**SOAR EDR PROJECT**

**By Adeen Mumtaz**

**Table of Contents**

|  |  |
| --- | --- |
| **1. OVERVIEW.........................................................................................................................** | 3 |
| **2. WORKFLOW.......................................................................................................................**  Project Steps...................................................................................................................................  Workflow Diagram…....................................................................................................................... | **3**  3  4 |
| **3. LIMACHARLIE INSTALLATION............................................................................................**  LimaCharlie Navigation and Overview...........................................................................................  Sensor Installation ………………………………........................................................................................  Navigating Sensor Information In LimaCharlie .............................................................................. | **4**  5  5  8 |
| **4. CREATING DETECTION AND RESPONSE RULE ....................................................................**  Lazagne…………………………………………..................................................................................................  Detection And Response Rule …………............................................................................................... | **12**  12  14 |
| **5. FORWARDING DETECTION TO TINES ……………………...........................................................**  Slack ..................................................................................................................................................  Setting Up Tines For Detection …..................................................................................................... | **16**  16  17 |
| **6. GENERATING PLAYBOOK FOR AUTOMATION ....................................................................**  Playbook ……………………………………....................................................................................................  Linking Slack ………………………………....................................................................................................  Linking Email ……………………………………..............................................................................................  Adding Detection Details ……………………...........................................................................................  User Prompt ……………………………….................................................................................................... | **23**  23  24  29  30  32 |
| **7. SUMMARY …........................................................................................................................** | **35** |

1. **OVERVIEW**

In this project, we will create an EDR using LimaCharlie and Tines. We will learn how to develop our own detection and response rules and create playbooks for automation.

What we will be doing:

1. Sending a Slack message and an email containing information about detecting malicious content. The detection will be made by LimaCharlie.
2. Tines will be generating a user prompt to either isolate the machine or not.
3. If the user selects YES then LimaCharlie will isolate the machine
4. **WORKFLOW**

**Project Steps**

* Create a detection in LimaCharlie-for example, hacktool detection, and then push it to Tines. Tines will then send a Slack message and email about detection.
* Slack messages and emails will contain the following information:

1. Time
2. Computer Name
3. Source IP
4. Process
5. Command Line
6. File Path
7. Sensor ID
8. Link to detection file (if applicable)

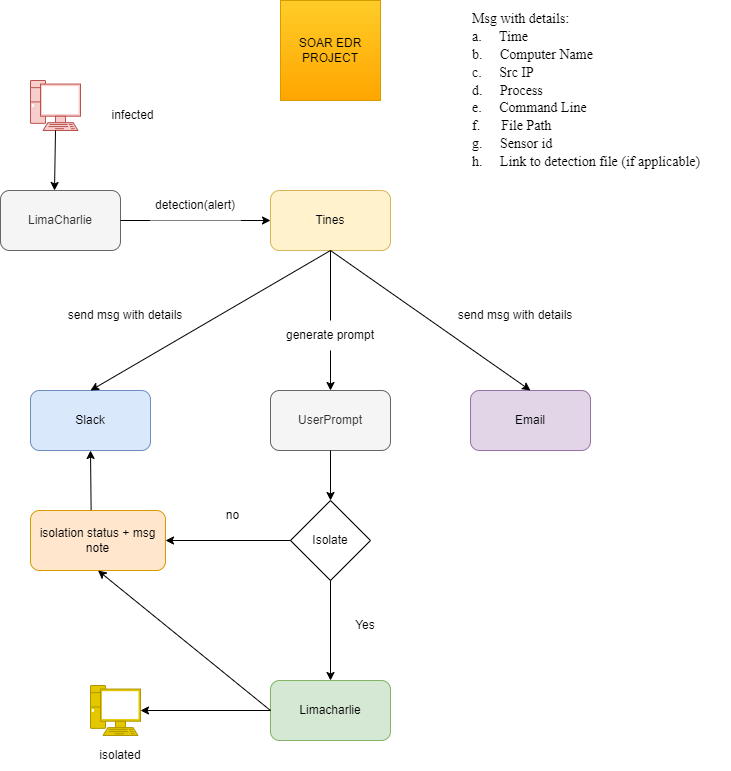
* Tines will then prompt the user to either isolate the machine or not
* If the user selects *YES*

then LimaCharlie should isolate the machine and send a message to the slack with isolation status and a note that “the computer <computername> has been isolated”

* If the user selects *NO*

then LimaCharlie will not isolate the machine and send a message to Slack with isolation status and a note that “the computer <computername> was not isolated”

**Workflow Diagram**



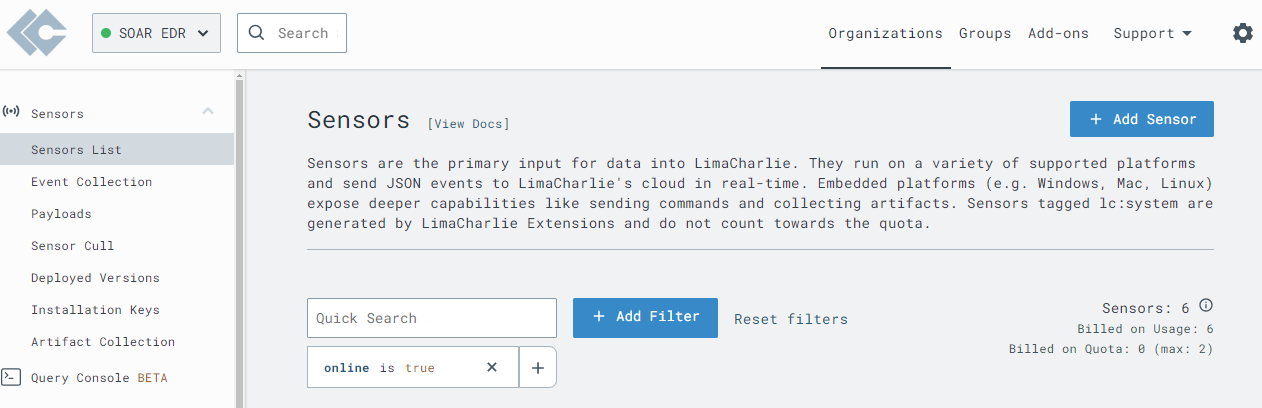
*Workflow Diagram*

1. **LimaCharlie Installation**

According to the documentation, “ LimaCharlie is a SecOps Cloud Platform that aims to deliver security operations for the modern era.”

LimaCharlie is a cybersecurity platform that provides Endpoint Detection and Response (EDR) capabilities as a service. It offers tools for security teams to monitor, detect, and respond to threats across their network. The platform is designed to be highly flexible and scalable, allowing users to integrate it into their existing security infrastructure and workflows.

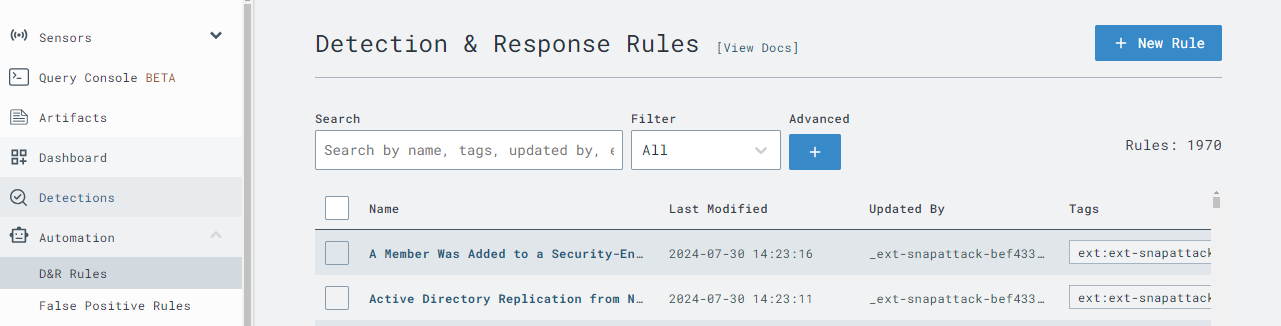
Create an account on LimaCharlie and create an organization (like a project) with the name “SOAR EDR”



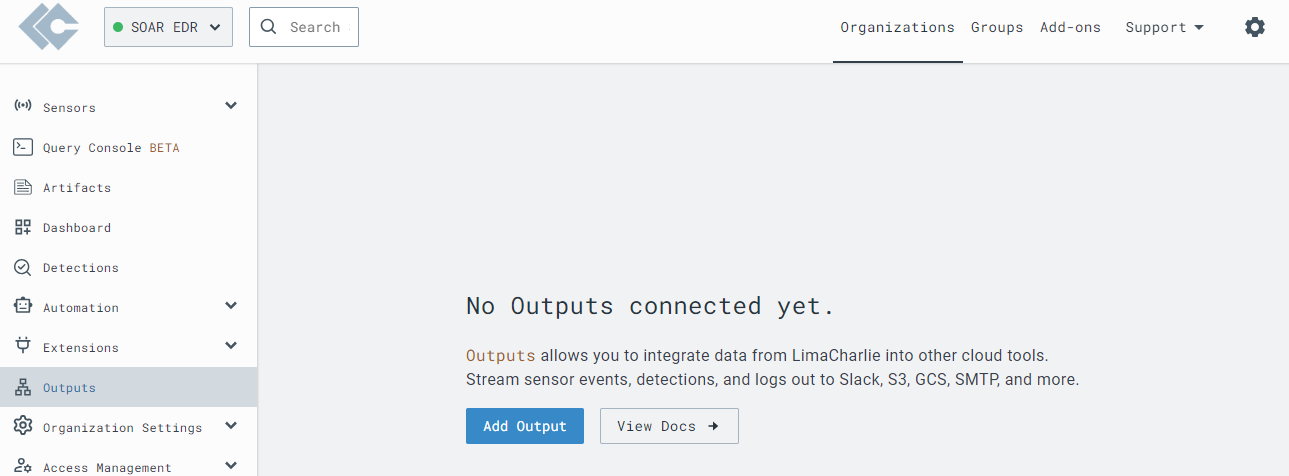
*LimaCharlie Installation*

**LimaCharlie Navigation and Overview**

We will be using detections and automation in our project frequently.

