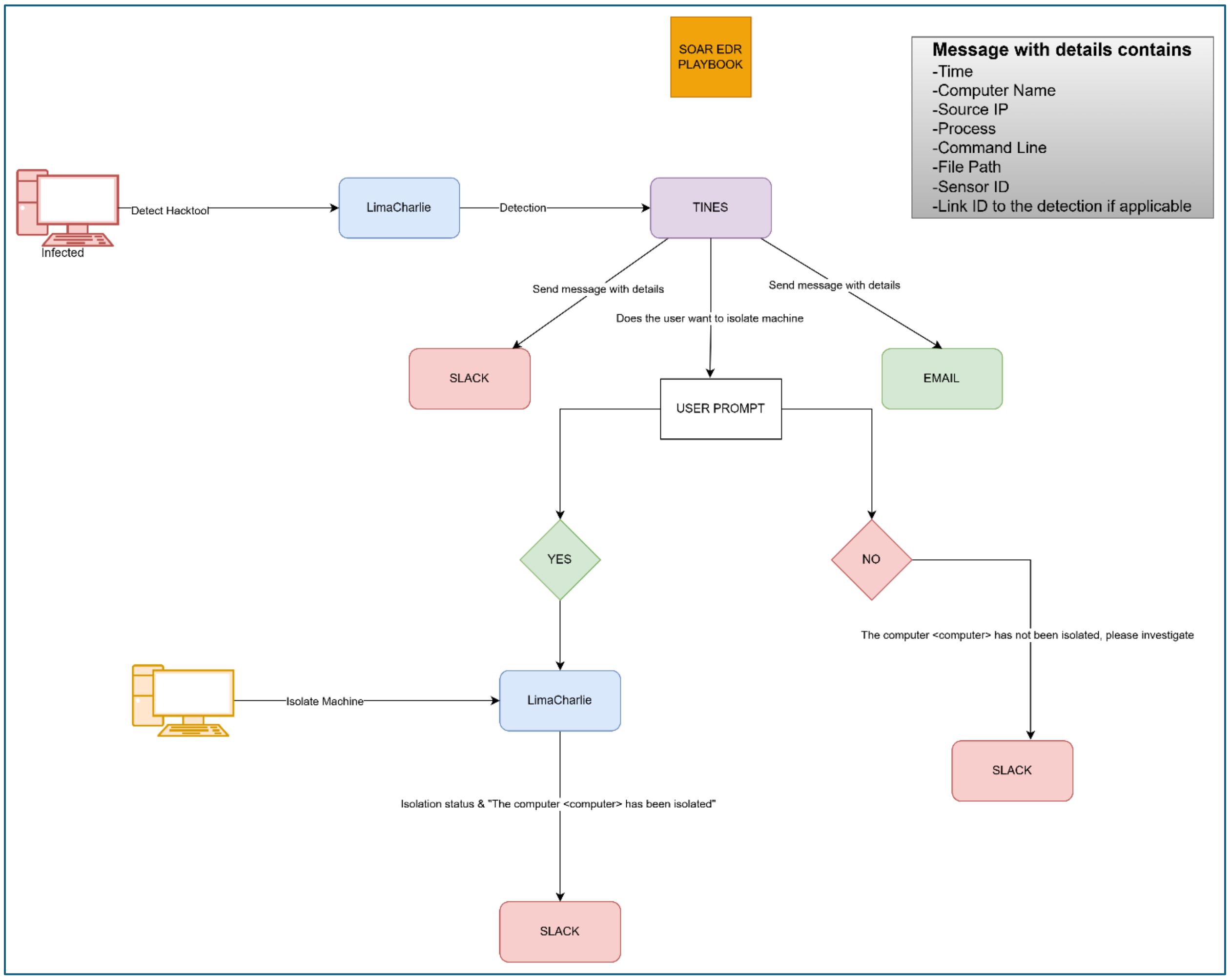
1. **Tool Agnosticism**: While we used **Tines** as the SOAR platform and **LimaCharlie** as the EDR tool, the principles of SOAR implementation remain largely vendor-agnostic. The key takeaway is that SOC processes should be designed with flexibility in mind, as the tools may vary, but the goals of automation and structured responses stay the same.
2. **Importance of Iteration in Playbook Development**: During the creation of playbooks, it's crucial to anticipate changes and continuously refine the workflows. Playbooks evolve based on operational requirements, tool limitations, and new security threats.
3. **Practical Experience Matters**: This project reinforced that hands-on practice with SOAR tools is essential for building confidence and technical proficiency in SOC operations. The integration of LimaCharlie and Tines provided valuable insight into orchestrating responses in real-time.

**Conclusion:**

This project successfully demonstrated the use of **SOAR platforms** like Tines to streamline SOC workflows by automating repetitive tasks, providing real-time incident notifications, and enabling user interaction during incident response. The integration of EDR with SOAR platforms is a critical strategy for modern SOC teams to ensure that security incidents are handled efficiently and effectively. This hands-on project provided a solid foundation for future SOC automation projects and real-world application of SOAR tools.

## Part 1

This s what my workflow looks like and everything I'd be configuring and integrating.


SOAR EDR PLAYBOOK
Message with details contains
-Time
-Computer Name
-Source IP
-Process
-Command Line
-File Path
-Sensor ID
-Link ID to the detection if applicable
Infected
Detect Hacktool
LimaCharlie
Detection
TINES
Send message with details
Send message with details
Does the user want to isolate machine
SLACK
EMAIL
USER PROMPT
YES
NO
The computer <computer> has not been isolated, please investigate
Isolate Machine
LimaCharlie
SLACK
Isolation status & "The computer <computer> has been isolated"
SLACK

## Part 2: LimaCharlie EDR and Tines Integration

In this phase, the goal is to:

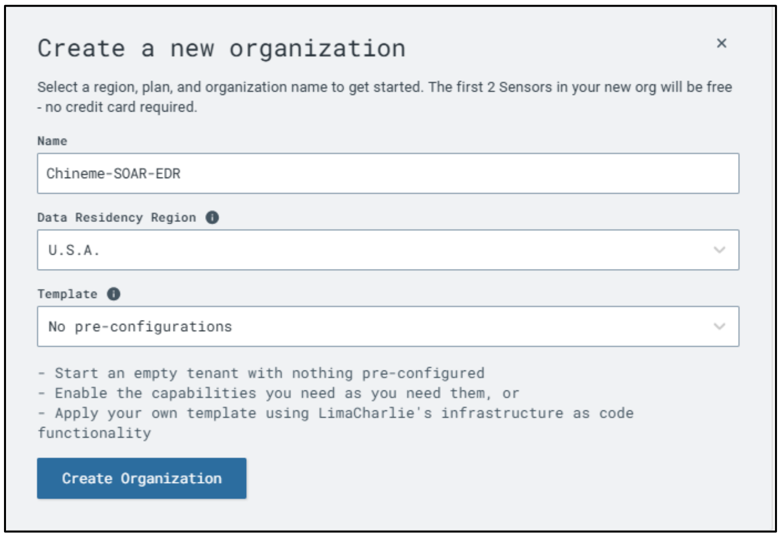
- Install and set up **LimaCharlie EDR** on my **Windows Server 2022 virtual machine (VM)** running in VirtualBox.
- Ensure the VM is enrolled in **LimaCharlie** and is generating **event data**.
- Prepare the environment for creating **custom detection and response rules**.

**Step-by-Step Setup Process:**

1. **Environment Setup (Windows Server 2022 in VirtualBox):**

- I will be using my existing VM running Windows Server 2022 on VirtualBox. It's important to ensure that the VM has internet connectivity since LimaCharlie needs to communicate with its cloud dashboard to report events.

2. **LimaCharlie Setup:**
    - Go to the **LimaCharlie website** and sign up if you haven't done so already.
    - Once logged in, follow the LimaCharlie onboarding wizard, which includes creating an organization.
    - Give your organization a meaningful name, such as **"chineme-SOAR-EDR"**.
    - Then select the appropriate **data residency region**, like "USA" or your preferred location.



Create a new organization

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

Name: Chineme-SOAR-EDR

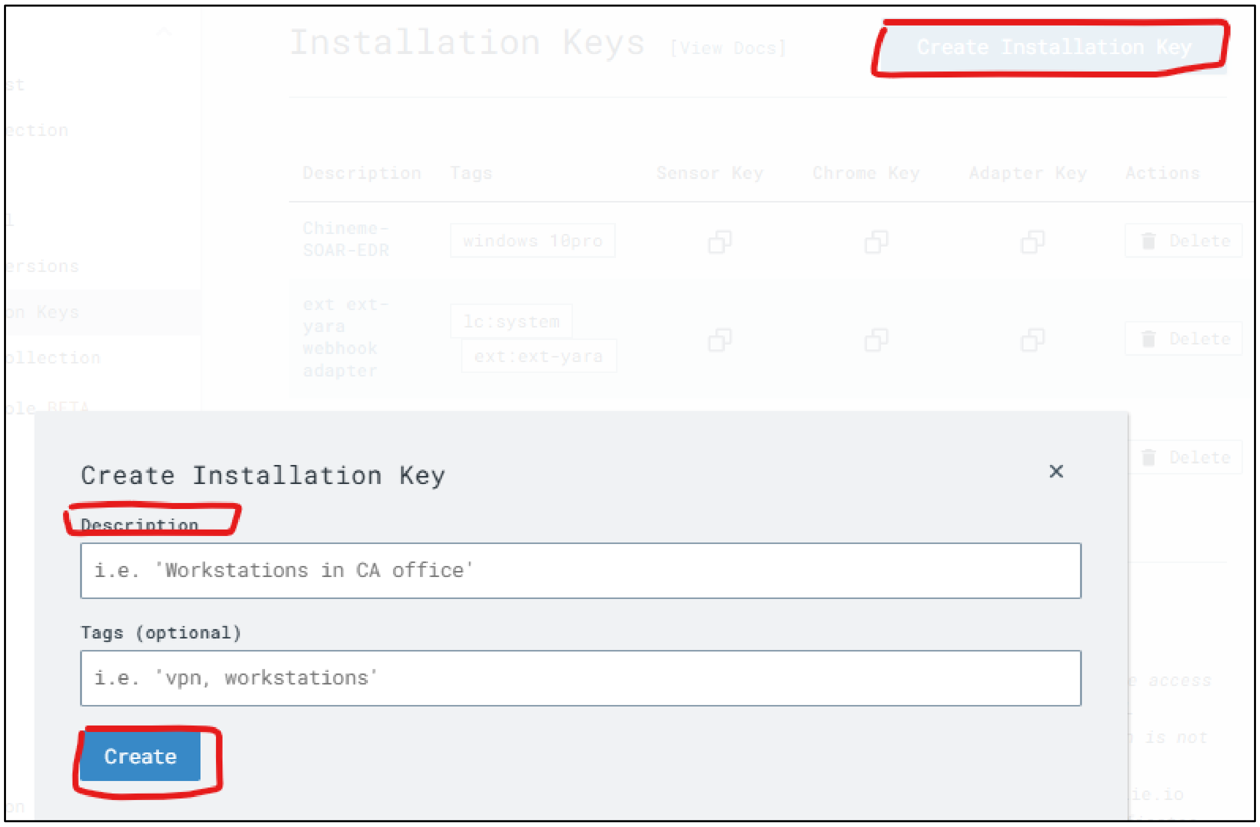
Data Residency Region: U.S.A.

Template: No pre-configurations

- Start an empty tenant with nothing pre-configured
- Enable the capabilities you need as you need them, or
- Apply your own template using LimaCharlie's infrastructure as code functionality

3. **Creating Installation Keys in LimaCharlie:**

- Navigate to the **Installation Keys** section on the LimaCharlie dashboard. These keys are used to enroll sensors (my VM) into the LimaCharlie environment.
- Click on **Create Installation Key** and provide a description such as **"Chineme-SOAR-EDR"**.
- Save the key for later use during the agent installation process.



Create Installation Key

Description

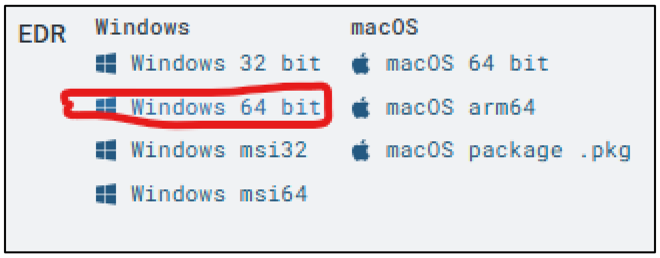
i.e. 'Workstations in CA office'

Tags (optional)

i.e. 'vpn, workstations'

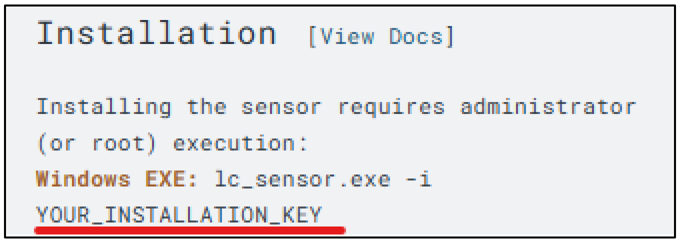
Create

4. **Installing the LimaCharlie Agent on Windows Server 2022:**
   - Now that I have my installation key, it's time to install the **LimaCharlie Agent** on my Windows Server VM.
   - Download the **LimaCharlie Windows Agent** from the platform.
     - In LimaCharlie, navigate to the **Downloads** section and select the Windows agent for **64-bit systems**.



| EDR | Windows | macOS |
|---|---|---|
| | Windows 32 bit | macOS 64 bit |
| | Windows 64 bit | macOS arm64 |
| | Windows msi32 | macOS package .pkg |
| | Windows msi64 | |

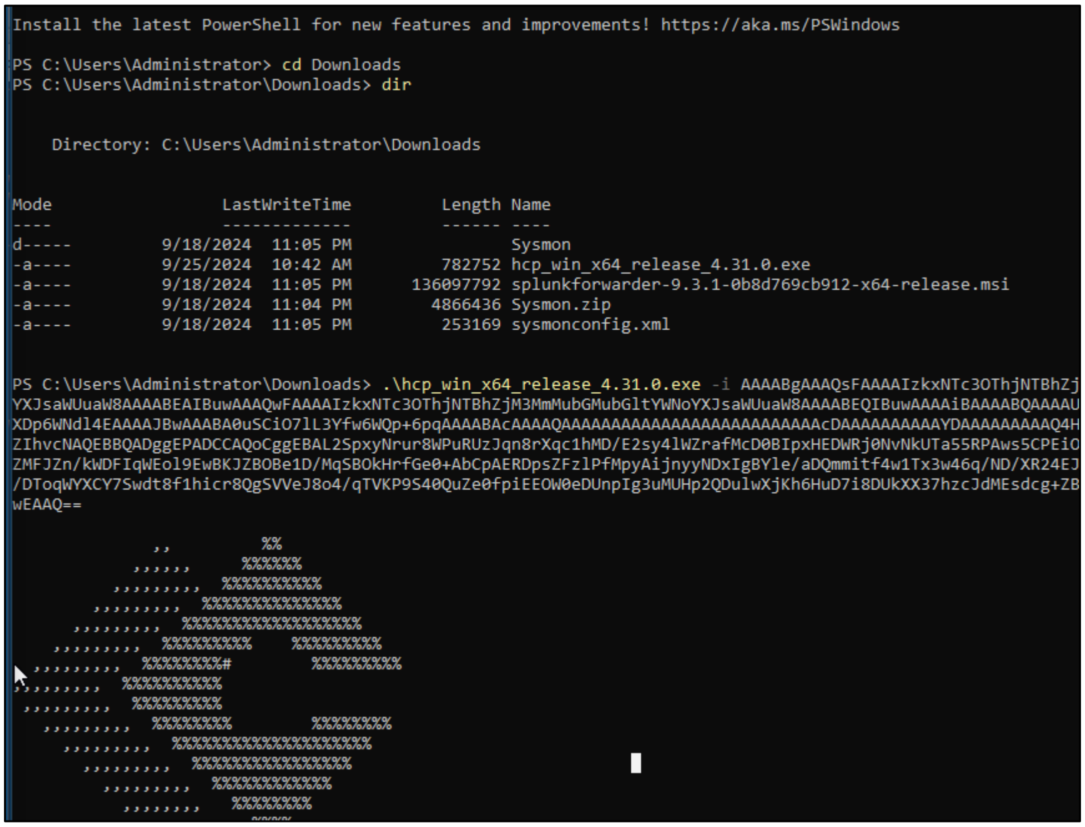
- Once downloaded, I'll open **PowerShell as Administrator** to install the agent:
  - Navigate to the directory where the file was downloaded using cd Downloads.
  - Run the installation using the command:



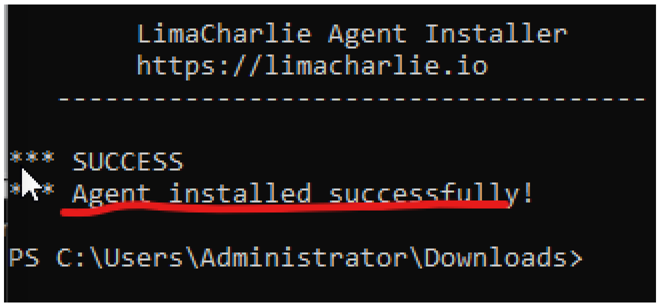
```
Installation [View Docs]

Installing the sensor requires administrator
(or root) execution:
Windows EXE: lc_sensor.exe -i
YOUR_INSTALLATION_KEY
```

.\limaCharlieInstaller.exe -i <Your_Installation_Key>

  - I'll ensure the agent installs successfully, and I should see a confirmation message once the installation is complete.