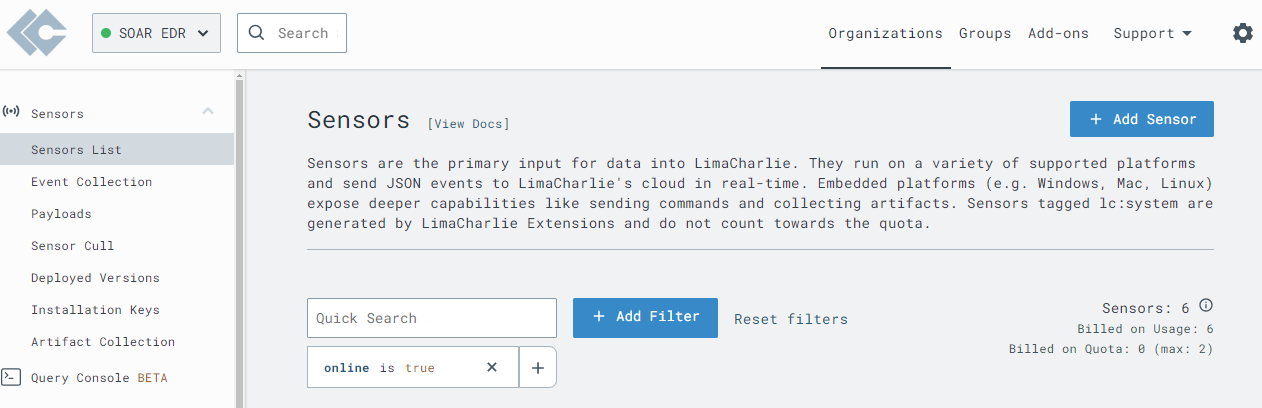


Outputs is where we will be sending detection out to Tines.

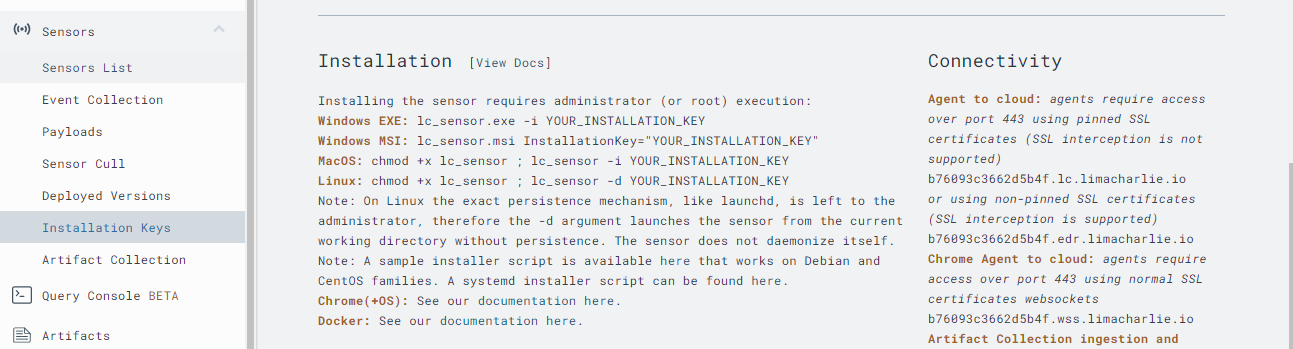


And we can navigate further and check out the other tabs as well.

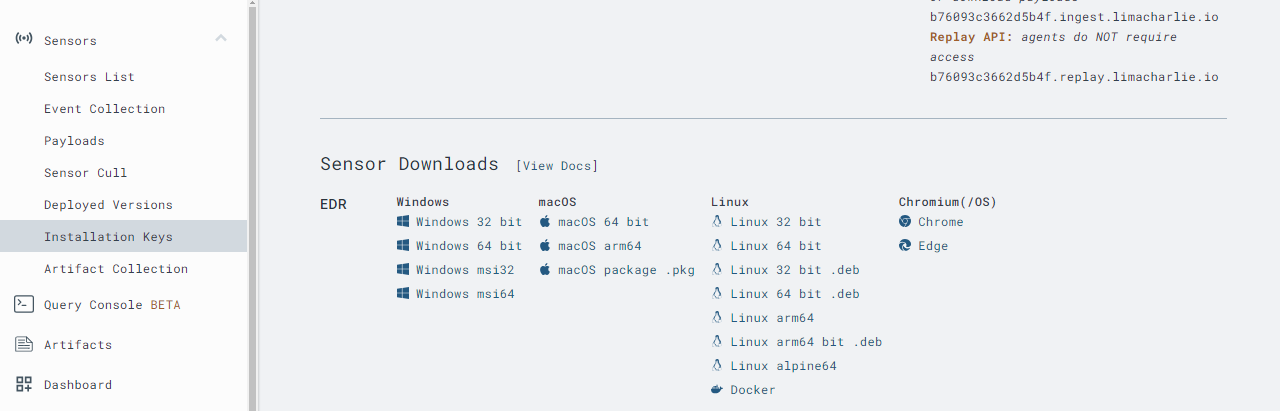
**Sensor Installation**

In LimaCharlie, "sensors" refer to software agents deployed on endpoints (e.g., desktops, laptops, servers) to monitor and collect telemetry data. These sensors play a crucial role in the platform’s ability to detect, analyze, and respond to security threats in real time. 

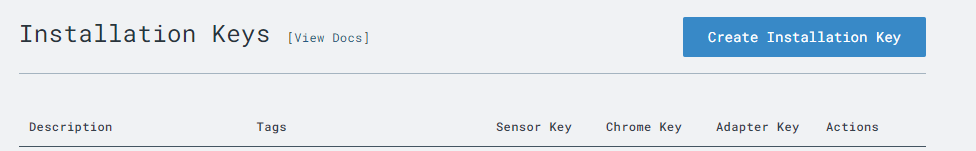
We will be using Windows 10 in our project. Download the windows sensor which will be used to integrate LimaCharlie into our environment

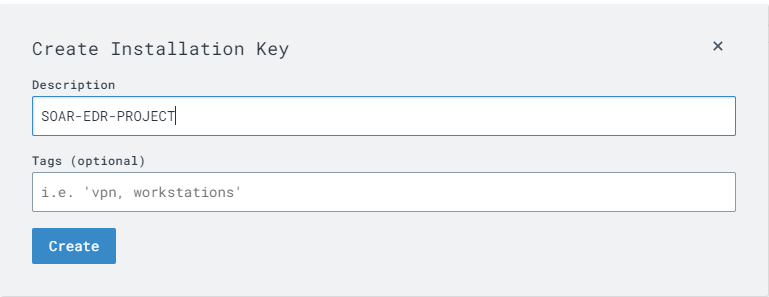


We are interested in EDR and that is where we will download the sensor file for Windows.

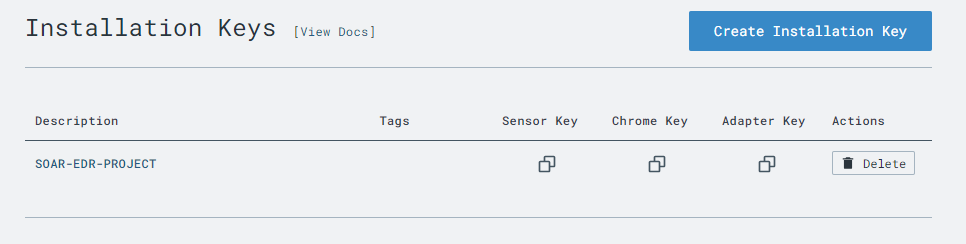


For the sensor, we need the installation key to enroll our Windows machine over to LimaCharlie. We will create a new installation key for our machine.



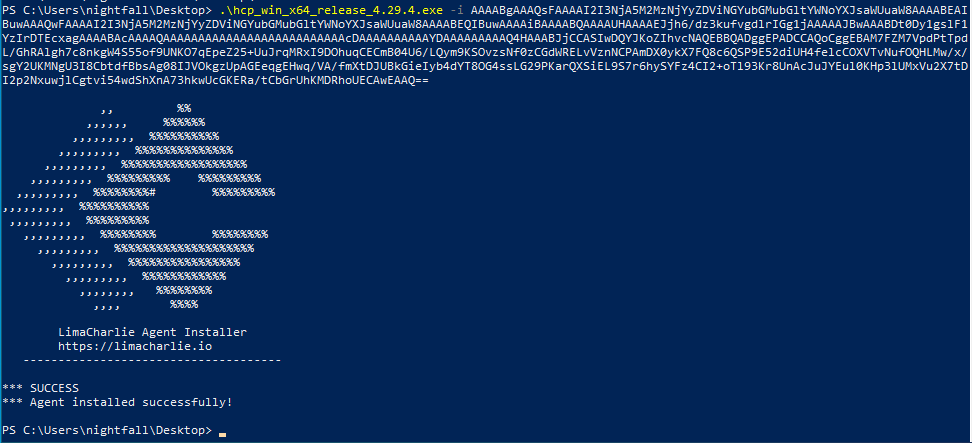


Copy the installation key as it will be used in the installation command for LimaCharlie.

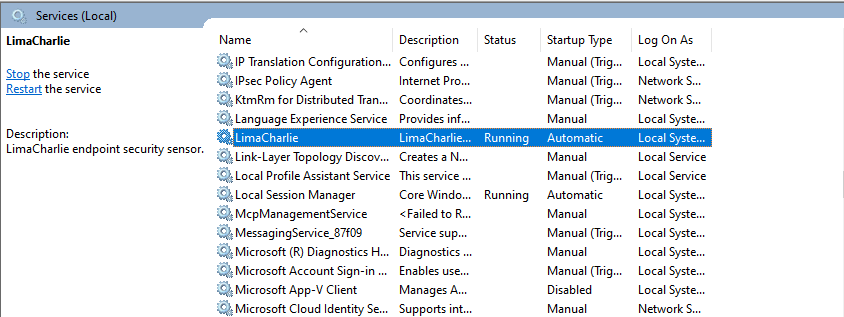


Use command

*#./file.exe -i installation key*

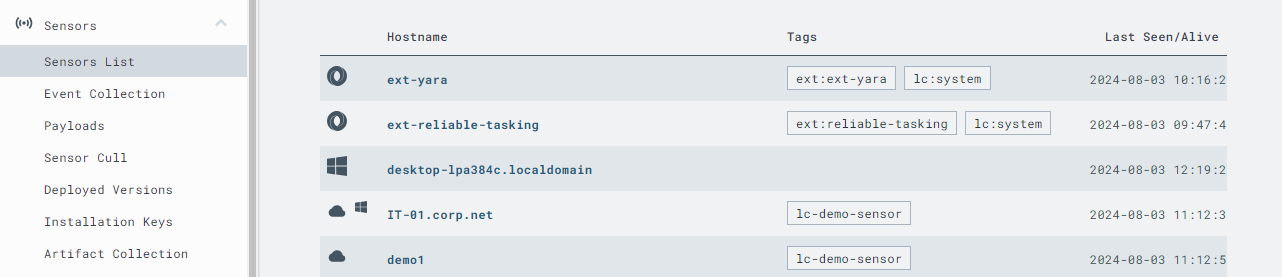


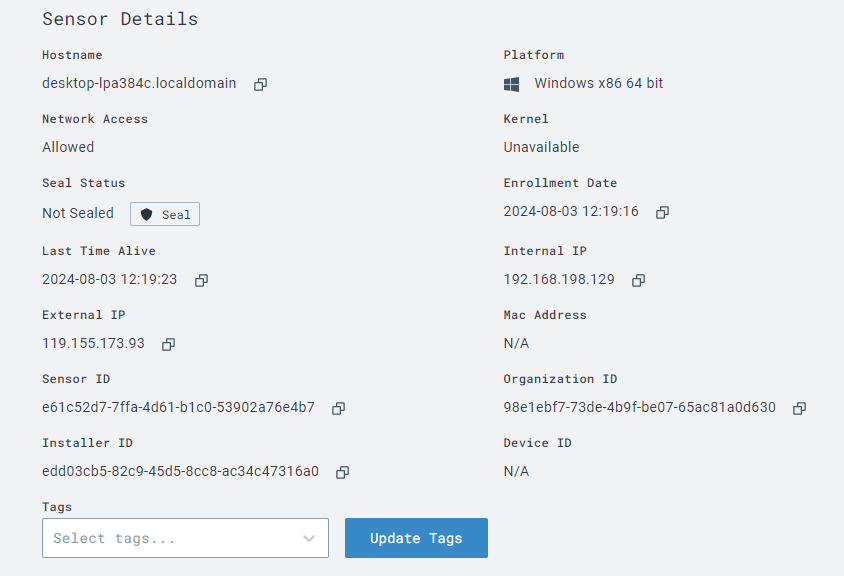
Head over to services to see if LimaCharlie is running on our Windows machine.



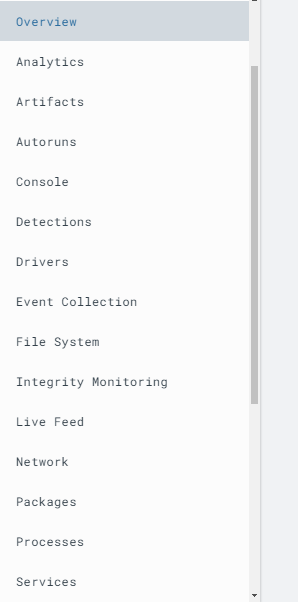
**Navigating Sensor Information In LimaCharlie**

Move to the sensor list under the sensor tab in LimaCharlie and locate the server/PC name.

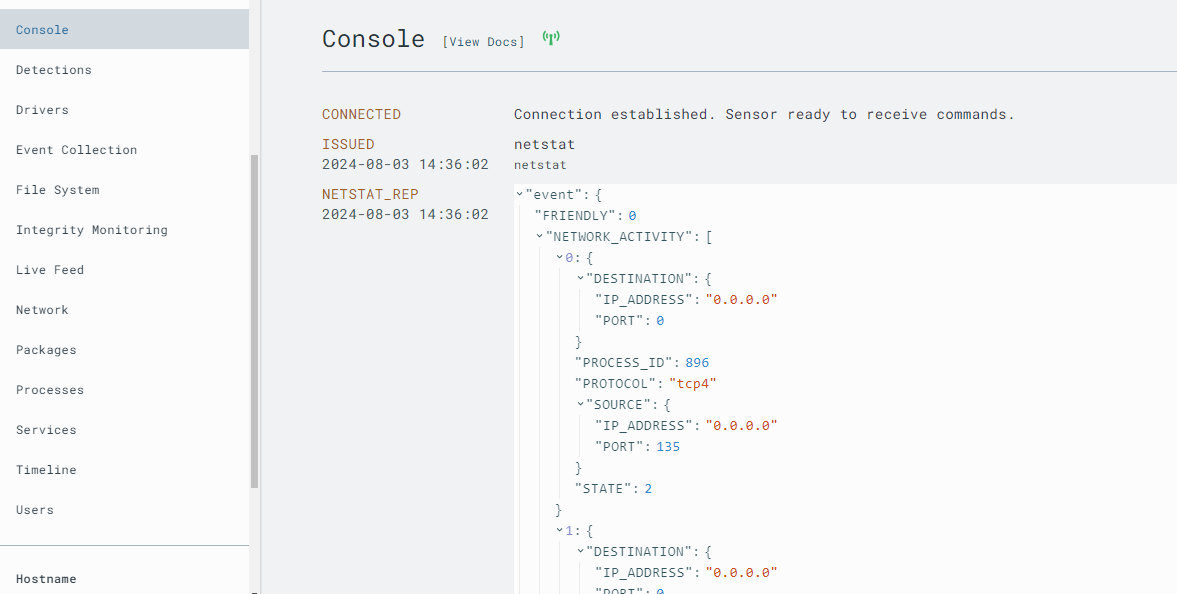




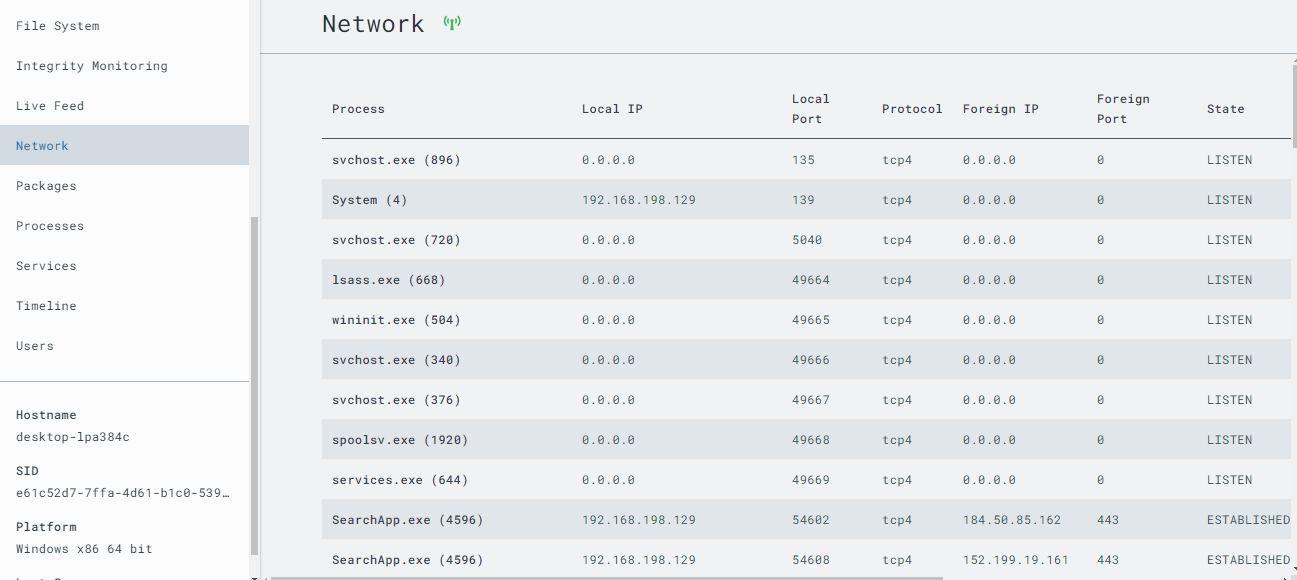
We can see a collection of information, most likely everything on our server or machine can be accessed and viewed via LimaCharlie



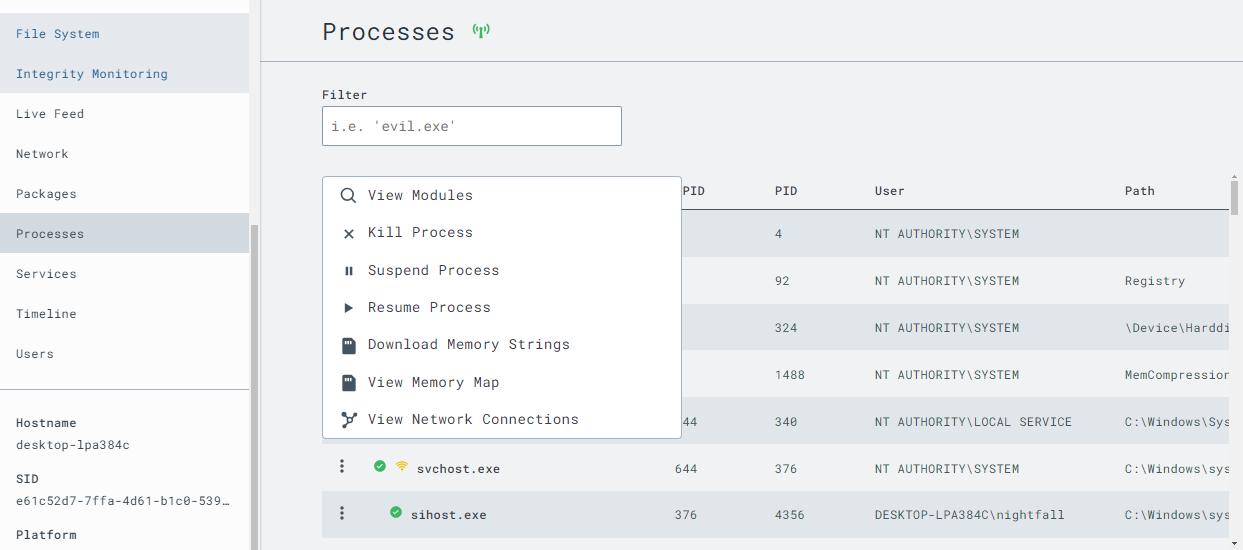
The console lets us execute commands as we execute them on our local machine.