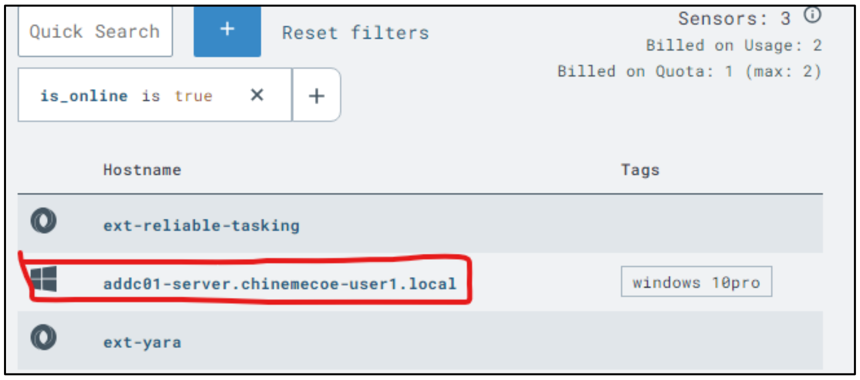


```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Administrator> cd Downloads
PS C:\Users\Administrator\Downloads> dir


    Directory: C:\Users\Administrator\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/18/2024  11:05 PM                Sysmon
-a----         9/25/2024  10:42 AM         782752 hcp_win_x64_release_4.31.0.exe
-a----         9/18/2024  11:05 PM      136097792 splunkforwarder-9.3.1-0b8d769cb912-x64-release.msi
-a----         9/18/2024  11:04 PM        4866436 Sysmon.zip
-a----         9/18/2024  11:05 PM         253169 sysmonconfig.xml


PS C:\Users\Administrator\Downloads> .\hcp_win_x64_release_4.31.0.exe -i AAAABgAAAQsFAAAAIzkxNTc3OThjNTBhZj
YXJsaWUuaW8AAAABEAIBuwAAAQwFAAAAIzkxNTc3OThjNTBhZjM3MmMubGMubGltYWNoYXJsaWUuaW8AAAABEQIBuwAAAAiBAAAABQAAAAU
XDp6WNdl4EAAAAJBwAAABA0uSCiO7lL3Yfw6WQp+6pqAAAABAcAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAcDAAAAAAAAAAAYDAAAAAAAAAAAQ4H
ZIhvcNAQEBBQADggEPADCCAQoCggEBAL2SpxyNrur8WPuRUzJqn8rXqc1hMD/E2sy4lWZrafMcD0BIpxHEDWRj0NvNkUTa55RPAws5CPEiO
ZMFJZn/kWDFIqWEol9EwBKJZBOBe1D/MqSBOkHrfGe0+AbCpAERDpsZFzlPfMpyAijnyyNDxIgBYle/aDQmmitf4w1Tx3w46q/ND/XR24EJ
/DToqWYXCY7Swdt8f1hicr8QgSVVeJ8o4/qTVKP9S40QuZe0fpiEEOW0eDUnpIg3uMUHp2QDulwXjKh6HuD7i8DUkXX37hzcJdMEsdcg+ZB
wEAAQ==

              ,,         %%
            ,,,,,,     %%%%%%
          ,,,,,,,,,  %%%%%%%%%%
        ,,,,,,,,,  %%%%%%%%%%%%%%
      ,,,,,,,,,  %%%%%%%%%%%%%%%%%%
    ,,,,,,,,,  %%%%%%%%%     %%%%%%%%%
  ,,,,,,,,,  %%%%%%%%#         %%%%%%%%%
,,,,,,,,,  %%%%%%%%%%
 ,,,,,,,,,  %%%%%%%%%
   ,,,,,,,,,  %%%%%%%%         %%%%%%%%
     ,,,,,,,,,  %%%%%%%%%%%%%%%%%%%%
       ,,,,,,,,,  %%%%%%%%%%%%%%%%
         ,,,,,,,,,  %%%%%%%%%%%%
           ,,,,,,,,  %%%%%%%%
```



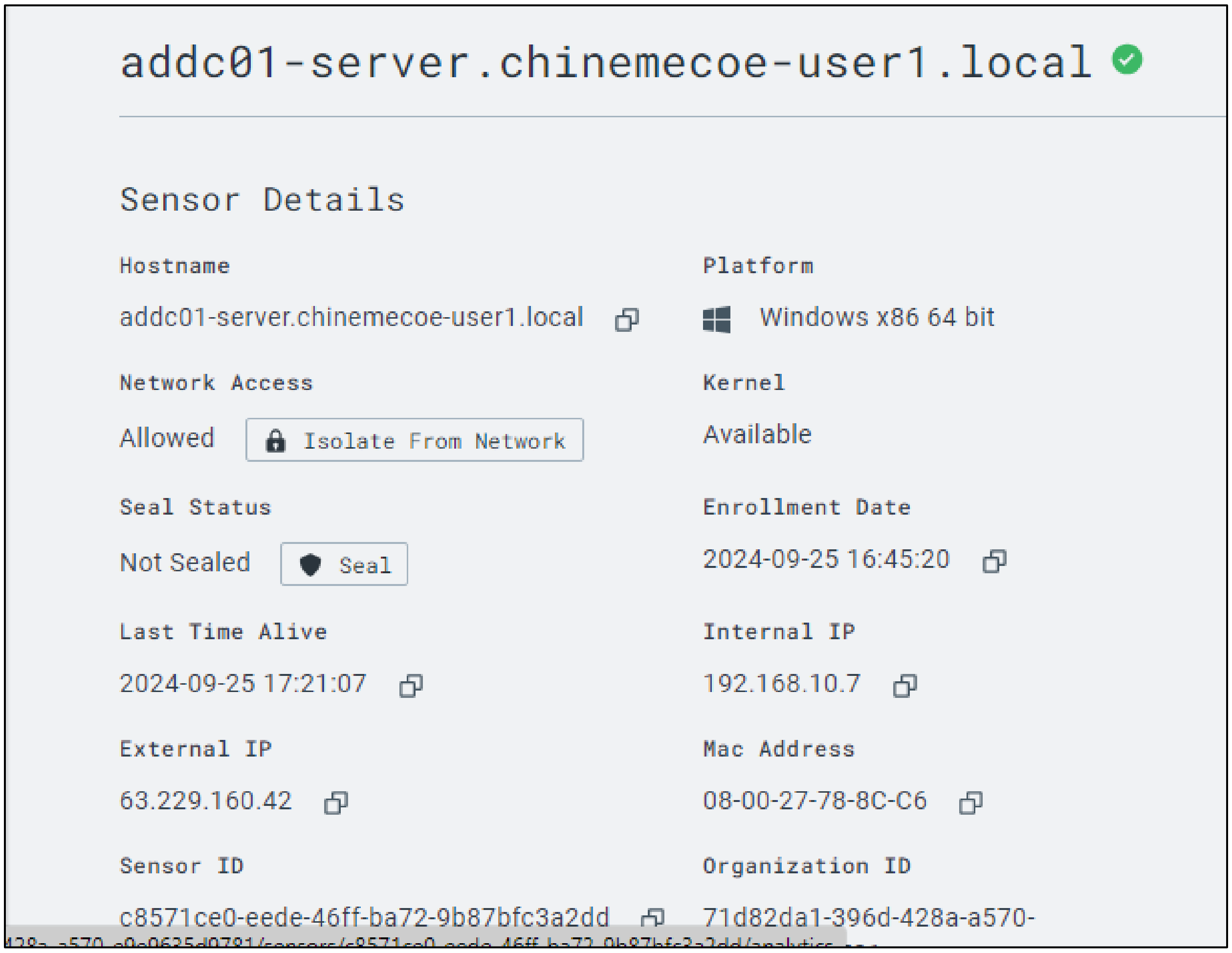
```
      LimaCharlie Agent Installer
      https://limacharlie.io
   -------------------------------------

*** SUCCESS
*** Agent installed successfully!

PS C:\Users\Administrator\Downloads>
```

5. **Verifying Enrollment in LimaCharlie:**
   - After installation, head back to the **LimaCharlie dashboard** and navigate to **Sensors**.



| Hostname | Tags |
|---|---|
| ext-reliable-tasking | |
| addc01-server.chinemecoe-user1.local | windows 10pro |
| ext-yara | |

   - You should now see **Windows Server 2022 VM** listed under the **Sensors List**. The dashboard will provide details such as:
     - Hostname
     - Internal and external IP addresses
     - Enrollment date
     - Platform type
   - I can also see basic analytics like **event collection** and **file system** data.

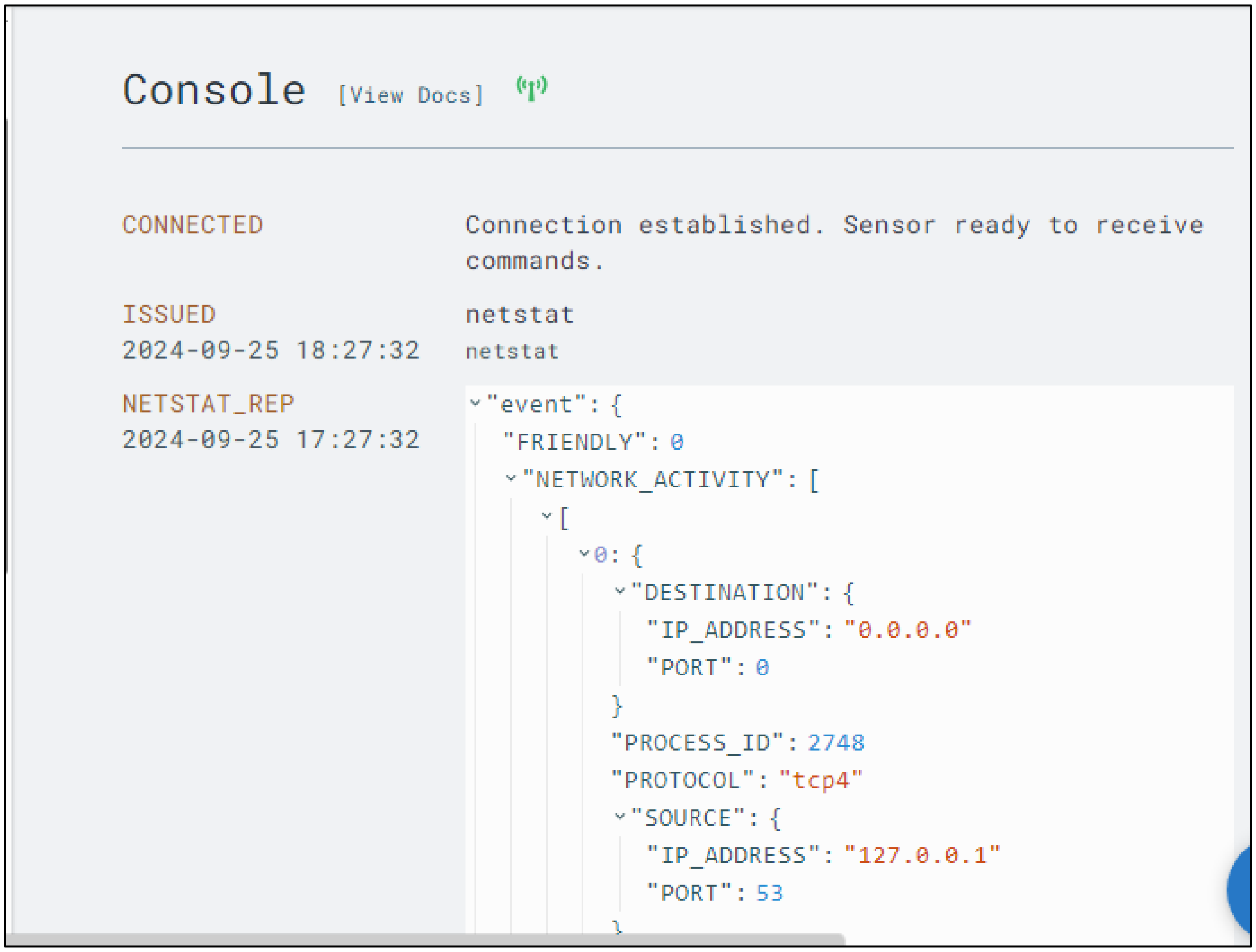


addc01-server.chinemecoe-user1.local ✓

Sensor Details

| | |
|---|---|
| Hostname | Platform |
| addc01-server.chinemecoe-user1.local | Windows x86 64 bit |
| Network Access | Kernel |
| Allowed | Available |
| Seal Status | Enrollment Date |
| Not Sealed | 2024-09-25 16:45:20 |
| Last Time Alive | Internal IP |
| 2024-09-25 17:21:07 | 192.168.10.7 |
| External IP | Mac Address |
| 63.229.160.42 | 08-00-27-78-8C-C6 |
| Sensor ID | Organization ID |
| c8571ce0-eede-46ff-ba72-9b87bfc3a2dd | 71d82da1-396d-428a-a570- |

6. **Initial Test Commands:**
   - With my sensor now visible in the LimaCharlie dashboard, it’s time to run some basic commands to verify the setup.
     - I’ll use the **Netstat** command from within the LimaCharlie console to view active network connections on my Windows Server VM:
       - In LimaCharlie, navigate to **Console** under the enrolled sensor and run:
       - I’ll verify that the output shows network connections along with their process IDs (PIDs).



```
CONNECTED              Connection established. Sensor ready to receive
                       commands.

ISSUED                 netstat
2024-09-25 18:27:32    netstat

NETSTAT_REP            "event": {
2024-09-25 17:27:32      "FRIENDLY": 0
                         "NETWORK_ACTIVITY": [
                           [
                             0: {
                               "DESTINATION": {
                                 "IP_ADDRESS": "0.0.0.0"
                                 "PORT": 0
                               }
                               "PROCESS_ID": 2748
                               "PROTOCOL": "tcp4"
                               "SOURCE": {
                                 "IP_ADDRESS": "127.0.0.1"
                                 "PORT": 53
```

7. **Preparing for Custom Detection and Response:**

   - Now that my VM is generating telemetry and I've verified the communication with LimaCharlie, I'm ready for the next phase of the project: **creating detection and response rules**.
   - The focus will be on generating specific events related to **password recovery tools** and crafting custom responses to mitigate the threat.

## Part 3: Generating Telemetry and Creating Detection & Response Rules in LimaCharlie

In this part, our main objective is to:

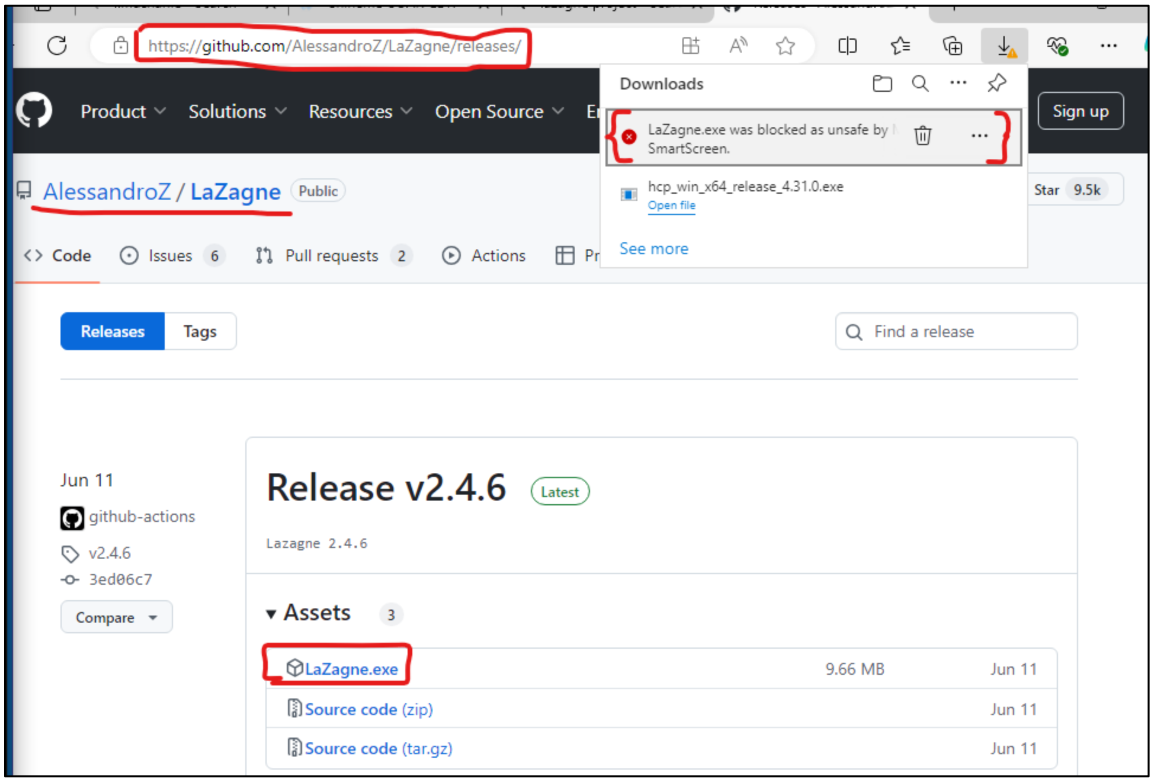
1. Generate telemetry from the **LaZagnea password recovery tool**.
2. Create a **detection and response rule** in LimaCharlie to identify this activity.
3. Set the groundwork for forwarding the detection to **Tines** for automated actions.