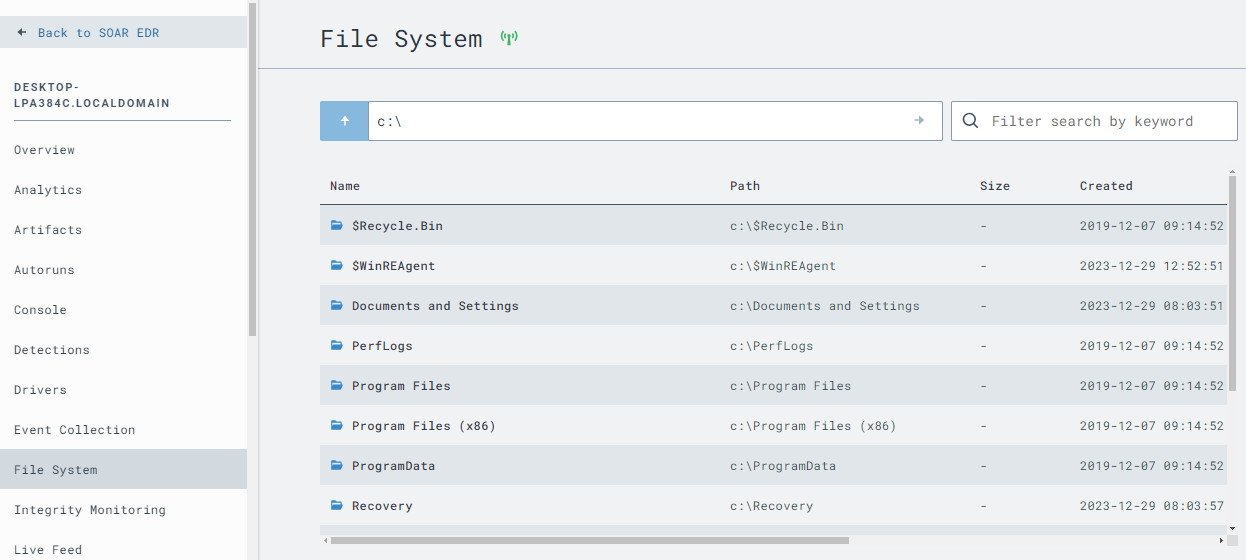
The network tab tells us about different network connections on our server.



All the processes running on the system can be viewed here and we can also kill, modify, and do various stuff here as well.



A great thing is that the complete file system can also be accessed here.



Other things like processes, services, timelines, users, etc can also be viewed and accessed on LimaCharlie.

1. **CREATING DETECTION AND RESPONSE RULE**

Now let’s move to our next step, which is to create a detection and response rule in LimaCharlie.

Firstly we generate telemetry (i.e.; lazagne) so that it can be detected by LimaCharlie and then send it over to Tines for automation.

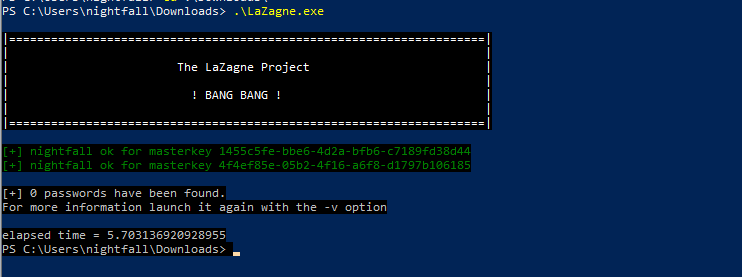
**LAZAGNE**

The LaZagne project is an open-source application used to retrieve passwords stored on a local computer. LaZagne can extract passwords from various software on different operating systems, including Windows, Linux, and macOS. The tool is often used by penetration testers and security researchers to demonstrate the risks associated with password storage practices on local machines.

Download Lazagne from GitHub using the link: <https://github.com/AlessandroZ/LaZagne/releases/>

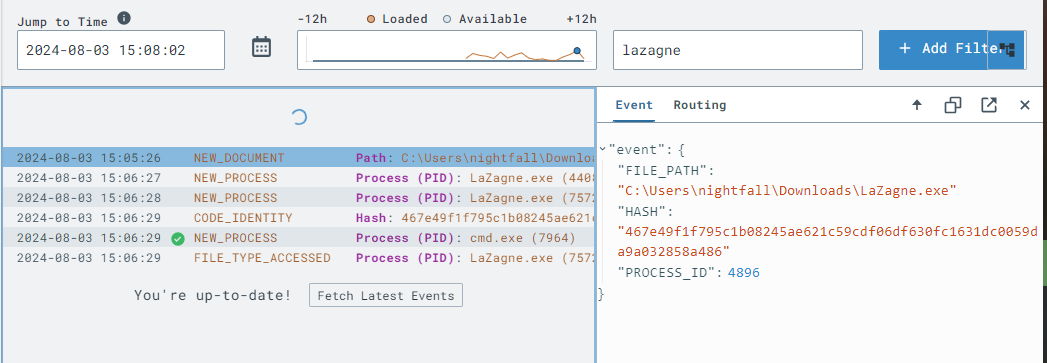
Make sure to turn the defender off when downloading.

Now execute it and then see what we get in LimaCharlie



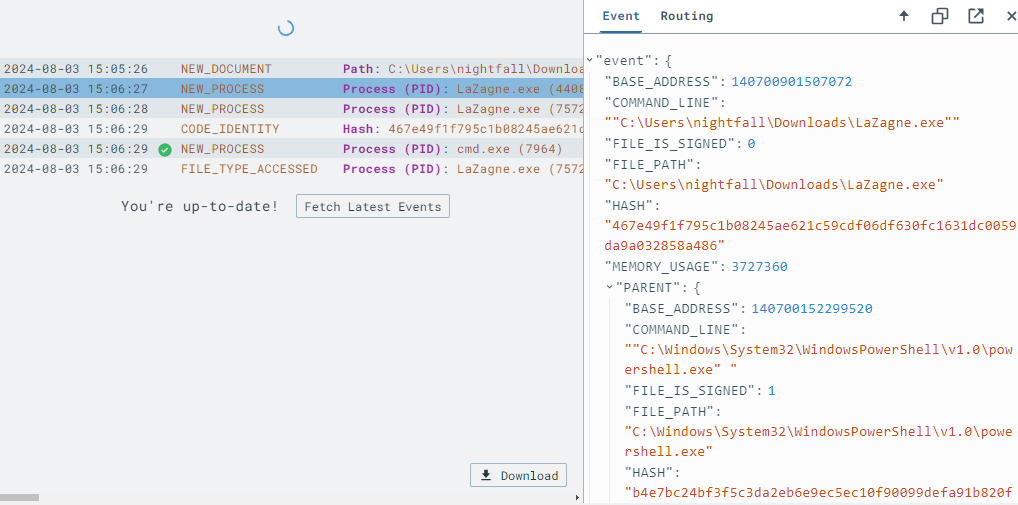
Let’s head to Timeline in LimaCharlie and view the events.

Here’s the download event.



Next we see the NEW-PROCESS event that is generated when we executed the Lazagne.exe file.

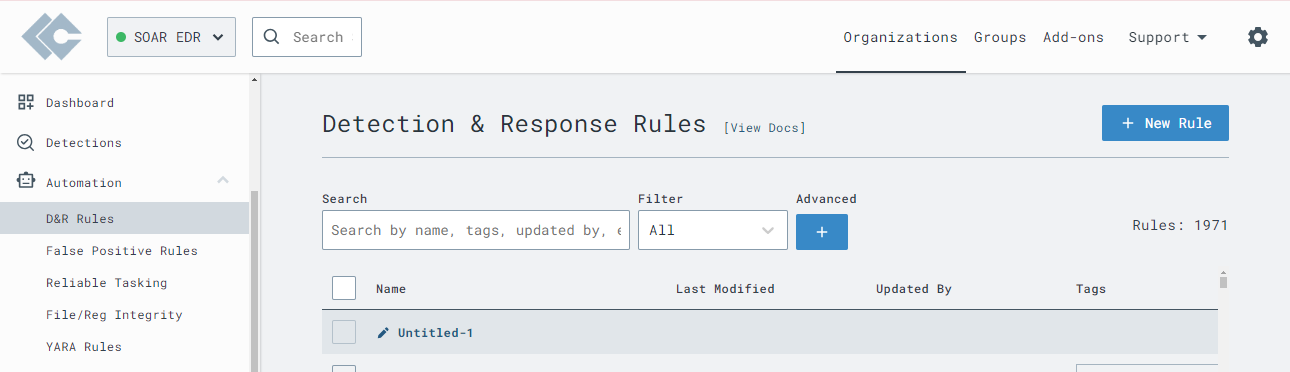
Let’s head over to the event and check out the information



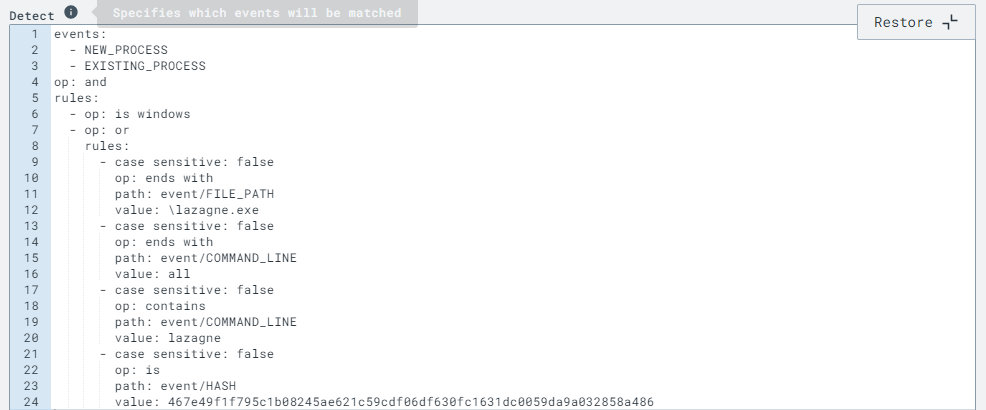
From this information we are going to create our rule.

**Detection And Response Rule**

Head over to Detections and then D&R rules. From there click on the “New Rule” button.



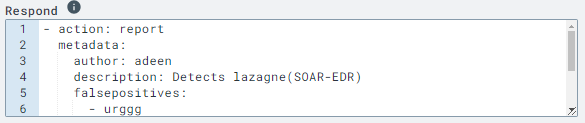
(If you are new to rule creation, you can take hints from already created rules to see how a new rule can be created)

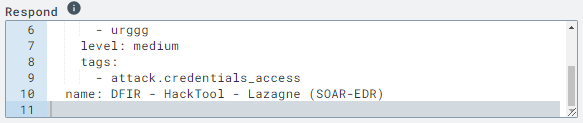
**Detection Rule**

The rule trigger if the process event is under the *New Process* or *Existing process* type and the machine is windows machine.

The rule will ignore case sensitivity and then search for the file path ending with lazagne.exe, and search for ‘all’ flag at the end and also compares the hash value.

**Response**

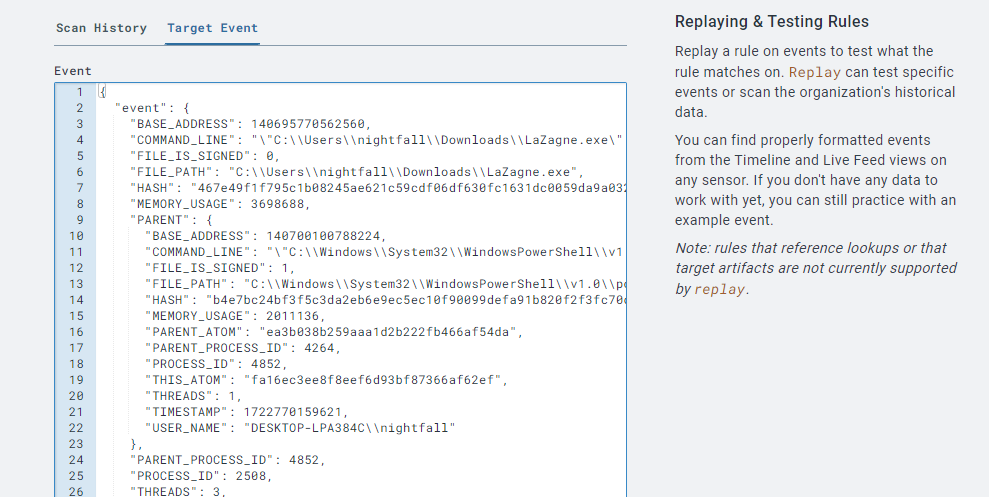




**Testing**

We will save and test the rule.

Grab the event from Timeline and paste it in test area provided at the end of the rule page.



**Result**



So our rule is working and providing detection when Lazagne.exe is executed.

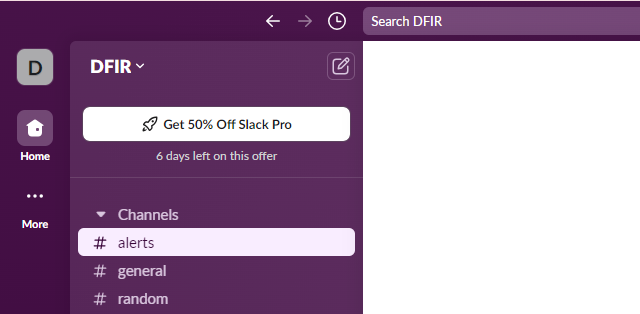
1. **Forwarding Detection To Tines**

We will setup our slack and Tines in this section and then test our connection between LimaCharlie and Tines, making sure our soar is receiving detection made by LimaCharlie.

**Slack**

Slack is a collaboration platform designed to streamline communication and collaboration within teams and organizations. It combines messaging, file sharing, and integration with other tools to create a unified workspace.

Create a new workspace in slack. Then create a channel *#alerts* for receiving messages of detection from Tines.

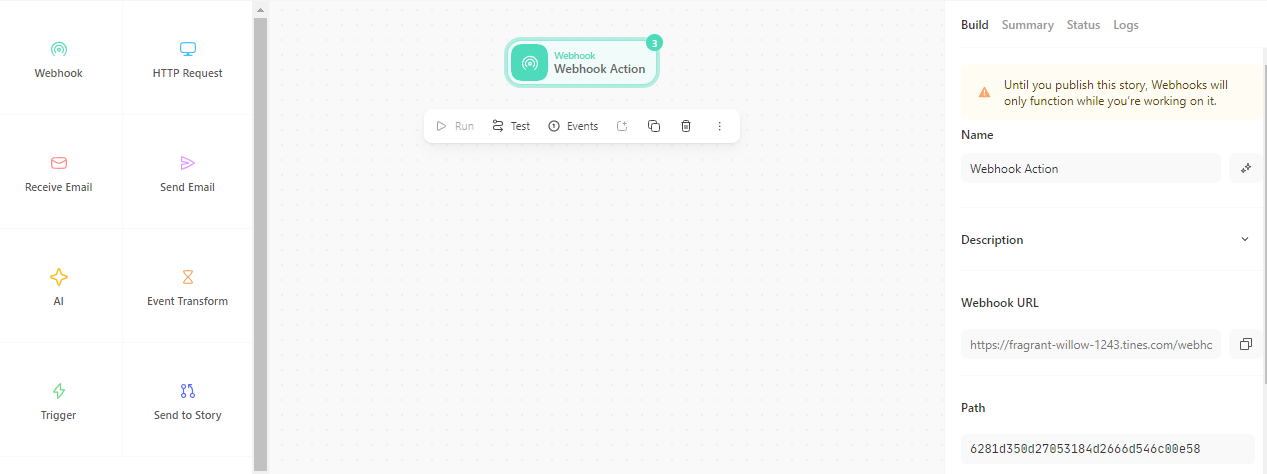


**Setting Up Tines For Detection**

The **Tines** is a security automation tool designed to help security teams automate repetitive tasks, manage workflows, and respond to incidents efficiently. Unlike traditional SOAR platforms, Tines aims to simplify the automation process with a user-friendly, no-code interface.

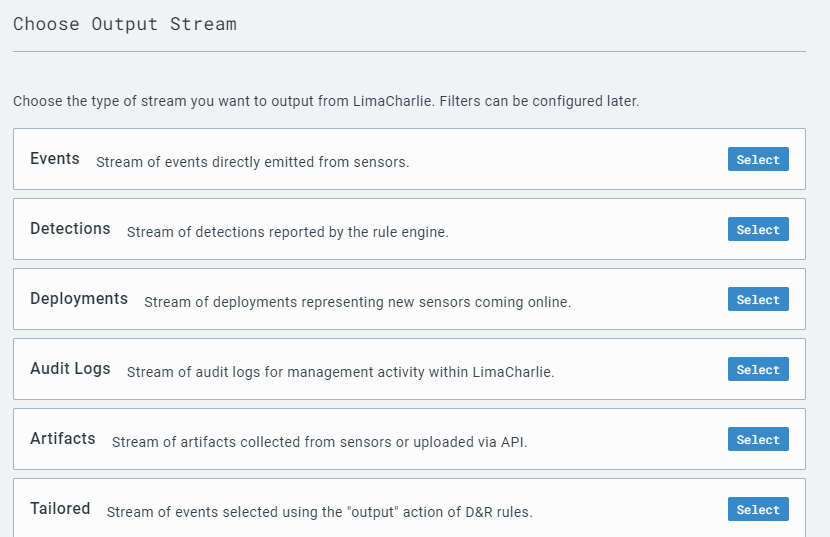
In Tines workflows (called stories/playbook) are created that automatically perform tasks like detecting threats, responding to incidents, and managing alerts. By connecting with various security tools, Tines helps make the security process faster and more efficient.

Create a new story in Tines. Select webhook and copy the URL.

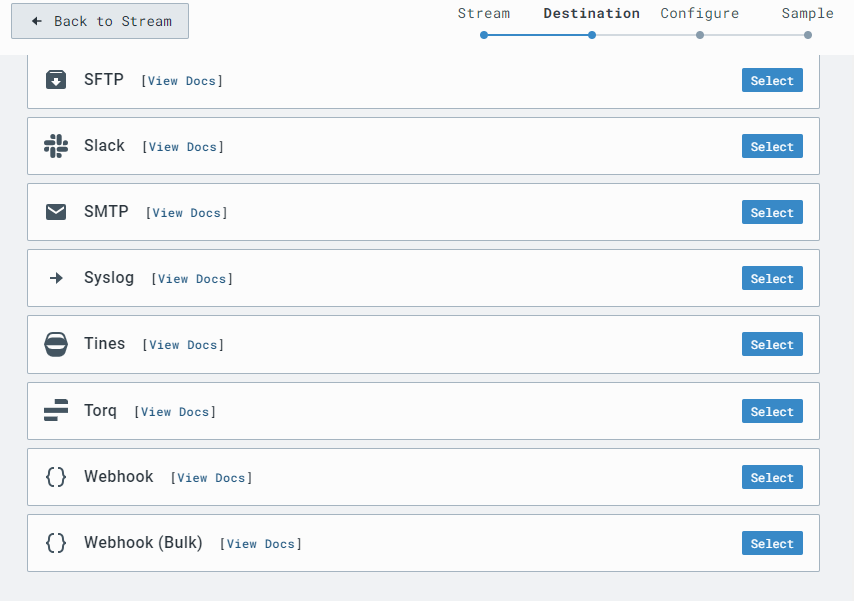


We will establish a link between LimaCharlie and Tines using this URL.