### Step 1: Download and Run LaZagne

The first step is to generate activity by downloading and running **LaZagne**, a popular password recovery tool. LimaCharlie will monitor this execution, allowing us to create a detection rule based on telemetry.

1. **Download LaZagne:**
   - Visit **LaZagne 's GitHub** repository.

     Lazagne: https://github.com/AlessandroZ/LaZagne
   - Navigate to the **Releases** section and download the **LaZagne.exe** file.



https://github.com/AlessandroZ/LaZagne/releases/

Downloads

LaZagne.exe was blocked as unsafe by SmartScreen.

hcp_win_x64_release_4.31.0.exe
Open file

See more

AlessandroZ / LaZagne Public

Release v2.4.6 Latest

Lazagne 2.4.6

Jun 11
github-actions
v2.4.6
3ed06c7

Assets 3

| | | |
|---|---|---|
| LaZagne.exe | 9.66 MB | Jun 11 |
| Source code (zip) | | Jun 11 |
| Source code (tar.gz) | | Jun 11 |

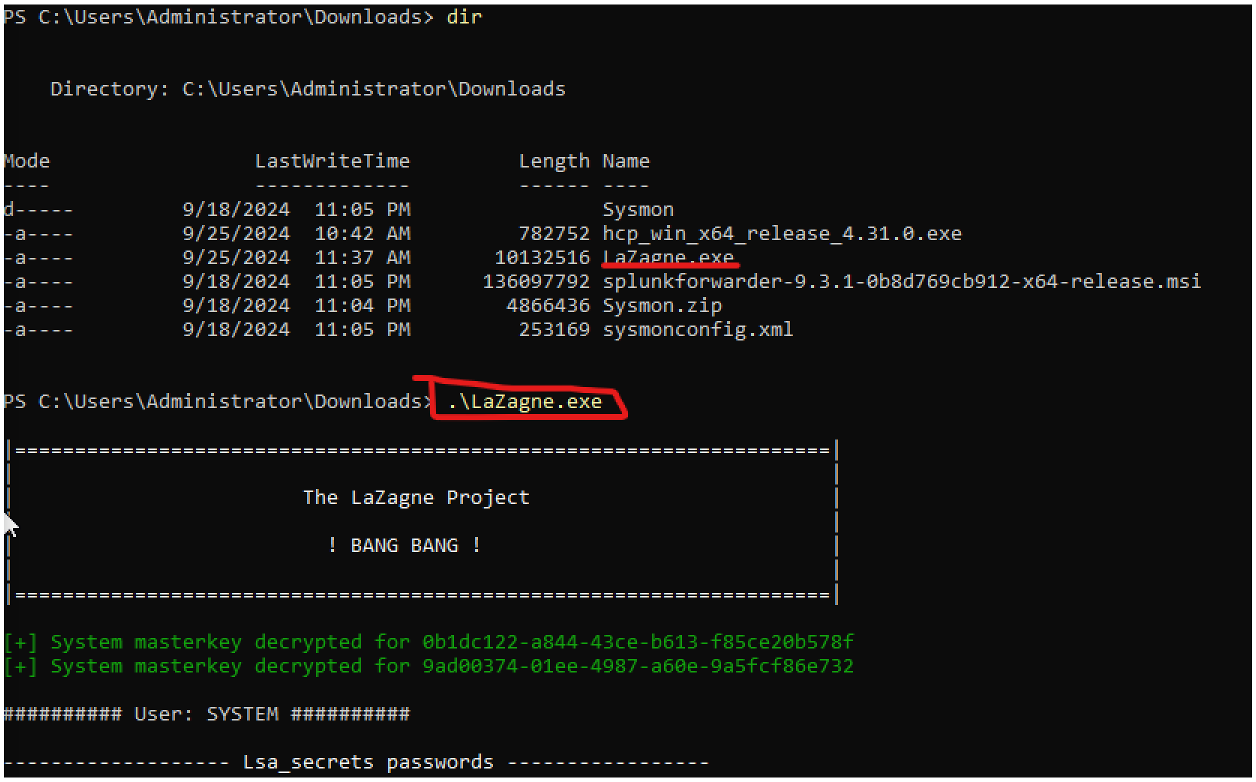
2. **Disable Windows Security:**
   - Open **Windows Security** and go to **Virus & Threat Protection**.

- Temporarily **disable real-time protection** to prevent Windows Defender from blocking LaZagne.

***Note**: Disabling real-time protection is necessary for this demo but should be re-enabled after completing the project.*

3. **Running LaZagne:**
   - Open **PowerShell** in the directory where LaZagne is downloaded.
   - Run the tool by typing:

.\LaZagne.exe



```
PS C:\Users\Administrator\Downloads> dir


    Directory: C:\Users\Administrator\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/18/2024  11:05 PM                Sysmon
-a----         9/25/2024  10:42 AM         782752 hcp_win_x64_release_4.31.0.exe
-a----         9/25/2024  11:37 AM       10132516 LaZagne.exe
-a----         9/18/2024  11:05 PM      136097792 splunkforwarder-9.3.1-0b8d769cb912-x64-release.msi
-a----         9/18/2024  11:04 PM        4866436 Sysmon.zip
-a----         9/18/2024  11:05 PM         253169 sysmonconfig.xml


PS C:\Users\Administrator\Downloads> .\LaZagne.exe

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

[+] System masterkey decrypted for 0b1dc122-a844-43ce-b613-f85ce20b578f
[+] System masterkey decrypted for 9ad00374-01ee-4987-a60e-9a5fcf86e732

########## User: SYSTEM ##########

------------------- Lsa_secrets passwords -----------------
```

4. At this point, LimaCharlie should begin capturing telemetry related to the execution of LaZagne.

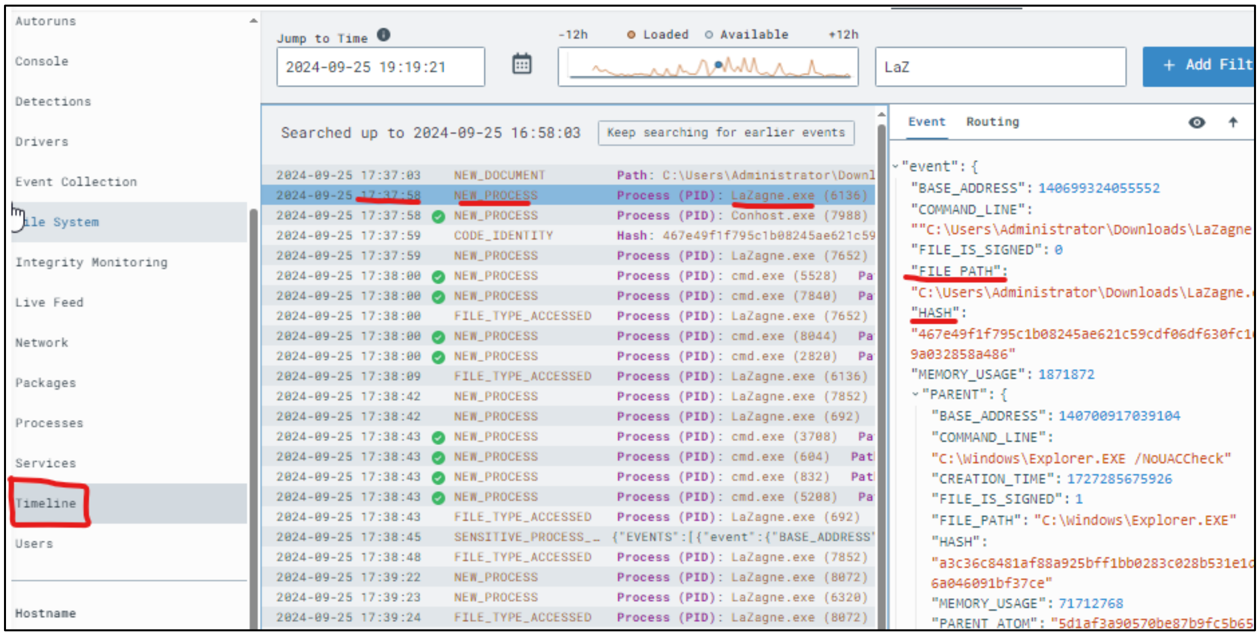
**Step 2: Viewing Telemetry in LimaCharlie**

After running LaZagne, we can head over to **LimaCharlie** to view the generated events.

1. **Navigate to the Sensor:**
   - In LimaCharlie, go to the **Sensors List** and select your **Windows Server VM**.
2. **View the Event Timeline:**

- Navigate to the **Timeline** tab to see recent events. Since we just executed LaZagne, the telemetry should be visible here.
- Filter the events using the term **LaZagne** to quickly find the specific process creation event.

3. **Telemetry Details:**
    - Click on the event for more details. Here, you'll see useful information such as the **file path**, **command line arguments**, **process ID**, and more.
    - This information will form the foundation for our custom detection rule.



```
Searched up to 2024-09-25 16:58:03

2024-09-25 17:37:03    NEW_DOCUMENT           Path: C:\Users\Administrator\Downl
2024-09-25 17:37:58    NEW_PROCESS            Process (PID): LaZagne.exe (6136)
2024-09-25 17:37:58  ✓ NEW_PROCESS            Process (PID): Conhost.exe (7988)
2024-09-25 17:37:59    CODE_IDENTITY          Hash: 467e49f1f795c1b08245ae621c59
2024-09-25 17:37:59    NEW_PROCESS            Process (PID): LaZagne.exe (7652)
2024-09-25 17:38:00  ✓ NEW_PROCESS            Process (PID): cmd.exe (5528)   Pa
2024-09-25 17:38:00  ✓ NEW_PROCESS            Process (PID): cmd.exe (7840)   Pa
2024-09-25 17:38:00    FILE_TYPE_ACCESSED     Process (PID): LaZagne.exe (7652)
2024-09-25 17:38:00  ✓ NEW_PROCESS            Process (PID): cmd.exe (8044)   Pa
2024-09-25 17:38:00  ✓ NEW_PROCESS            Process (PID): cmd.exe (2820)   Pa
2024-09-25 17:38:09    FILE_TYPE_ACCESSED     Process (PID): LaZagne.exe (6136)
2024-09-25 17:38:42    NEW_PROCESS            Process (PID): LaZagne.exe (7852)
2024-09-25 17:38:42    NEW_PROCESS            Process (PID): LaZagne.exe (692)
2024-09-25 17:38:43  ✓ NEW_PROCESS            Process (PID): cmd.exe (3708)   Pa
2024-09-25 17:38:43  ✓ NEW_PROCESS            Process (PID): cmd.exe (604)   Pat
2024-09-25 17:38:43  ✓ NEW_PROCESS            Process (PID): cmd.exe (832)   Pat
2024-09-25 17:38:43  ✓ NEW_PROCESS            Process (PID): cmd.exe (5208)   Pa
2024-09-25 17:38:43    FILE_TYPE_ACCESSED     Process (PID): LaZagne.exe (692)
2024-09-25 17:38:45    SENSITIVE_PROCESS_…  {"EVENTS":[{"event":{"BASE_ADDRESS"
2024-09-25 17:38:48    FILE_TYPE_ACCESSED     Process (PID): LaZagne.exe (7852)
2024-09-25 17:39:22    NEW_PROCESS            Process (PID): LaZagne.exe (8072)
2024-09-25 17:39:23    NEW_PROCESS            Process (PID): LaZagne.exe (6320)
2024-09-25 17:39:24    FILE_TYPE_ACCESSED     Process (PID): LaZagne.exe (8072)
```

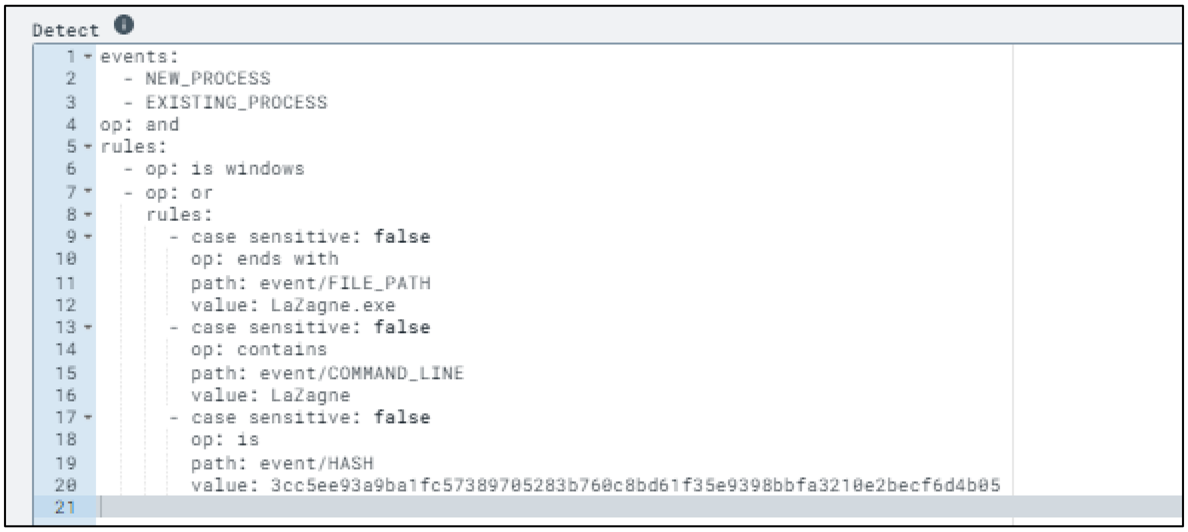
```
"event": {
  "BASE_ADDRESS": 140699324055552
  "COMMAND_LINE":
  ""C:\Users\Administrator\Downloads\LaZagne
  "FILE_IS_SIGNED": 0
  "FILE_PATH":
  "C:\Users\Administrator\Downloads\LaZagne.
  "HASH":
  "467e49f1f795c1b08245ae621c59cdf06df630fc1
  9a032858a486"
  "MEMORY_USAGE": 1871872
  "PARENT": {
    "BASE_ADDRESS": 140700917039104
    "COMMAND_LINE":
    "C:\Windows\Explorer.EXE /NoUACCheck"
    "CREATION_TIME": 1727285675926
    "FILE_IS_SIGNED": 1
    "FILE_PATH": "C:\Windows\Explorer.EXE"
    "HASH":
    "a3c36c8481af88a925bff1bb0283c028b531e1c
    6a046091bf37ce"
    "MEMORY_USAGE": 71712768
    "PARENT_ATOM": "5d1af3a90570be87b9fc5b65
```

## Step 3: Creating the Detection Rule in LimaCharlie

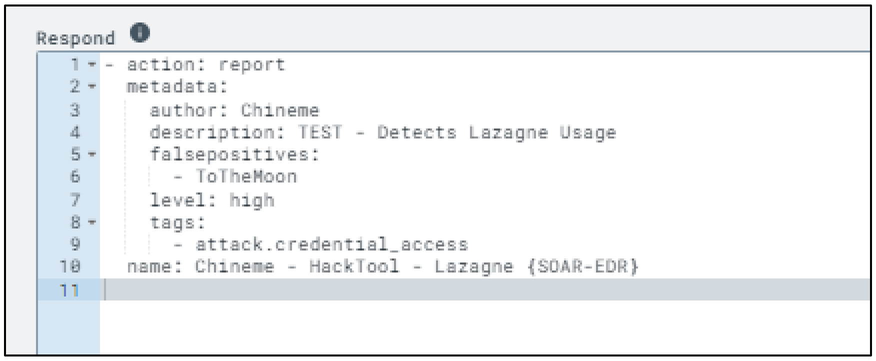
Now that we have telemetry from LaZagne, it's time to create a **detection and response (DNR) rule** in LimaCharlie. This rule will detect the execution of LaZagne in the future and trigger a response.

1. **Navigate to Detection & Response (DNR) Rules:**
    - Go to the **Automation** section in LimaCharlie.
    - Click on **New Rule** to create a new detection rule.
2. **Base the Rule on Existing Templates:**
    - Instead of starting from scratch, use an existing rule as a foundation.
    - Search for **Credential Access** related rules, as LaZagne is a password recovery tool. Select a rule related to **process creation** and view its content.

3. **Customizing the Rule for LaZagne:**
    - Edit the rule to fit our use case. The key event type we're interested in is **new process creation**.
    - We will match the **file path** and **command line arguments** from the telemetry we viewed earlier.
    - For example, the rule can trigger if the file path ends with **LaZagne.exe** or if the **command line** contains specific arguments (like **all**).
4. **Refining the Criteria:**
    - The rule should check:
        - **Process Creation Event**: The event type should be either **new process** or **existing process**.
        - **File Path**: The file path should end with LaZagne.exe (case-insensitive).
        - **Command Line**: The command line should contain **all** or similar relevant arguments from the telemetry.
        - **File Hash**: Optionally, you can match the hash of the LaZagne executable to ensure accuracy.



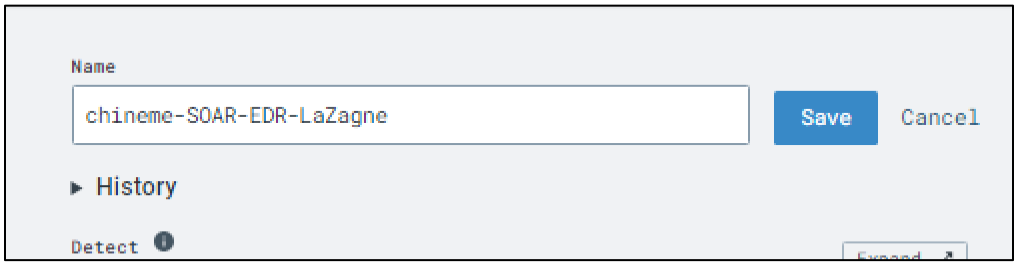
```
events:
  - NEW_PROCESS
  - EXISTING_PROCESS
op: and
rules:
  - op: is windows
  - op: or
    rules:
      - case sensitive: false
        op: ends with
        path: event/FILE_PATH
        value: LaZagne.exe
      - case sensitive: false
        op: contains
        path: event/COMMAND_LINE
        value: LaZagne
      - case sensitive: false
        op: is
        path: event/HASH
        value: 3cc5ee93a9ba1fc57389705283b760c8bd61f35e9398bbfa3210e2becf6d4b05
```


Respond

```
- action: report
  metadata:
    author: Chineme
    description: TEST - Detects Lazagne Usage
    falsepositives:
      - ToTheMoon
    level: high
    tags:
      - attack.credential_access
  name: Chineme - HackTool - Lazagne {SOAR-EDR}
```

Here is the GitHub link to the rule: GitHub - MyDFIR/SOAR-EDR-Project

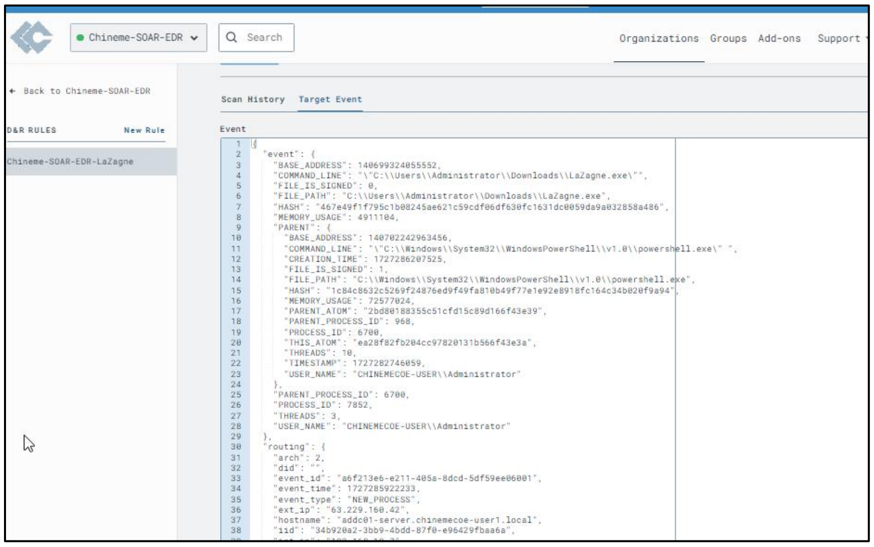
5. **Saving the Rule:**
   - Once customized, name the rule something relevant like **"LaZagne Detection Rule"** and save it.


Name: chineme-SOAR-EDR-LaZagne

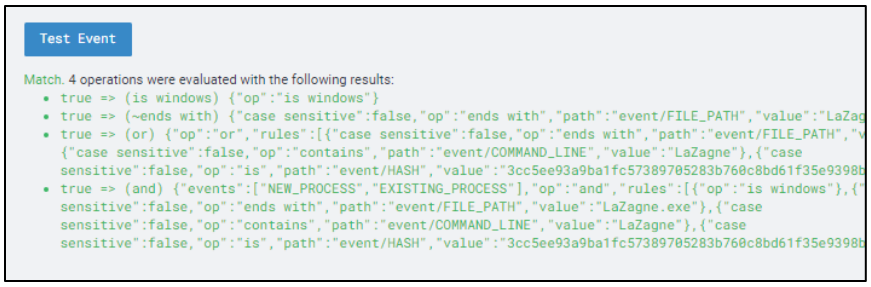
**Step 4: Testing the Rule**

After saving the rule, it's important to test whether it works as intended.

1. **Testing with Past Events:**
   - Go back to the previously captured event for LaZagne.
   - Use LimaCharlie's **Test Event** functionality by pasting the raw event data into the test field. Run the test and ensure that the rule is correctly detecting the event.

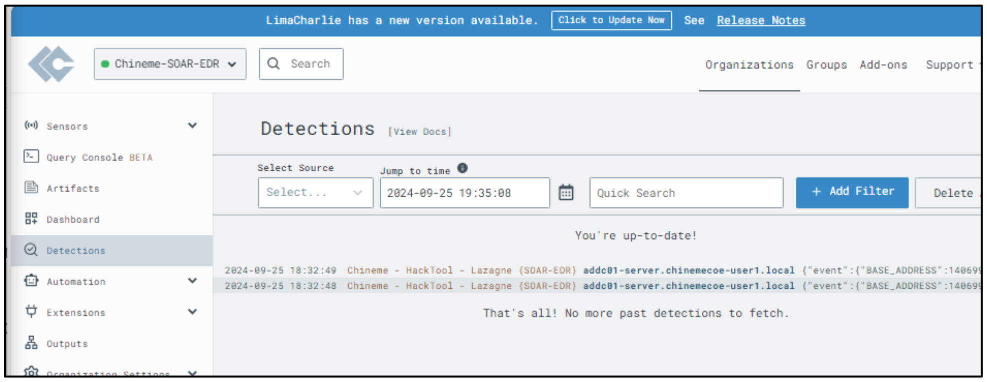


```
{
  "event": {
    "BASE_ADDRESS": 140699324055552,
    "COMMAND_LINE": "\"C:\\Users\\Administrator\\Downloads\\LaZagne.exe\"",
    "FILE_IS_SIGNED": 0,
    "FILE_PATH": "C:\\Users\\Administrator\\Downloads\\LaZagne.exe",
    "HASH": "467e49f1f795c1b08245ae621c59cdf06df630fc1631dc0059da9a032858a486",
    "MEMORY_USAGE": 4911104,
    "PARENT": {
      "BASE_ADDRESS": 140702242963456,
      "COMMAND_LINE": "\"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\" ",
      "CREATION_TIME": 1727286207525,
      "FILE_IS_SIGNED": 1,
      "FILE_PATH": "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe",
      "HASH": "1c84c8632c5269f24876ed9f49fa810b49f77e1e92e8918fc164c34b020f9a94",
      "MEMORY_USAGE": 72577024,
      "PARENT_ATOM": "2bd80188355c51cfd15c89d166f43e39",
      "PARENT_PROCESS_ID": 968,
      "PROCESS_ID": 6700,
      "THIS_ATOM": "ea28f82fb204cc97820131b566f43e3a",
      "THREADS": 10,
      "TIMESTAMP": 1727282746059,
      "USER_NAME": "CHINEMECOE-USER\\Administrator"
    },
    "PARENT_PROCESS_ID": 6700,
    "PROCESS_ID": 7852,
    "THREADS": 3,
    "USER_NAME": "CHINEMECOE-USER\\Administrator"
  },
  "routing": {
    "arch": 2,
    "did": "",
    "event_id": "a6f213e6-e211-405a-8dcd-5df59ee06001",
    "event_time": 1727285922233,
    "event_type": "NEW_PROCESS",
    "ext_ip": "63.229.160.42",
    "hostname": "addc01-server.chinemecoe-user1.local",
    "iid": "34b920a2-3bb9-4bdd-87f0-e96429fbaa6a",
```



```
Match. 4 operations were evaluated with the following results:
  • true => (is windows) {"op":"is windows"}
  • true => (~ends with) {"case sensitive":false,"op":"ends with","path":"event/FILE_PATH","value":"LaZag
  • true => (or) {"op":"or","rules":[{"case sensitive":false,"op":"ends with","path":"event/FILE_PATH","v
    {"case sensitive":false,"op":"contains","path":"event/COMMAND_LINE","value":"LaZagne"},{"case
    sensitive":false,"op":"is","path":"event/HASH","value":"3cc5ee93a9ba1fc57389705283b760c8bd61f35e9398b
  • true => (and) {"events":["NEW_PROCESS","EXISTING_PROCESS"],"op":"and","rules":[{"op":"is windows"},{"
    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":"3cc5ee93a9ba1fc57389705283b760c8bd61f35e9398b
```

2. **Running LaZagne Again:**
   - Clear the previous telemetry for a clean slate.
   - Run **LaZagne.exe** again in your Windows Server VM.
   - Check **LimaCharlie's detection tab** to verify that the rule triggered and generated a detection.



```
You're up-to-date!

2024-09-25 18:32:49  Chineme - HackTool - Lazagne {SOAR-EDR} addc01-server.chinemecoe-user1.local {"event":{"BASE_ADDRESS":140699
2024-09-25 18:32:48  Chineme - HackTool - Lazagne {SOAR-EDR} addc01-server.chinemecoe-user1.local {"event":{"BASE_ADDRESS":140699

That's all! No more past detections to fetch.
```

## Part 4: Integrating Slack, Tines, and LimaCharlie for Automation

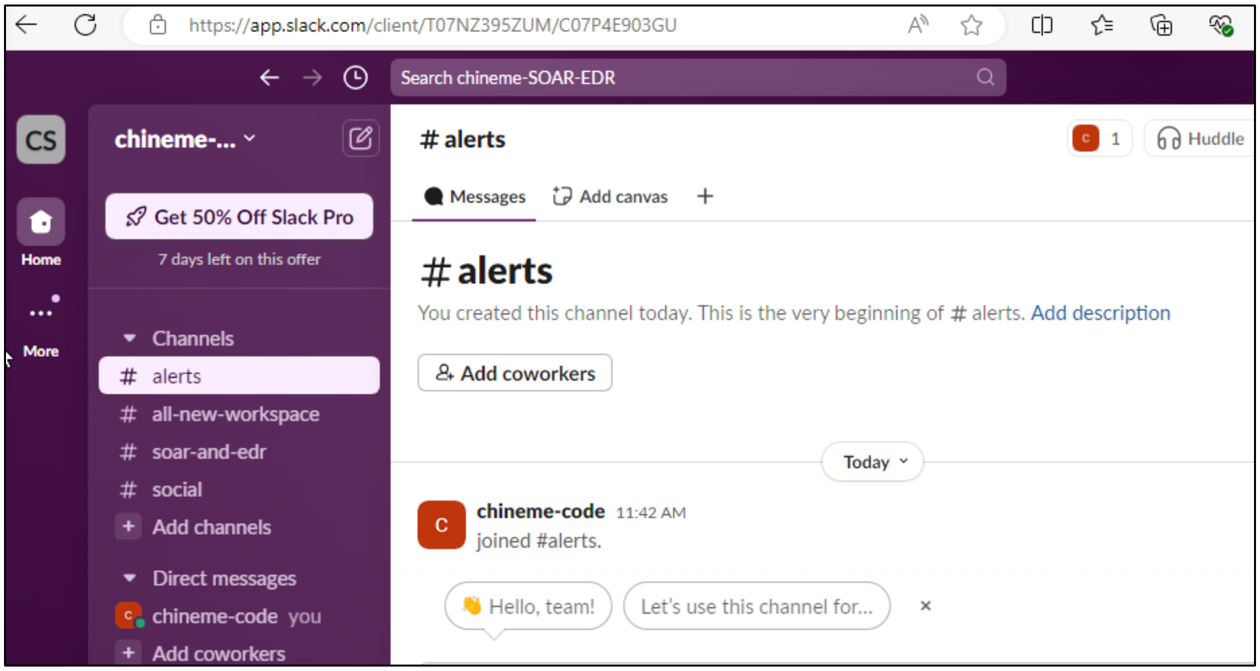
In this phase, we will:

1. Set up **Slack** to receive alerts.
2. Configure **Tines** to connect with LimaCharlie.
3. Test the connection between LimaCharlie and Tines to ensure that **SOAR** (Security Orchestration, Automation, and Response) processes are receiving the detections generated by LimaCharlie.

### Step 1: Setting up Slack

The first tool we'll configure is **Slack**, where we'll be sending our alerts from LimaCharlie via Tines.

1. **Create a Slack Workspace**:
   - Go to Slack.com and click on the "Get Started Free" button to create an account.
   - Use a **valid email account**, as Slack will send a confirmation code to complete the signup process.
   - Name your workspace something meaningful like **"chineme-SOAR-EDR"**.
2. **Create an Alerts Channel**:
   - In your new workspace, create a dedicated channel for alerts:
     - Click on **Add Channels** and select **Create a New Channel**.
     - Name the channel **alerts** and set it to **public**.
     - Skip inviting people for now.
   - This channel will receive detection alerts from LimaCharlie via Tines.



# alerts

You created this channel today. This is the very beginning of # alerts. Add description

Add coworkers

Today

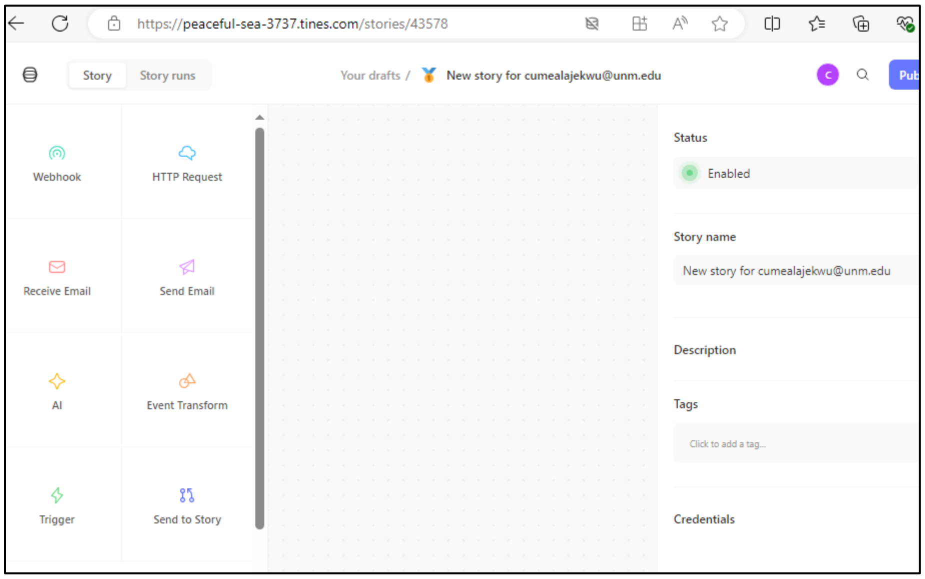
chineme-code 11:42 AM
joined #alerts.

Hello, team! | Let's use this channel for...

## Step 2: Setting up Tines

Next, we'll configure **Tines**, which will handle automation based on the detections reported by LimaCharlie.

1. **Create a Tines Account**:
   - Head over to [Tines.com]() and sign up using your email. You can also sign in with **Google** or **Microsoft** accounts.
   - Once logged in, exit the introduction tour by clicking on the "X" and "End Tour."