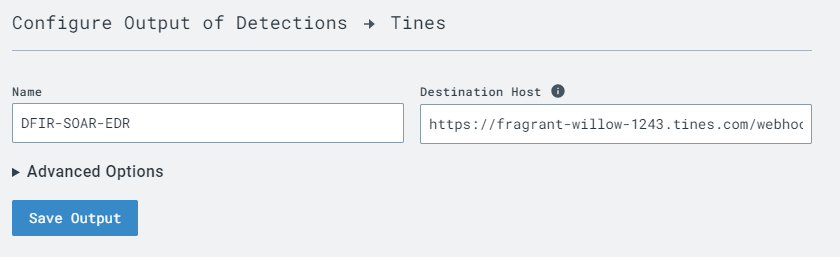
In LimaCharlie go to Outputs and then click add, we will see a list of outputs.



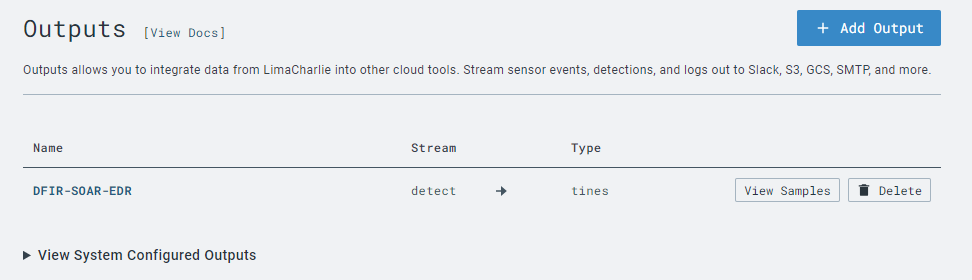
Select detections and from the list select Tines (we can also use webhooks if Tines is not present).



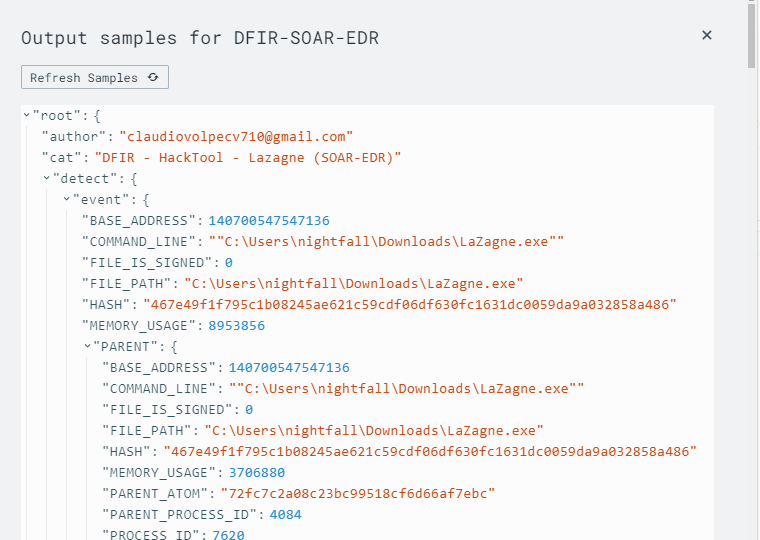
Paste the copied URL of webhook from Tines in Destination Host.



Save the output and check the connection to Tines.

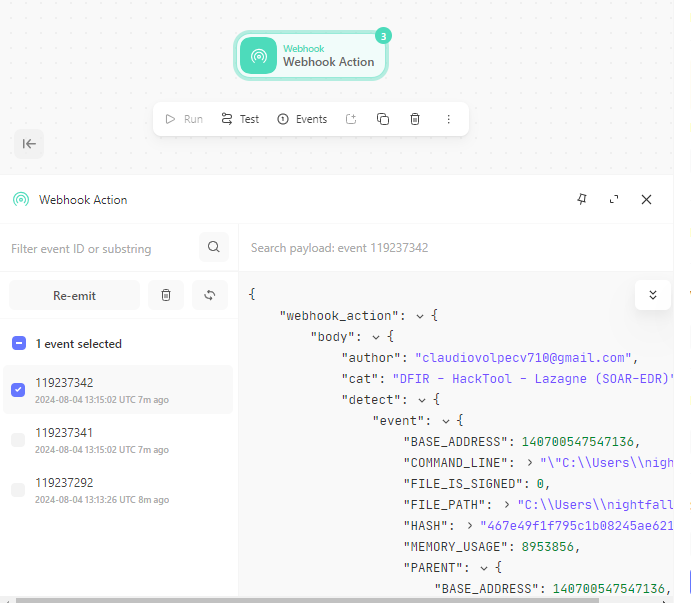


Generate the event again by executing lazagne. We get a sample output for Tines.



Head over to Tines to check for event detection.

From webhook click events and then select the most recent event.



We get the same exact detection on Tines.



Now we know the detection is working in both LimaCharlie and Tines. Next, we have to setup slack for receiving detection messages.

So now let’s create a playbook for automation.

1. **Generating Playbook For Automation**

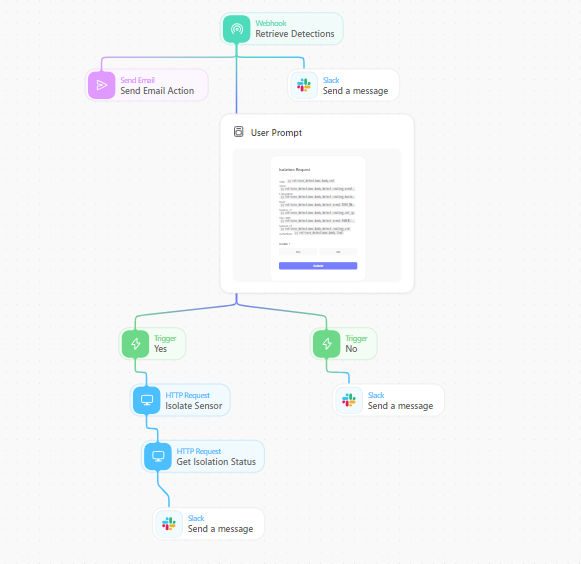
In this section we will create a playbook in Tines using the workflow diagram we created before.

**Playbook**

A **playbook** in the context of cybersecurity, especially within SOAR (Security Orchestration, Automation, and Response), is a predefined set of procedures and actions that are executed automatically in response to specific security incidents or threats. A playbook is like a detailed step-by-step guide for dealing with certain types of security problems. It tells you what to do and in what order to ensure the issue is handled correctly and efficiently.

The objective of this playbook will be to send a message to slack and also send an email containing information about detection. Tines will then generate a user prompt to either isolate the machine. If user selects *Yes,* then LimaCharlie should automatically isolate the machine.

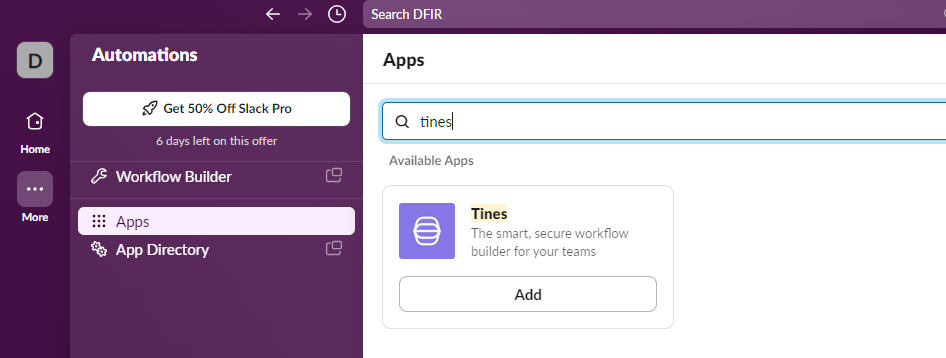
The following image shows the complete Tines playbook for this project.