Status: Enabled

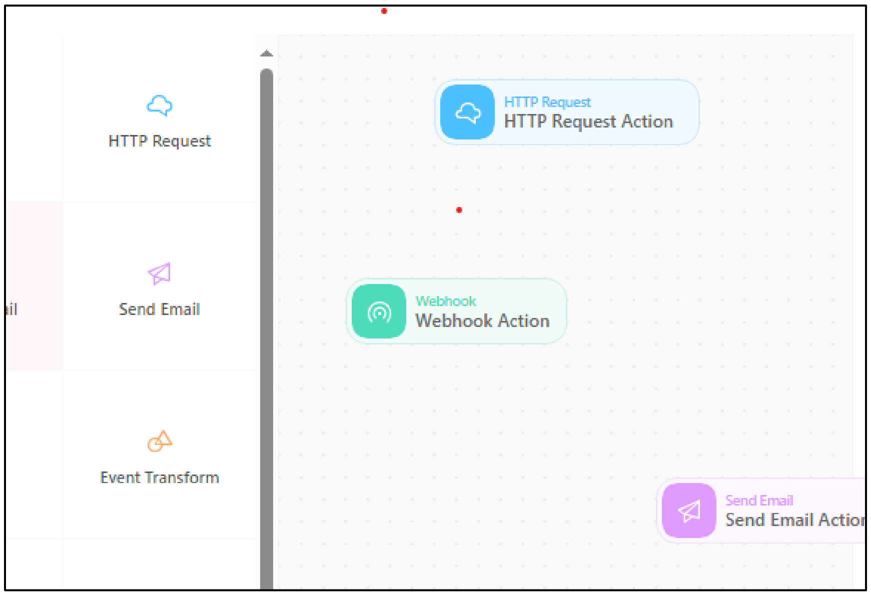
Story name: New story for cumealajekwu@unm.edu

Description

Tags: Click to add a tag...

Credentials

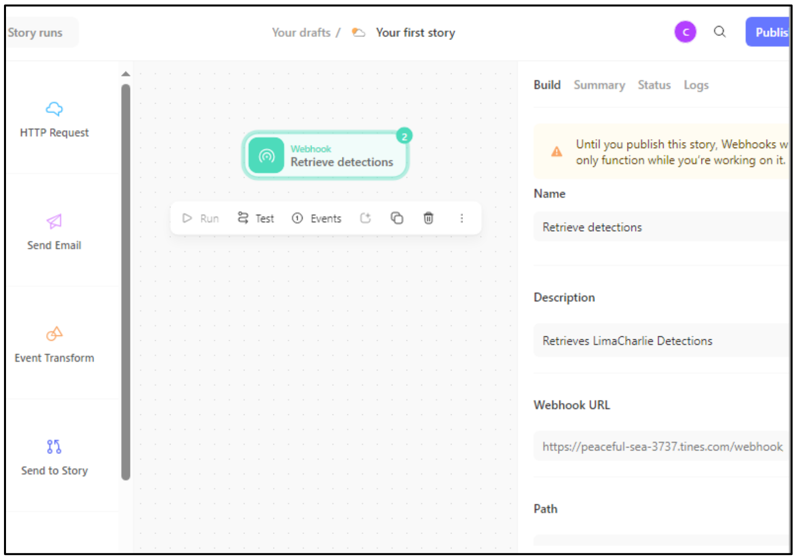
2. **Exploring Tines' Actions**:
   - On the left-hand side, you'll see different actions you can use to build your playbook (known as **stories** in Tines).
   - For example, you can drag an **HTTP Request** or **Web Hook** action onto the storyboard. We'll use a **Web Hook** to receive detections from LimaCharlie.
   - Templates are also available if you'd prefer starting with a pre-built example.



**Step 3: Connecting LimaCharlie and Tines**

Now that we have both **Slack** and **Tines** set up, the next step is to establish the connection between **LimaCharlie** and **Tines**, ensuring that detection data flows between them.

1. **Create a Web Hook in Tines**:
   - In Tines, drag a **Web Hook** onto the storyboard and name it **Retrieve Detections**.
   - For the **Web Hook URL**, copy the provided link.


Story runs

Your drafts / Your first story

Publis

HTTP Request

Send Email

Event Transform

Send to Story

Webhook
Retrieve detections

Run Test Events

Build Summary Status Logs

Until you publish this story, Webhooks w
only function while you're working on it.

Name

Retrieve detections

Description

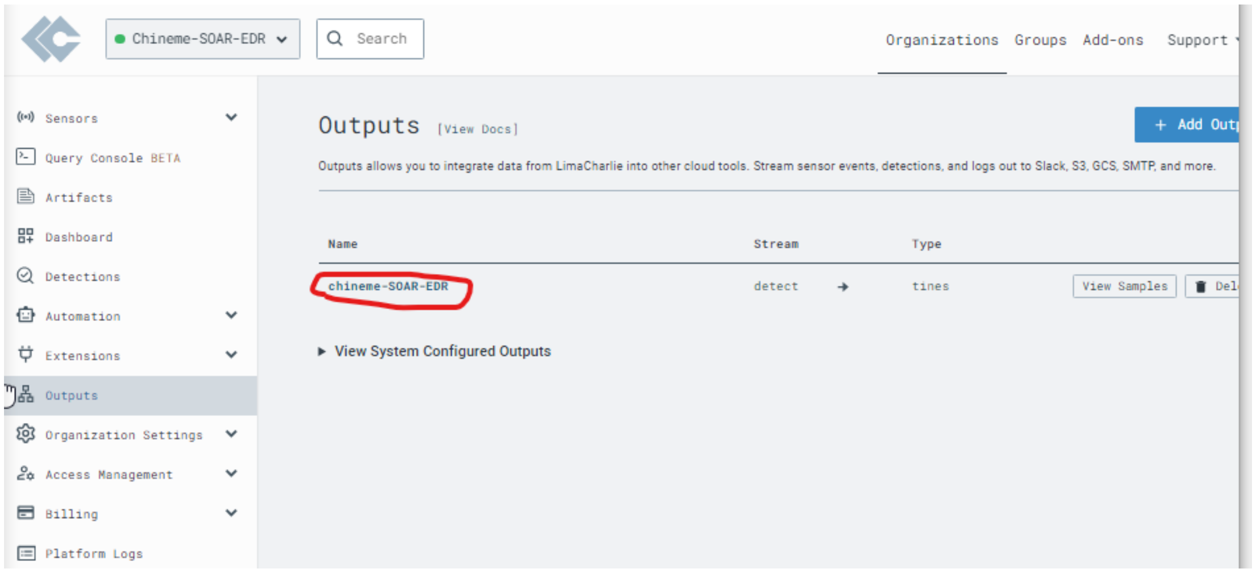
Retrieves LimaCharlie Detections

Webhook URL

https://peaceful-sea-3737.tines.com/webhook

Path

2. **Configuring Outputs in LimaCharlie**:
   - Head back to **LimaCharlie** and navigate to your organization's **Outputs**.
   - Click on **Add Output** and select **Detections** (since we want to capture all detections from our rule).
   - In the output configuration:
     - Select **Tines** as the destination.
     - Paste the **Web Hook URL** from Tines into the **Destination Host** field.
     - Name the output something like **chineme-SOAR-EDR**


Chineme-SOAR-EDR

Search

Organizations Groups Add-ons Support

Sensors
Query Console BETA
Artifacts
Dashboard
Detections
Automation
Extensions
Outputs
Organization Settings
Access Management
Billing
Platform Logs

Outputs [View Docs]

+ Add Outp

Outputs allows you to integrate data from LimaCharlie into other cloud tools. Stream sensor events, detections, and logs out to Slack, S3, GCS, SMTP, and more.

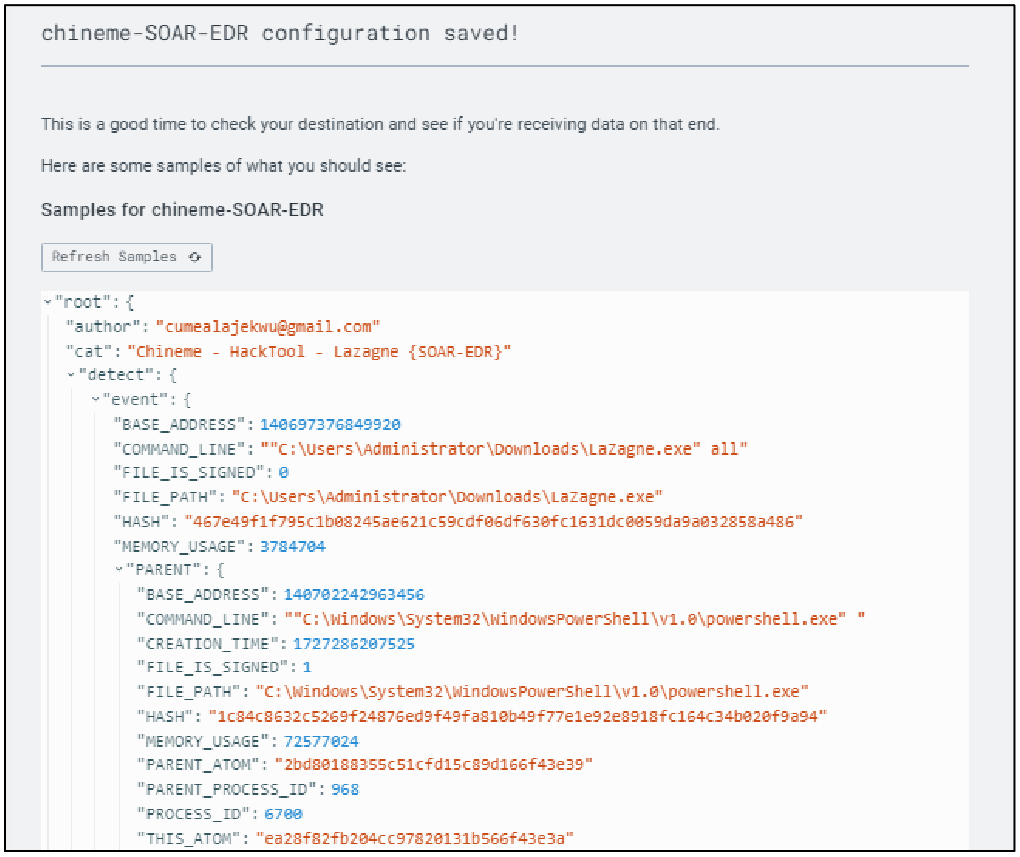
| Name | Stream | | Type | |
|---|---|---|---|---|
| chineme-SOAR-EDR | detect | → | tines | View Samples  Del |

▶ View System Configured Outputs

3. **Generating a Detection Event**:
   - To test if the connection is working, return to your **Windows Server VM** and run **Lasagna.exe** again:

```
.\Lasagna.exe
```

   - This will trigger a detection in LimaCharlie, which should be forwarded to **Tines** via the Web Hook.



chineme-SOAR-EDR configuration saved!

This is a good time to check your destination and see if you're receiving data on that end.

Here are some samples of what you should see:

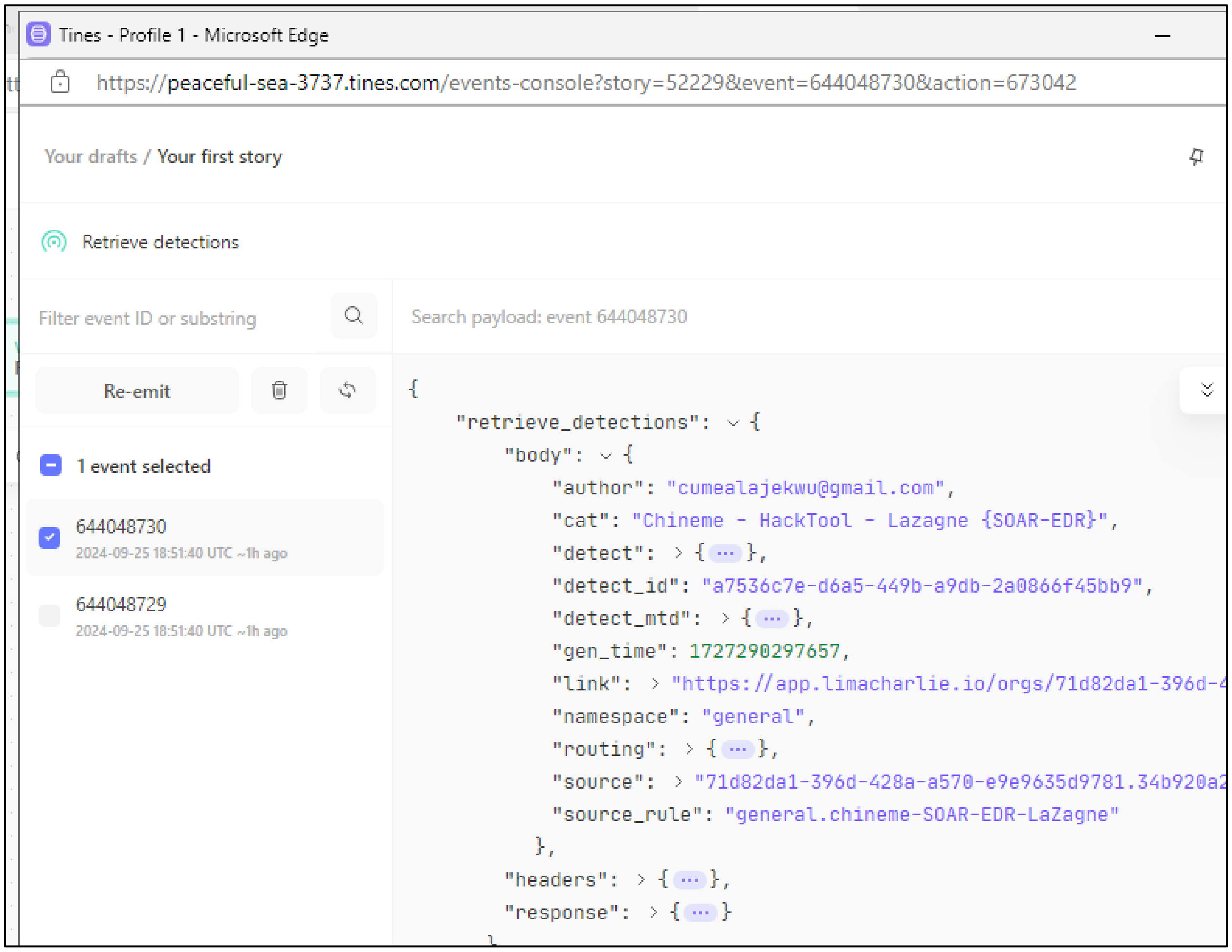
Samples for chineme-SOAR-EDR

```
"root": {
  "author": "cumealajekwu@gmail.com"
  "cat": "Chineme - HackTool - Lazagne {SOAR-EDR}"
  "detect": {
    "event": {
      "BASE_ADDRESS": 140697376849920
      "COMMAND_LINE": ""C:\Users\Administrator\Downloads\LaZagne.exe" all"
      "FILE_IS_SIGNED": 0
      "FILE_PATH": "C:\Users\Administrator\Downloads\LaZagne.exe"
      "HASH": "467e49f1f795c1b08245ae621c59cdf06df630fc1631dc0059da9a032858a486"
      "MEMORY_USAGE": 3784704
      "PARENT": {
        "BASE_ADDRESS": 140702242963456
        "COMMAND_LINE": ""C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" "
        "CREATION_TIME": 1727286207525
        "FILE_IS_SIGNED": 1
        "FILE_PATH": "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
        "HASH": "1c84c8632c5269f24876ed9f49fa810b49f77e1e92e8918fc164c34b020f9a94"
        "MEMORY_USAGE": 72577024
        "PARENT_ATOM": "2bd80188355c51cfd15c89d166f43e39"
        "PARENT_PROCESS_ID": 968
        "PROCESS_ID": 6700
        "THIS_ATOM": "ea28f82fb204cc97820131b566f43e3a"
```

**Step 4: Verifying the Integration**

Now that the connection is configured, let's confirm that **Tines** is receiving detections from LimaCharlie and forwarding them to **Slack**.

1. **Check the Detection in Tines**:
   - Go back to **Tines** and check the **Retrieve Detections** Web Hook.
   - You should see the detection event from **LimaCharlie**. This will contain information like the **command line**, **file path**, **hash**, and **username** associated with the Lasagna execution.



Tines - Profile 1 - Microsoft Edge

https://peaceful-sea-3737.tines.com/events-console?story=52229&event=644048730&action=673042

Your drafts / Your first story

Retrieve detections

Filter event ID or substring

Search payload: event 644048730

Re-emit

1 event selected

644048730
2024-09-25 18:51:40 UTC ~1h ago

644048729
2024-09-25 18:51:40 UTC ~1h ago

```
{
    "retrieve_detections": ⌄ {
        "body": ⌄ {
            "author": "cumealajekwu@gmail.com",
            "cat": "Chineme - HackTool - Lazagne {SOAR-EDR}",
            "detect": › { ··· },
            "detect_id": "a7536c7e-d6a5-449b-a9db-2a0866f45bb9",
            "detect_mtd": › { ··· },
            "gen_time": 1727290297657,
            "link": › "https://app.limacharlie.io/orgs/71d82da1-396d-4
            "namespace": "general",
            "routing": › { ··· },
            "source": › "71d82da1-396d-428a-a570-e9e9635d9781.34b920a2
            "source_rule": "general.chineme-SOAR-EDR-LaZagne"
        },
        "headers": › { ··· },
        "response": › { ··· }
```

**Part 5: Building the Final Playbook with LimaCharlie, Tines, and Slack Integration**

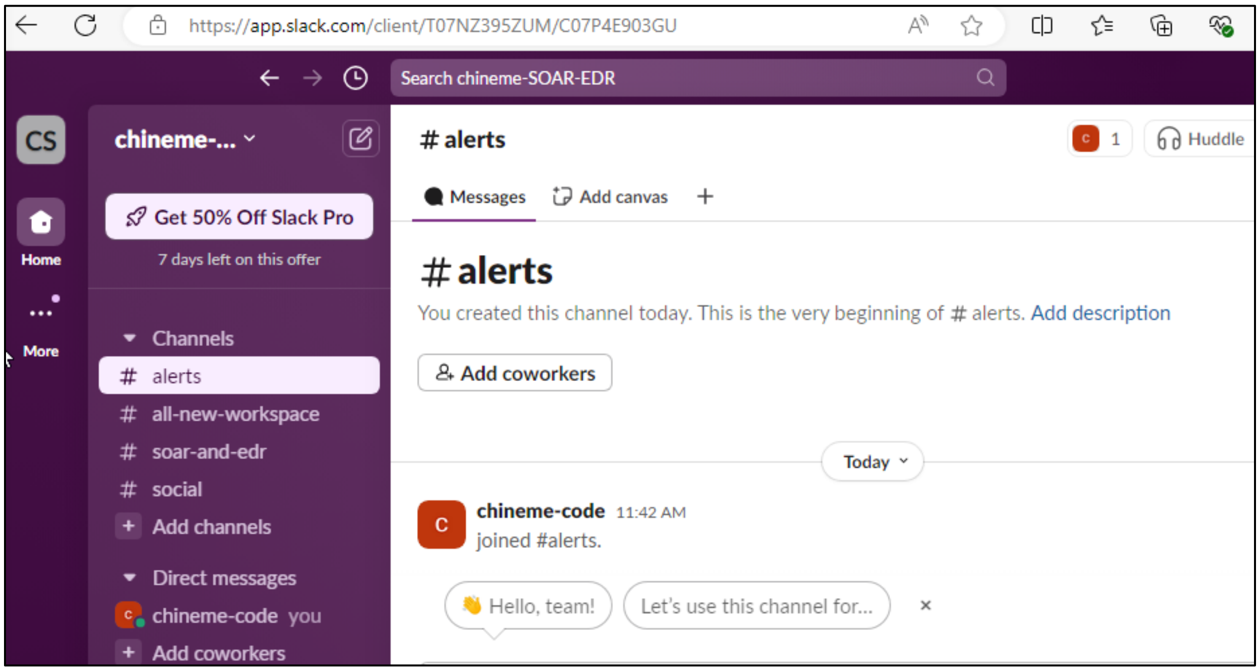
Today's objective is to build our **Playbook** (or story) that will:

1. **Send a message to Slack** with the details of the detection.
2. **Send an email** to notify the team.
3. **Prompt the user** to decide whether to isolate the affected machine.
4. **Isolate the machine** using LimaCharlie if the user chooses to do so.
5. **Send confirmation** of the machine's isolation status to Slack.

This is the culmination of all the steps we've been working on, and by the end of this section, you'll have a fully automated detection and response Playbook in **Tines** connected to **LimaCharlie**.

**Step 1: Connecting Slack to Tines**

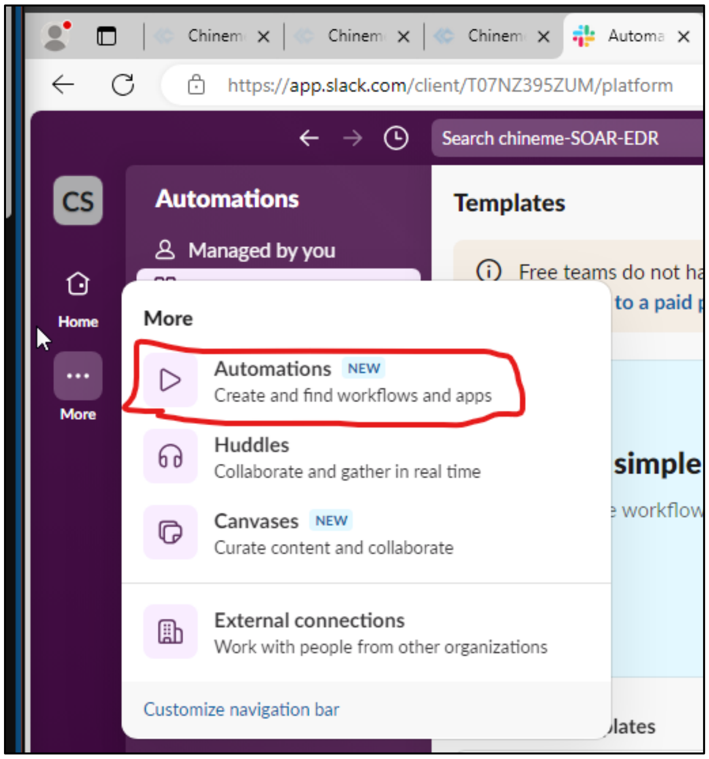
1. **Create Slack Workspace**:
   - Ensure you have your **Slack workspace** ready (as set up in Part 4).
   - You will also need a dedicated **alerts channel** where the messages from Tines will be sent.



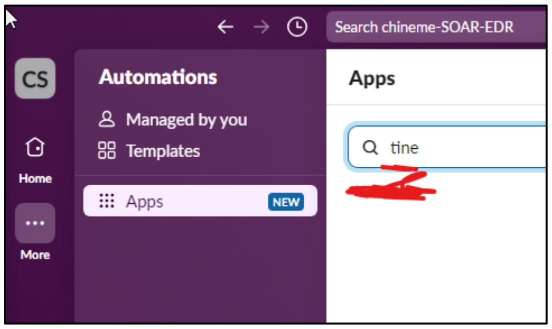
# alerts

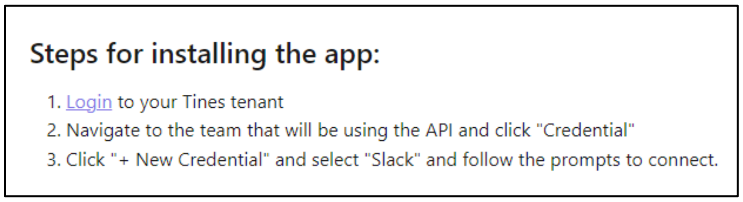
You created this channel today. This is the very beginning of # alerts. Add description

chineme-code 11:42 AM
joined #alerts.



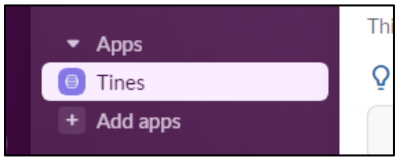
I already downloaded the tines app that's why I cant see it in my search



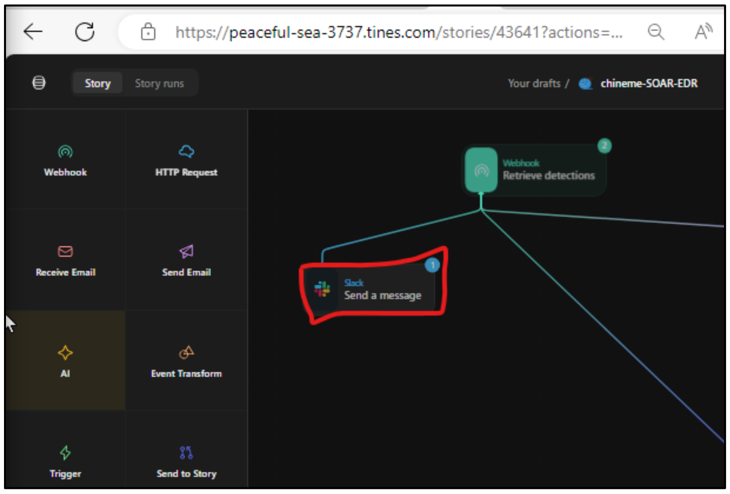


Steps for installing the app:

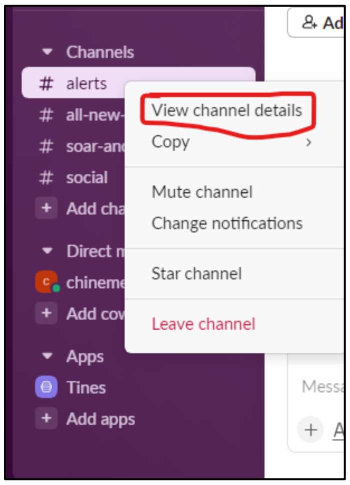
1. Login to your Tines tenant
2. Navigate to the team that will be using the API and click "Credential"
3. Click "+ New Credential" and select "Slack" and follow the prompts to connect.

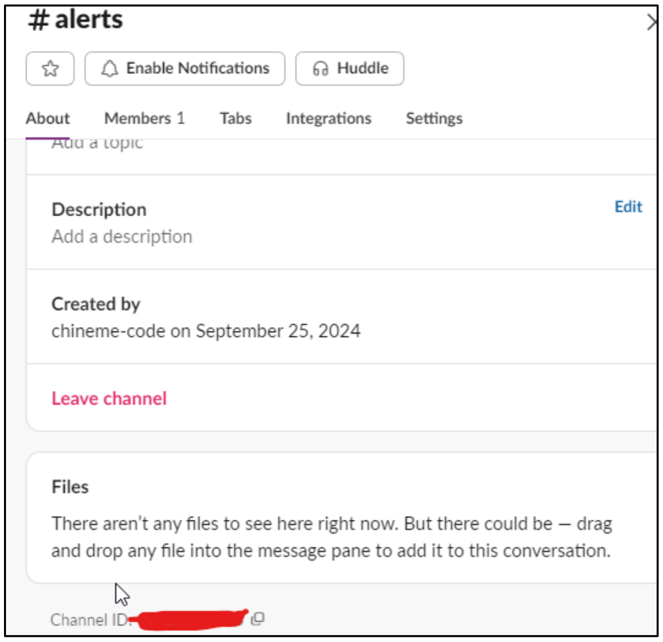


2. **Establish the Link Between Tines and Slack**:
   - In **Tines**, drag the **Slack** action from the template library into your Playbook.



- Use the Slack API to send a message to your **alerts** channel. The message should contain the details of the detection (such as the detection name, time, affected computer, and IP address).
- Use the Slack **Channel ID** to specify where the message should be sent.




# alerts

About · Members 1 · Tabs · Integrations · Settings

Add a topic

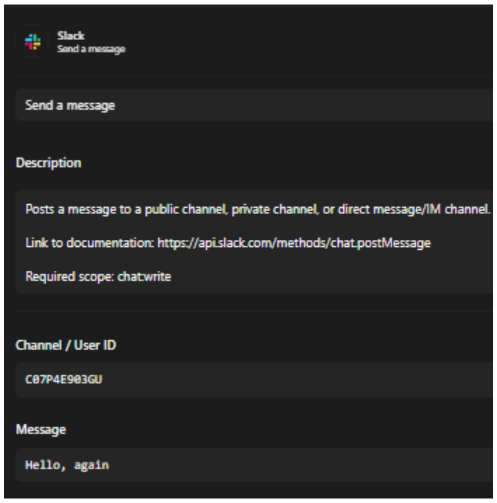
Description
Add a description

Created by
chineme-code on September 25, 2024

Leave channel

Files
There aren't any files to see here right now. But there could be — drag and drop any file into the message pane to add it to this conversation.

Channel ID: [redacted]


Slack
Send a message

Send a message

Description

Posts a message to a public channel, private channel, or direct message/IM channel.

Link to documentation: https://api.slack.com/methods/chat.postMessage

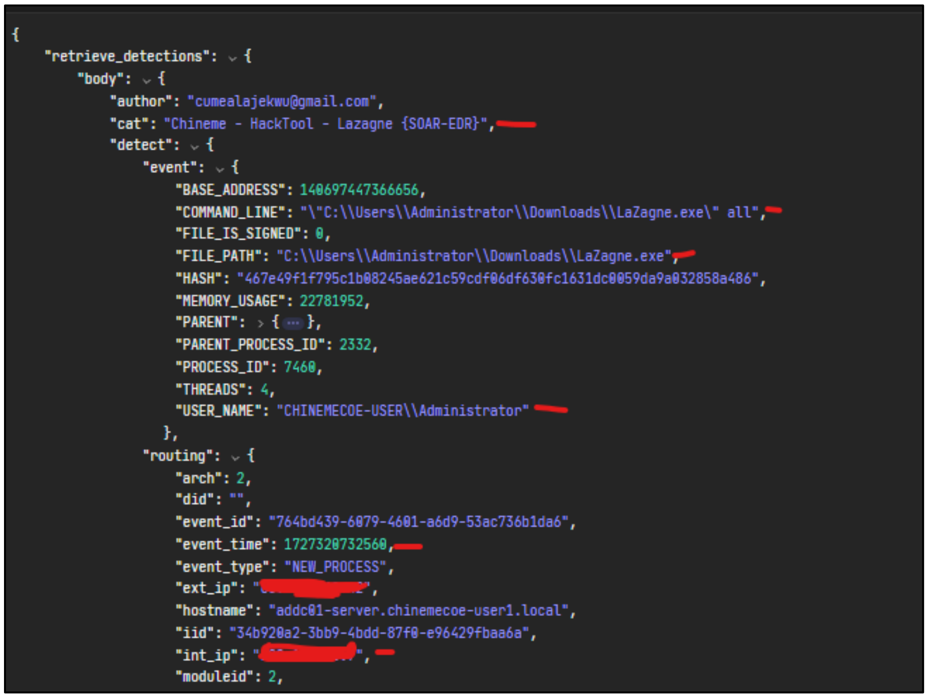
Required scope: chat:write

Channel / User ID

```
C07P4E903GU
```

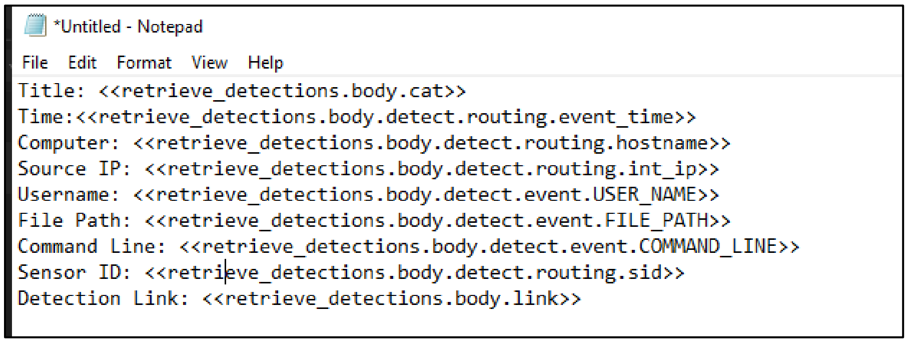
Message

```
Hello, again
```



```
{
    "retrieve_detections": ⌄ {
        "body": ⌄ {
            "author": "cumealajekwu@gmail.com",
            "cat": "Chineme - HackTool - Lazagne {SOAR-EDR}",
            "detect": ⌄ {
                "event": ⌄ {
                    "BASE_ADDRESS": 140697447366656,
                    "COMMAND_LINE": "\"C:\\Users\\Administrator\\Downloads\\LaZagne.exe\" all",
                    "FILE_IS_SIGNED": 0,
                    "FILE_PATH": "C:\\Users\\Administrator\\Downloads\\LaZagne.exe",
                    "HASH": "467e49f1f795c1b08245ae621c59cdf06df630fc1631dc0059da9a032858a486",
                    "MEMORY_USAGE": 22781952,
                    "PARENT": › { … },
                    "PARENT_PROCESS_ID": 2332,
                    "PROCESS_ID": 7460,
                    "THREADS": 4,
                    "USER_NAME": "CHINEMECOE-USER\\Administrator"
                },
                "routing": ⌄ {
                    "arch": 2,
                    "did": "",
                    "event_id": "764bd439-6079-4601-a6d9-53ac736b1da6",
                    "event_time": 1727320732560,
                    "event_type": "NEW_PROCESS",
                    "ext_ip": "[redacted]",
                    "hostname": "addc01-server.chinemecoe-user1.local",
                    "iid": "34b920a2-3bb9-4bdd-87f0-e96429fbaa6a",
                    "int_ip": "[redacted]",
                    "moduleid": 2,
```

Copy and paste on the notepad!