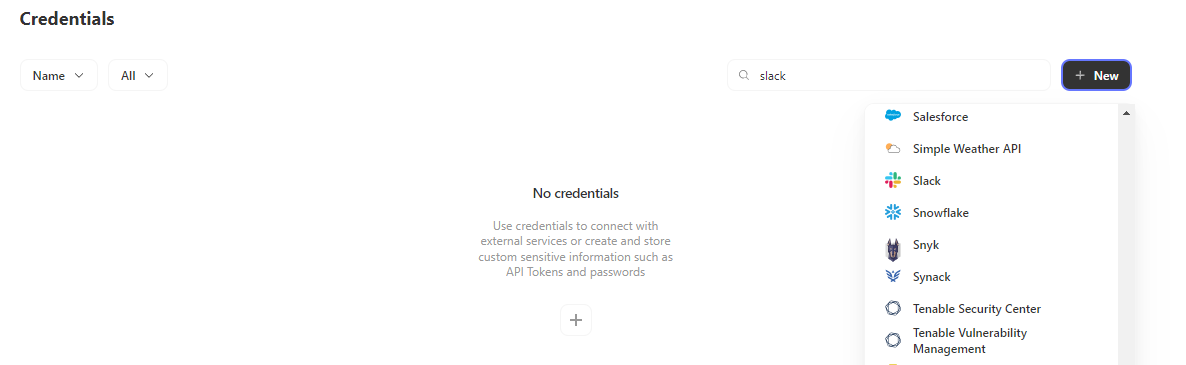
**Linking Slack**

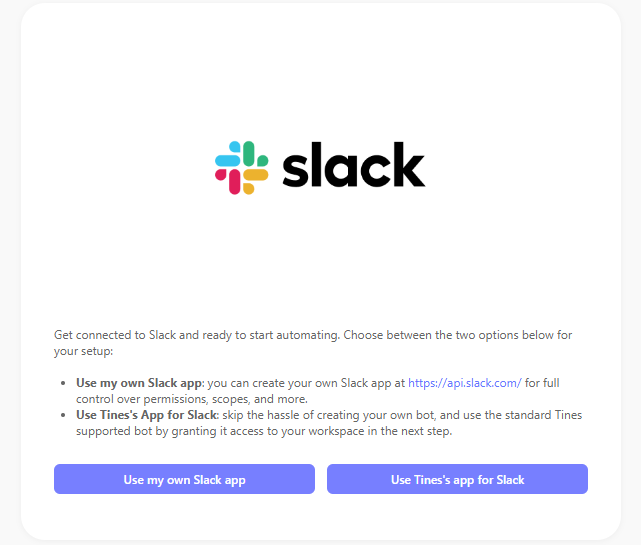
First we need to connect Tines and slack.

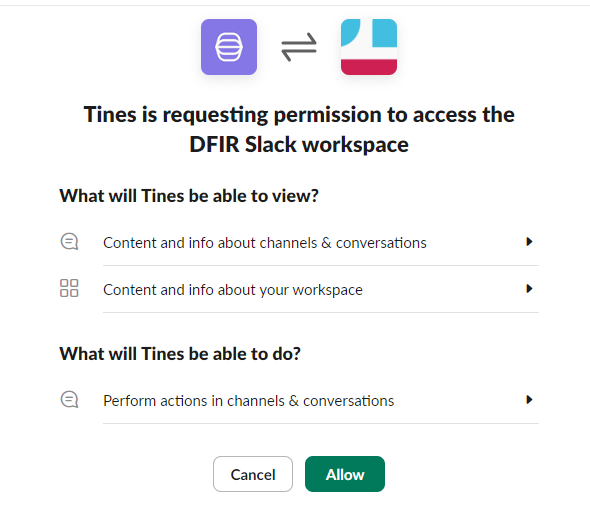
In slack go to more options and then select automations. Search for Tines and click add.



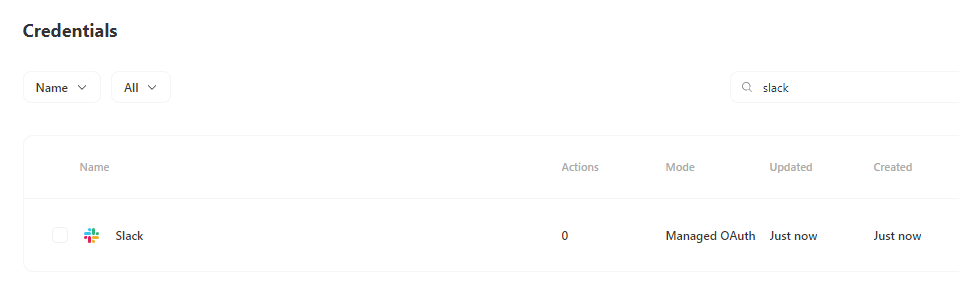
In Tines go back to the dashboard and then move to credentials. We need to add the slack API here.



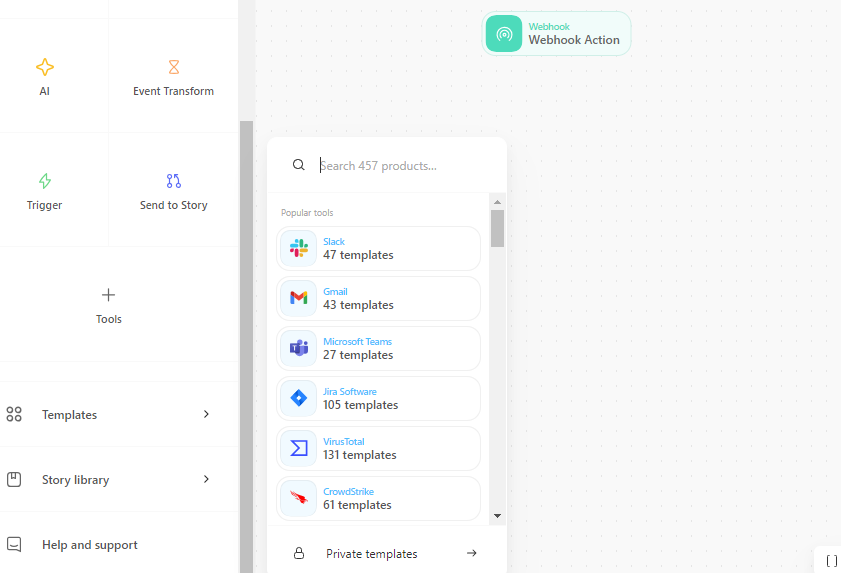




The link has been established between Tines and slack

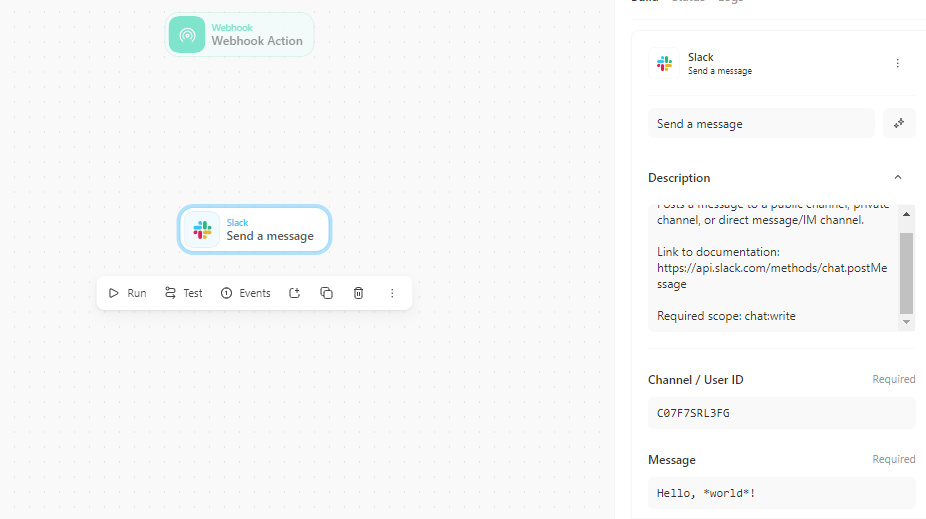


Head back to the story in Tines and add the slack template.

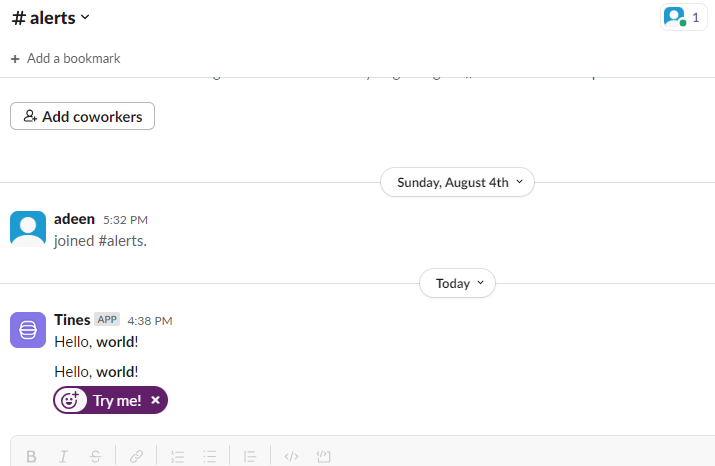


We want to send message to the slack, so search for “send a message’ on the right panel, this uses the “send message” template for slack and tells Tines to perform this action when automated.

We want the message to be sent to our *alerts* channel in slack, so will copy the channel ID from channel details in slack and add it here.

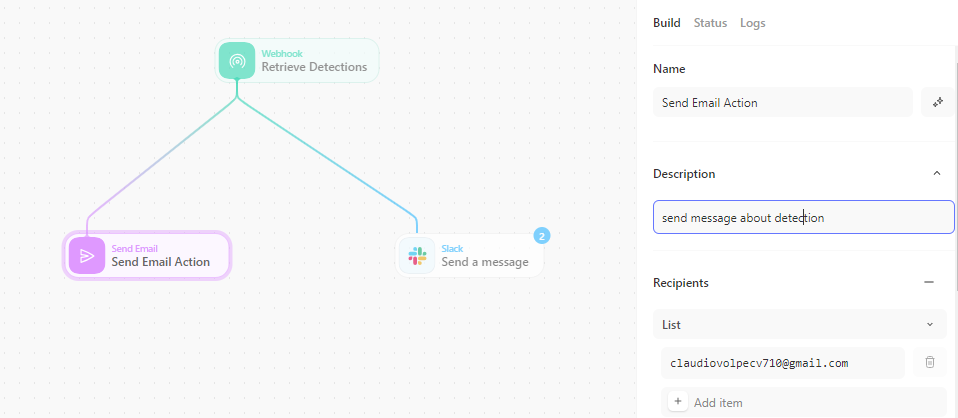


Run and check if messages are being delivered to slack.



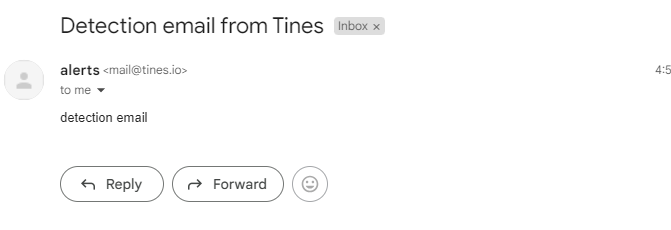
**Linking Email**

We are receiving messages in slack’s alert channel. Now, we also need to send an email. Add email from the items on left panel. Add description, recipient of email and the body of email.