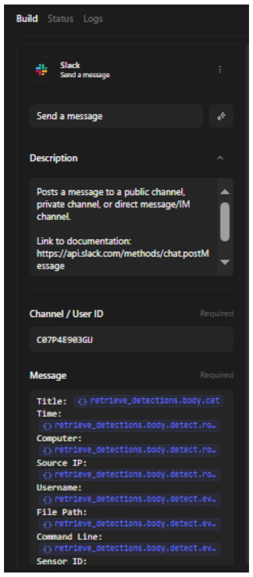


*Untitled - Notepad

File Edit Format View Help

```
Title: <<retrieve_detections.body.cat>>
Time:<<retrieve_detections.body.detect.routing.event_time>>
Computer: <<retrieve_detections.body.detect.routing.hostname>>
Source IP: <<retrieve_detections.body.detect.routing.int_ip>>
Username: <<retrieve_detections.body.detect.event.USER_NAME>>
File Path: <<retrieve_detections.body.detect.event.FILE_PATH>>
Command Line: <<retrieve_detections.body.detect.event.COMMAND_LINE>>
Sensor ID: <<retrieve_detections.body.detect.routing.sid>>
Detection Link: <<retrieve_detections.body.link>>
```

Paste in slack(tines)



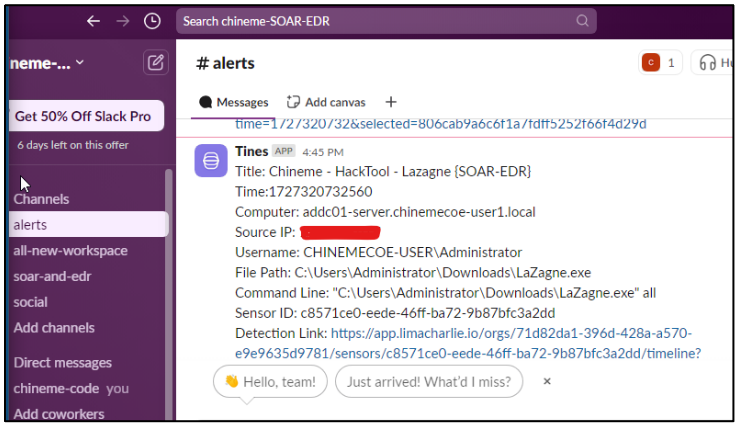
Posts a message to a public channel, private channel, or direct message/IM channel.

Link to documentation: https://api.slack.com/methods/chat.postMessage

C07P4E903GU

```
Title: retrieve_detections.body.cat
Time:
 retrieve_detections.body.detect.ro…
Computer:
 retrieve_detections.body.detect.ro…
Source IP:
 retrieve_detections.body.detect.ro…
Username:
 retrieve_detections.body.detect.ev…
File Path:
 retrieve_detections.body.detect.ev…
Command Line:
 retrieve_detections.body.detect.ev…
Sensor ID:
```

Run test. Here is the result of the test

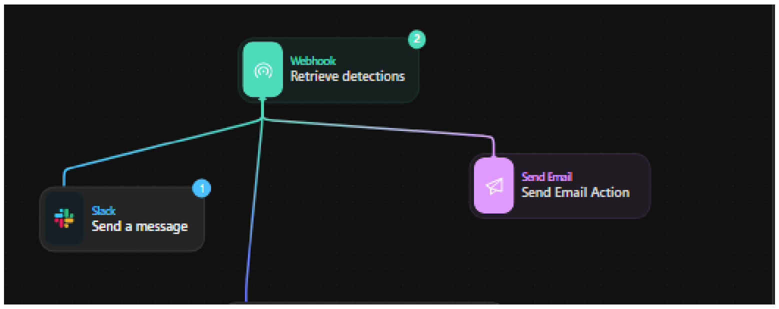


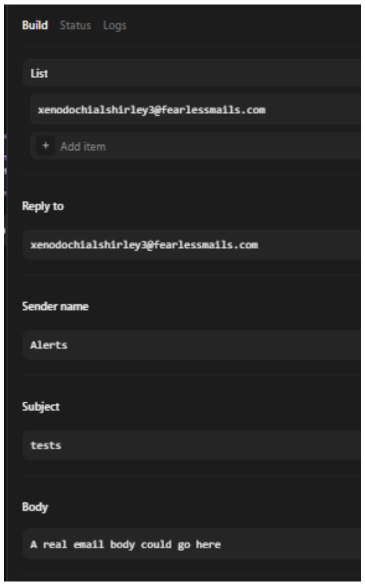
time=1727320732&selected=806cab9a6c6f1a7fdff5252f66f4d29d

Tines APP 4:45 PM
Title: Chineme - HackTool - Lazagne (SOAR-EDR)
Time:1727320732560
Computer: addc01-server.chinemecoe-user1.local
Source IP: [redacted]
Username: CHINEMECOE-USER\Administrator
File Path: C:\Users\Administrator\Downloads\LaZagne.exe
Command Line: "C:\Users\Administrator\Downloads\LaZagne.exe" all
Sensor ID: c8571ce0-eede-46ff-ba72-9b87bfc3a2dd
Detection Link: https://app.limacharlie.io/orgs/71d82da1-396d-428a-a570-e9e9635d9781/sensors/c8571ce0-eede-46ff-ba72-9b87bfc3a2dd/timeline?

## Step 2: Configuring Email Notifications in Tines

1. **Set up the Email Notification**:
   - Add an **email notification** action in Tines to send the detection details via email.
   - Use your organization's email system or a disposable email provider (like **Squarex**) to test.
   - Ensure the email includes relevant information such as the **detection name**, **computer name**, **IP address**, and a **link to the detection** in LimaCharlie.





List: xenodochialshirley3@fearlessmails.com

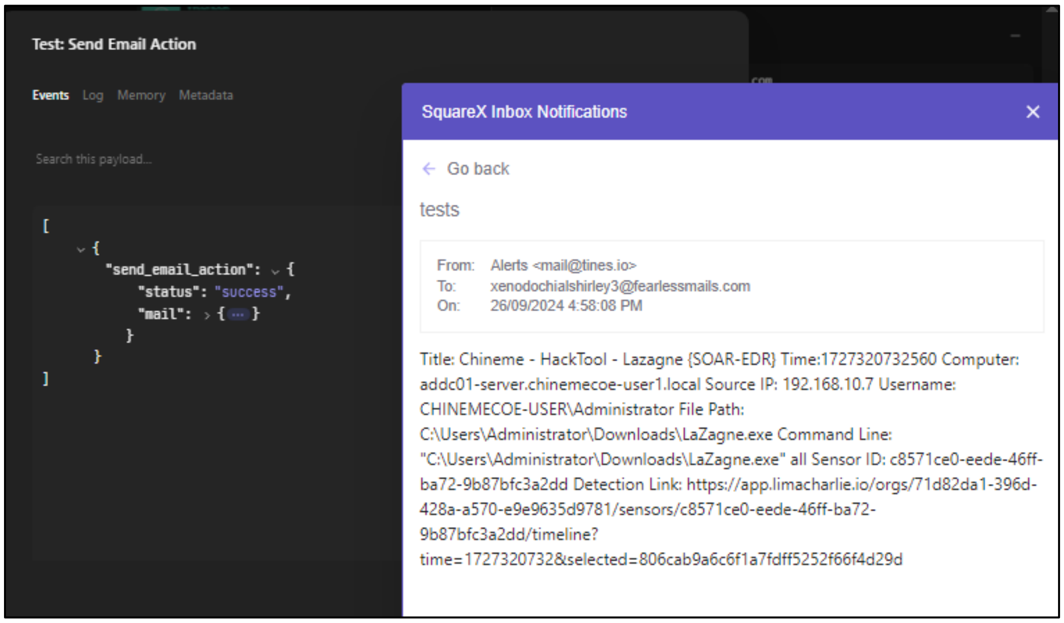
Reply to: xenodochialshirley3@fearlessmails.com

Sender name: Alerts

Subject: tests

Body: A real email body could go here

Use the same step as step 3. Just copy and paste the message you want to receive in your email.



```
[
  {
    "send_email_action": {
      "status": "success",
      "mail": { … }
    }
  }
]
```

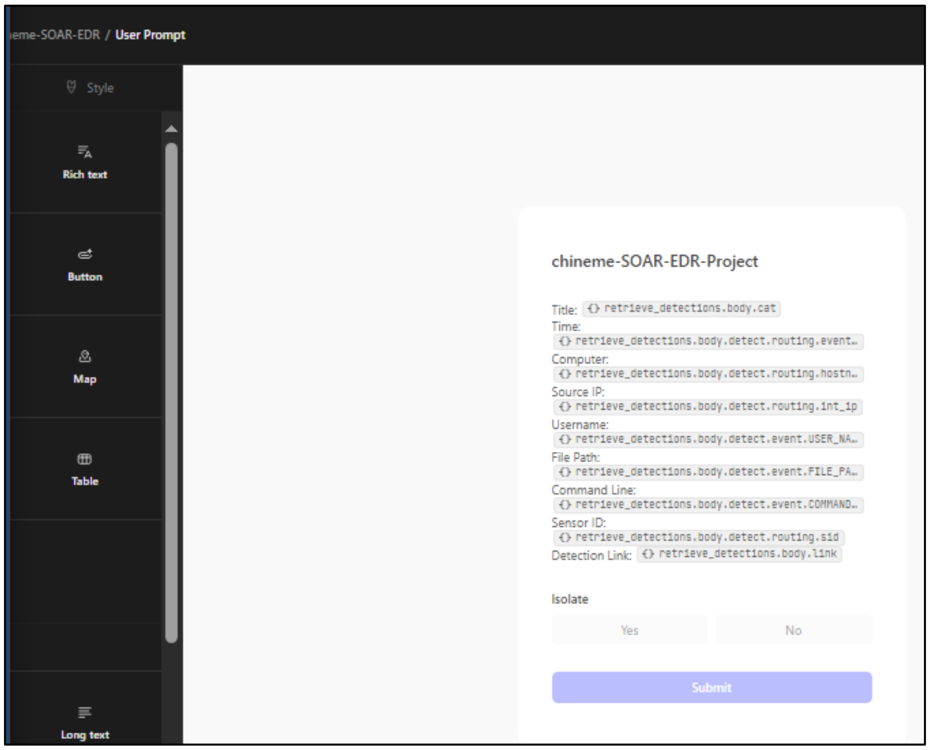
tests

From: Alerts <mail@tines.io>
To: xenodochialshirley3@fearlessmails.com
On: 26/09/2024 4:58:08 PM

Title: Chineme - HackTool - Lazagne {SOAR-EDR} Time:1727320732560 Computer: addc01-server.chinemecoe-user1.local Source IP: 192.168.10.7 Username: CHINEMECOE-USER\Administrator File Path: C:\Users\Administrator\Downloads\LaZagne.exe Command Line: "C:\Users\Administrator\Downloads\LaZagne.exe" all Sensor ID: c8571ce0-eede-46ff-ba72-9b87bfc3a2dd Detection Link: https://app.limacharlie.io/orgs/71d82da1-396d-428a-a570-e9e9635d9781/sensors/c8571ce0-eede-46ff-ba72-9b87bfc3a2dd/timeline?time=1727320732&selected=806cab9a6c6f1a7fdff5252f66f4d29d

**Step 3: Adding a User Prompt for Isolation Decision**

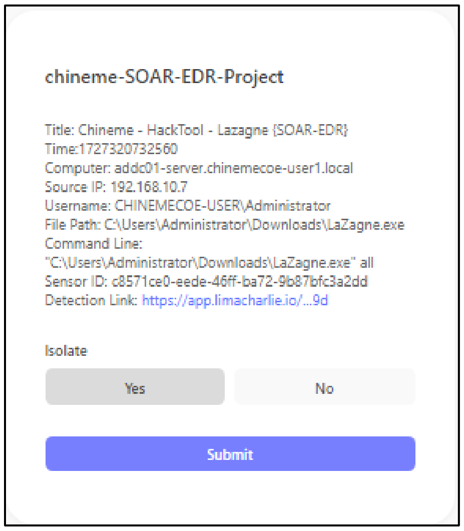
1. **Create a User Prompt Page**:
   - Now, create a **User Prompt Page** in Tines that asks the user whether they want to isolate the machine.
   - The page should display the detection details for the user to review before making the decision.
2. **Yes/No Decision**:
   - Include **Yes** and **No** buttons on the page.
   - If the user selects **Yes**, the Playbook should continue to isolate the machine using **LimaCharlie**. If the user selects **No**, a message should be sent to Slack indicating that the machine was not isolated.



chineme-SOAR-EDR-Project

Title: {} retrieve_detections.body.cat
Time:
{} retrieve_detections.body.detect.routing.event…
Computer:
{} retrieve_detections.body.detect.routing.hostn…
Source IP:
{} retrieve_detections.body.detect.routing.int_ip
Username:
{} retrieve_detections.body.detect.event.USER_NA…
File Path:
{} retrieve_detections.body.detect.event.FILE_PA…
Command Line:
{} retrieve_detections.body.detect.event.COMMAND…
Sensor ID:
{} retrieve_detections.body.detect.routing.sid
Detection Link: {} retrieve_detections.body.link

Isolate

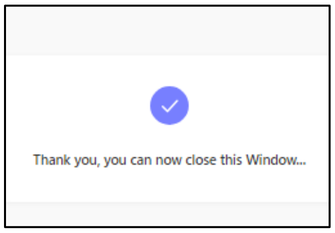


chineme-SOAR-EDR-Project

Title: Chineme - HackTool - Lazagne {SOAR-EDR}
Time:1727320732560
Computer: addc01-server.chinemecoe-user1.local
Source IP: 192.168.10.7
Username: CHINEMECOE-USER\Administrator
File Path: C:\Users\Administrator\Downloads\LaZagne.exe
Command Line:
"C:\Users\Administrator\Downloads\LaZagne.exe" all
Sensor ID: c8571ce0-eede-46ff-ba72-9b87bfc3a2dd
Detection Link: https://app.limacharlie.io/...9d

Isolate

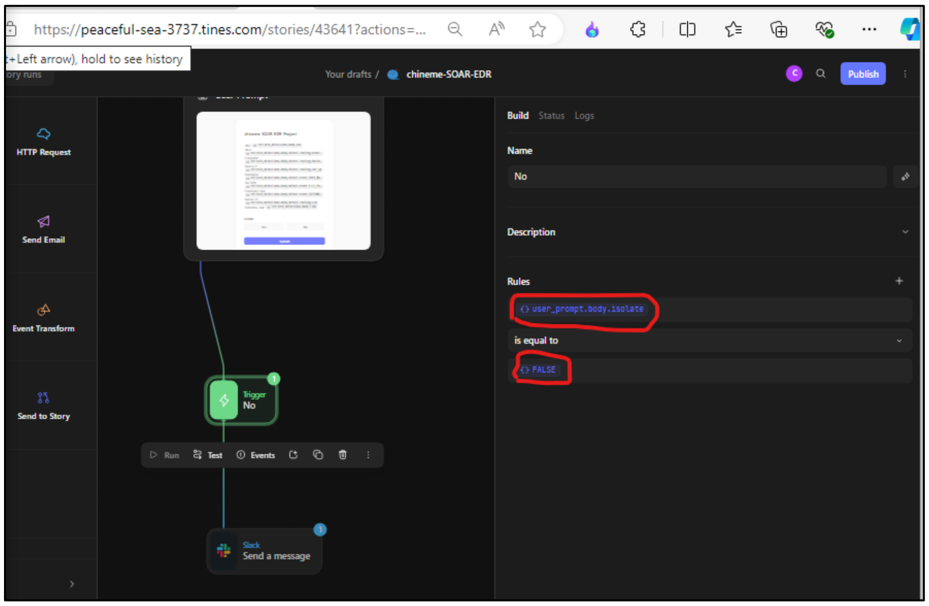
Yes [✓ No]



Thank you, you can now close this Window...

Configuring the workflow so that when you say no, it shows you a message and then sends you a slack message

This can be done with this configuration

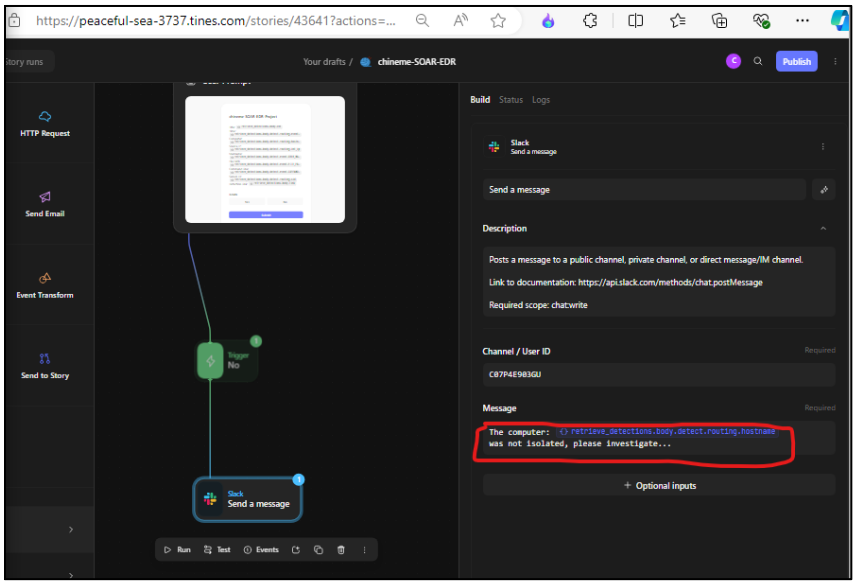


https://peaceful-sea-3737.tines.com/stories/43641?actions=...
+Left arrow), hold to see history
Your drafts / chineme-SOAR-EDR
Publish
HTTP Request
Send Email
Event Transform
Send to Story
Trigger
No
Run
Test
Events
Slack
Send a message
Build
Status
Logs

Name
No

Description

Rules
user_prompt.body.isolate
is equal to
FALSE



Slack
Send a message

Send a message

Description

Posts a message to a public channel, private channel, or direct message/IM channel.

Link to documentation: https://api.slack.com/methods/chat.postMessage

Required scope: chat:write

Channel / User ID — Required

```
C07P4E903GU
```

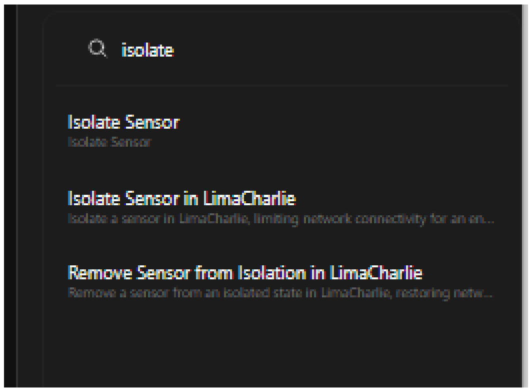
Message — Required

```
The computer: retrieve_detections.body.detect.routing.hostname
was not isolated, please investigate...
```

## Step 4: Isolating the Machine in LimaCharlie

1. **Automate Machine Isolation**:
    - If the user chooses to isolate the machine, configure Tines to send an **isolation request** to LimaCharlie.

If you are in doubt what to put in the use the search bar ad type in "isolate"



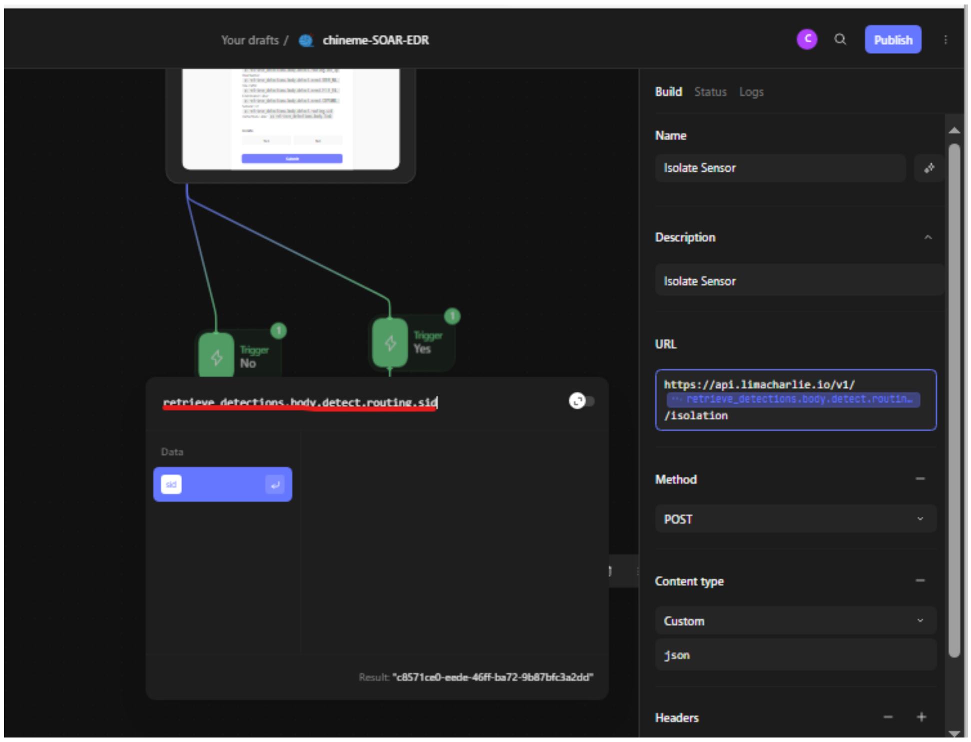
isolate

Isolate Sensor
Isolate Sensor

Isolate Sensor in LimaCharlie
Isolate a sensor in LimaCharlie, limiting network connectivity for an en...

Remove Sensor from Isolation in LimaCharlie
Remove a sensor from an isolated state in LimaCharlie, restoring netw...

- Use the **Sensor ID** (which you'll get from the detection event) to identify the machine to be isolated.



Name: Isolate Sensor

Description: Isolate Sensor

URL:
```
https://api.limacharlie.io/v1/
retrieve_detections.body.detect.routin…
/isolation
```

Method: POST

Content type: Custom

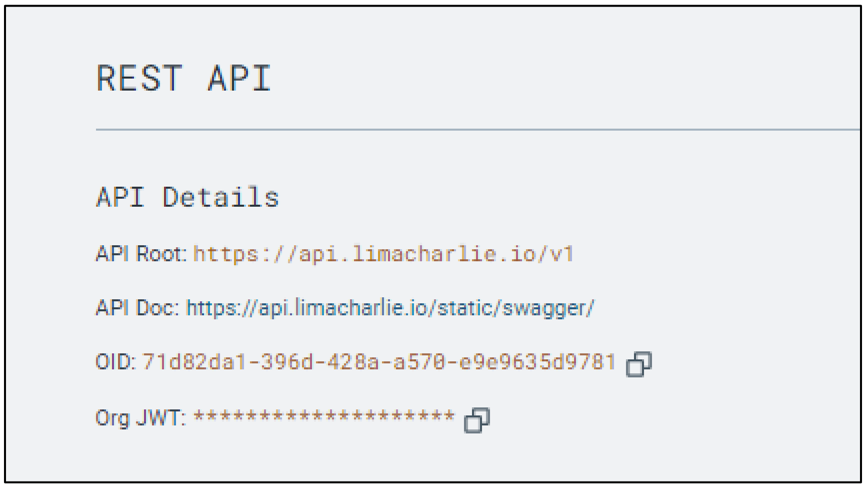
json

```
retrieve_detections.body.detect.routing.sid
```

Result: "c8571ce0-eede-46ff-ba72-9b87bfc3a2dd"

- Use the **LimaCharlie API** to issue the isolation command.

Under access management in limacharlie you'd click on REST API and copy Org JWT because that is what limacharlie documents suggested to be safer.



REST API

API Details

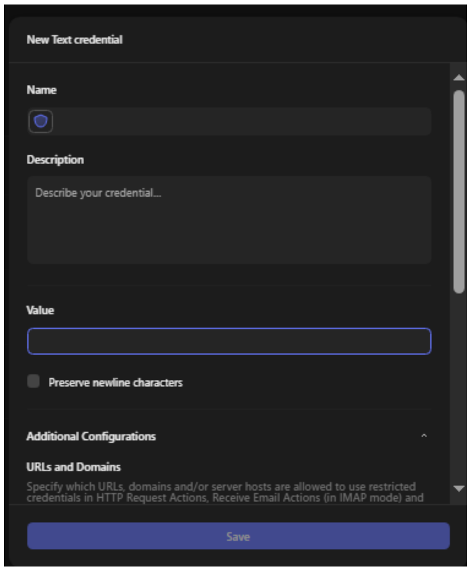
API Root: `https://api.limacharlie.io/v1`

API Doc: https://api.limacharlie.io/static/swagger/

OID: `71d82da1-396d-428a-a570-e9e9635d9781`

Org JWT: ********************

We would create a credential to add to tines so the limaCharlie can link and isolate the machine. In the Value box, that's where to post the API



New Text credential

Name

Description

Describe your credential...

Value

Preserve newline characters

Additional Configurations

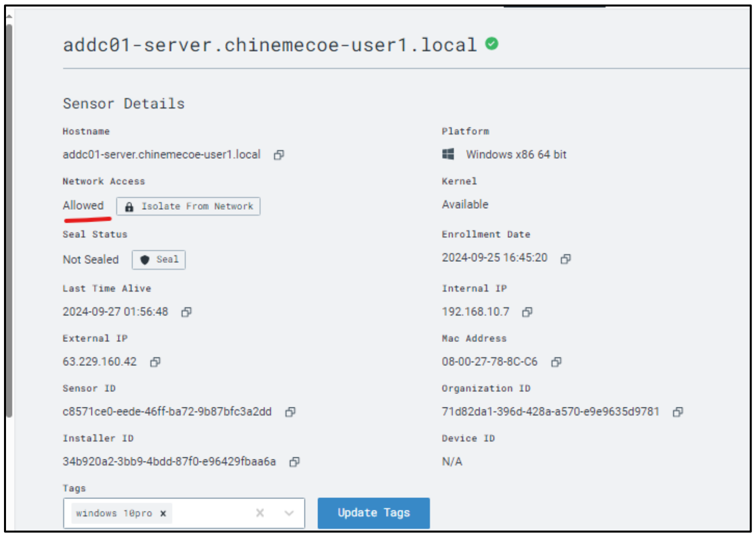
URLs and Domains

Specify which URLs, domains and/or server hosts are allowed to use restricted credentials in HTTP Request Actions, Receive Email Actions (in IMAP mode) and

Save

2. **Verify Isolation**:
   - After issuing the isolation command, verify that the machine has been successfully isolated by checking the **LimaCharlie dashboard**.

Before we Isolated



addc01-server.chinemecoe-user1.local

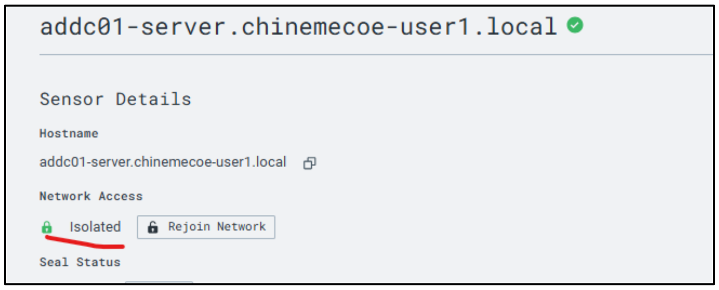
Sensor Details

| | |
|---|---|
| Hostname | Platform |
| addc01-server.chinemecoe-user1.local | Windows x86 64 bit |
| Network Access | Kernel |
| Allowed Isolate From Network | Available |
| Seal Status | Enrollment Date |
| Not Sealed Seal | 2024-09-25 16:45:20 |
| Last Time Alive | Internal IP |
| 2024-09-27 01:56:48 | 192.168.10.7 |
| External IP | Mac Address |
| 63.229.160.42 | 08-00-27-78-8C-C6 |
| Sensor ID | Organization ID |
| c8571ce0-eede-46ff-ba72-9b87bfc3a2dd | 71d82da1-396d-428a-a570-e9e9635d9781 |
| Installer ID | Device ID |
| 34b920a2-3bb9-4bdd-87f0-e96429fbaa6a | N/A |

Tags

windows 10pro

Update Tags



addc01-server.chinemecoe-user1.local

Sensor Details

Hostname

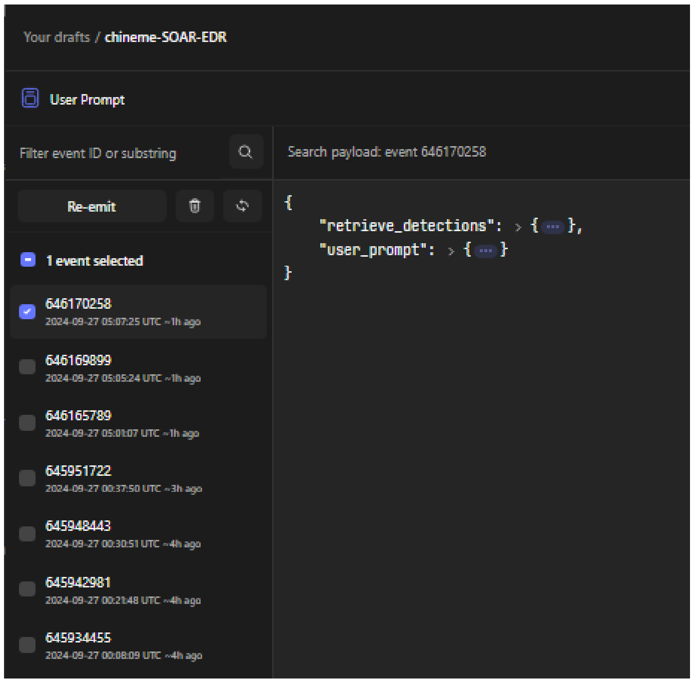
addc01-server.chinemecoe-user1.local

Network Access

Isolated Rejoin Network

Seal Status

N/B: To start another test click on re-emit in the events tag of user prompt. The ame thing goes for webhook as well. If you want to run another test.


Your drafts / chineme-SOAR-EDR
User Prompt
Filter event ID or substring
Search payload: event 646170258
Re-emit
1 event selected

```
{
    "retrieve_detections": > { ··· },
    "user_prompt": > { ··· }
}
```

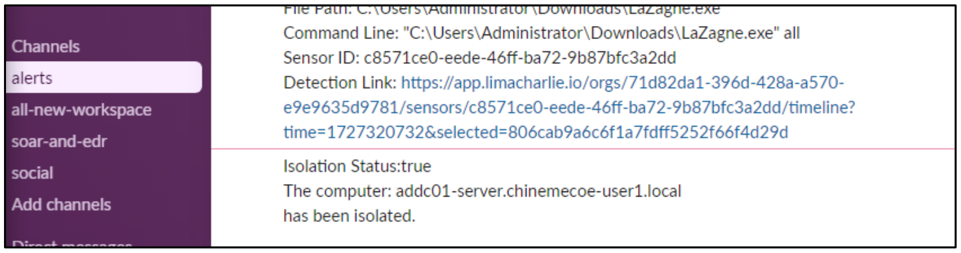
646170258
2024-09-27 05:07:25 UTC ~1h ago
646169899
2024-09-27 05:05:24 UTC ~1h ago
646165789
2024-09-27 05:01:07 UTC ~1h ago
645951722
2024-09-27 00:37:50 UTC ~3h ago
645948443
2024-09-27 00:30:51 UTC ~4h ago
645942981
2024-09-27 00:21:48 UTC ~4h ago
645934455
2024-09-27 00:08:09 UTC ~4h ago

### Step 5: Sending Confirmation to Slack

1. **Confirmation Message**:
   - Once the machine has been isolated, send a confirmation message to the Slack channel. This message should include the **isolation status** and the **computer name**.
2. **Customize the Message**:
   - Ensure the message is clear and formatted, for example:

The computer [Computer Name] has been isolated.

Isolation Status: Success



File Path: C:\Users\Administrator\Downloads\LaZagne.exe
Command Line: "C:\Users\Administrator\Downloads\LaZagne.exe" all
Sensor ID: c8571ce0-eede-46ff-ba72-9b87bfc3a2dd
Detection Link: https://app.limacharlie.io/orgs/71d82da1-396d-428a-a570-e9e9635d9781/sensors/c8571ce0-eede-46ff-ba72-9b87bfc3a2dd/timeline?time=1727320732&selected=806cab9a6c6f1a7fdff5252f66f4d29d

Isolation Status:true
The computer: addc01-server.chinemecoe-user1.local
has been isolated.

### Step 6: Testing the Entire Playbook

1. **Run the Playbook**:

    - Generate a **Lasagna.exe** detection on your Windows Server VM.
    - The detection should trigger the entire Playbook, sending a Slack message, an email, and a user prompt to isolate the machine.
2. **Check for Errors**:
    - Test all branches of the Playbook:
        - If the user chooses **No**, ensure the correct message is sent to Slack.
        - If the user chooses **Yes**, verify the machine is isolated and a confirmation message is sent.