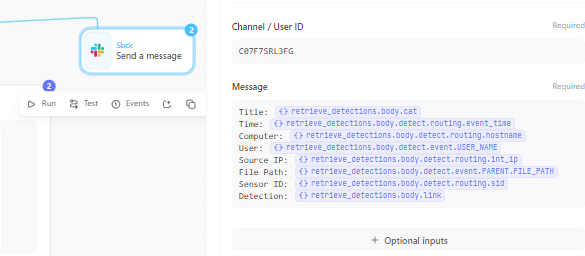
Run to check if it is sending email.

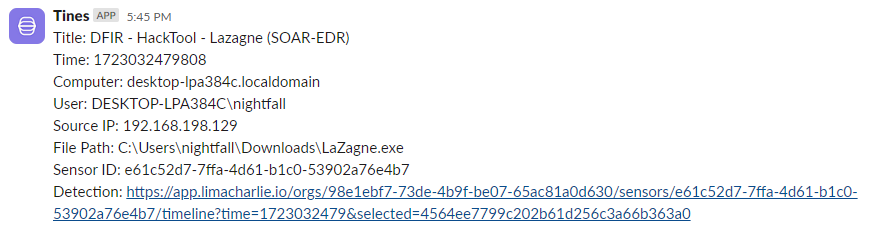
We get email in our inbox.



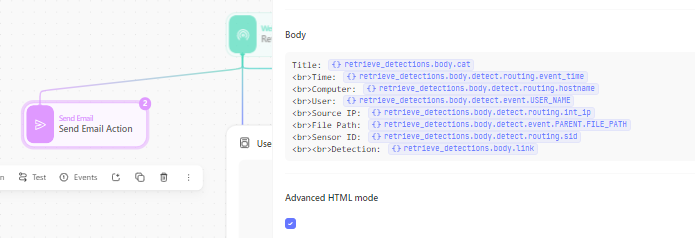
**Adding Detection Details**

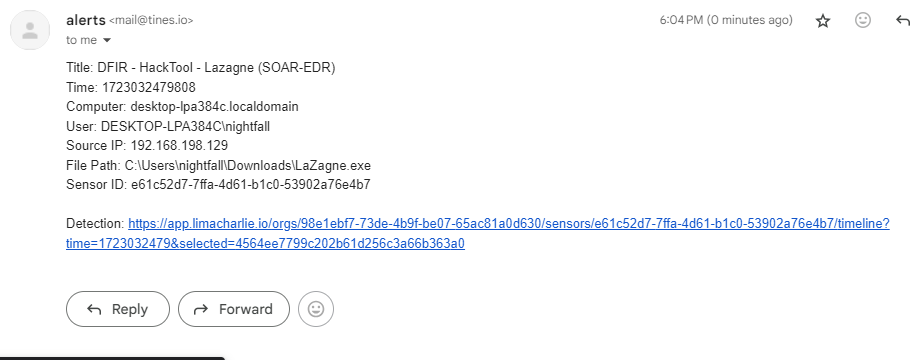
Add the details of detection in send message for slack and also in email body.





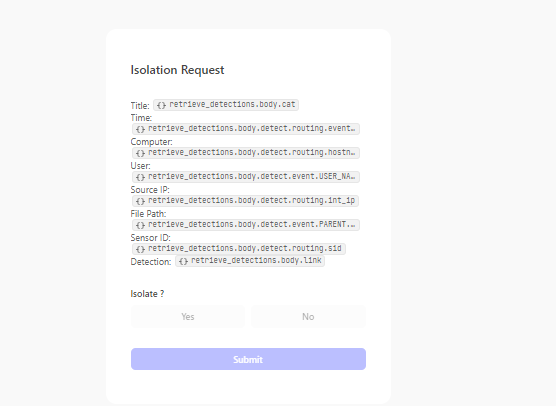
Add details of detection in email body section.



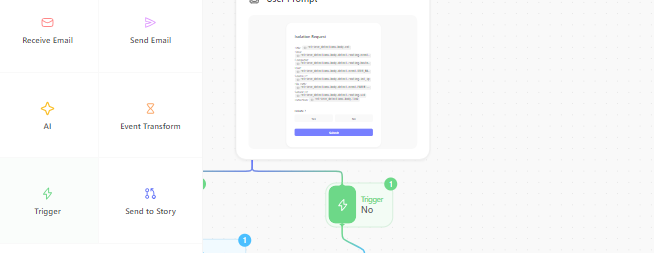


**USER Prompt**

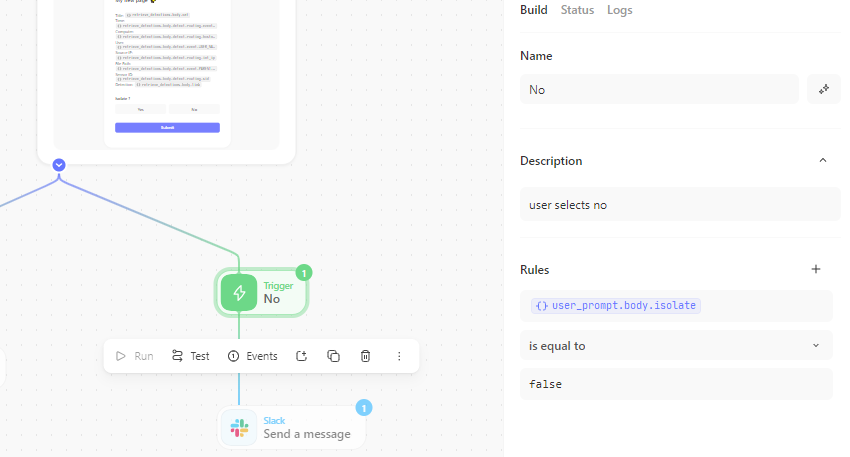
For generating a user prompt we will use page in Tines. Also add the details of detection here and ask the user to either isolate the machine or not.



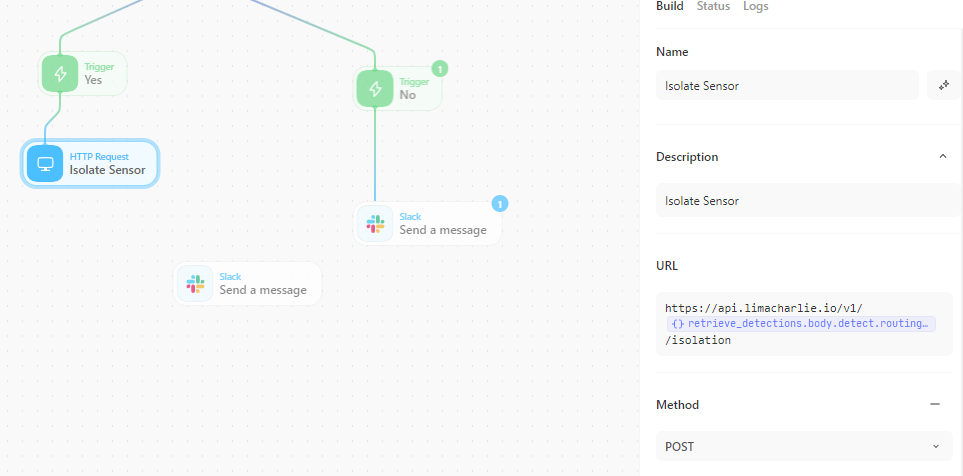
For user’s decision we use trigger in Tines. If user selects no then we need to simply send a non-isolation status message.



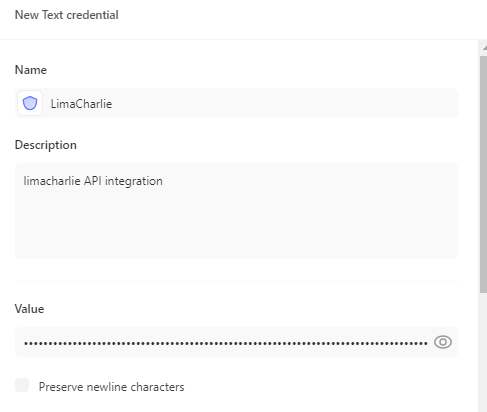
In the rules section add the isolation status.



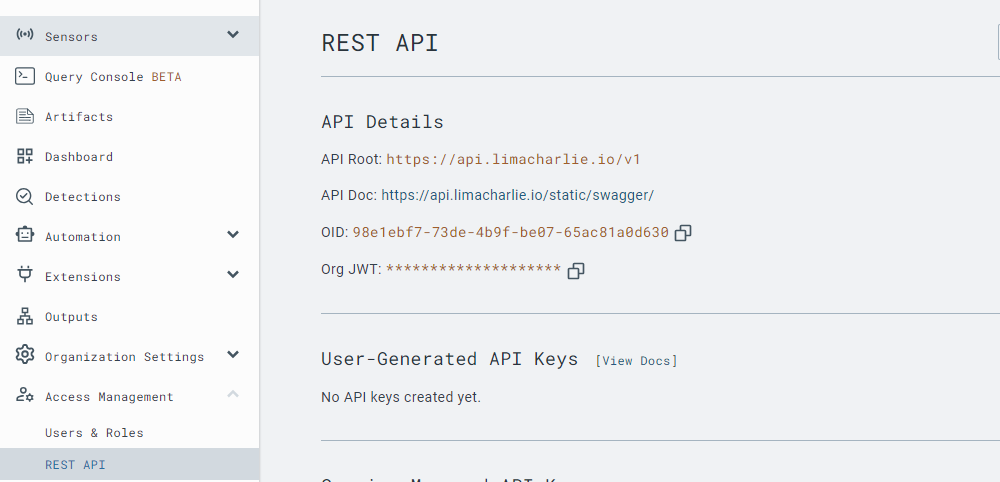
Adding the same slack template as before while changing the URL.



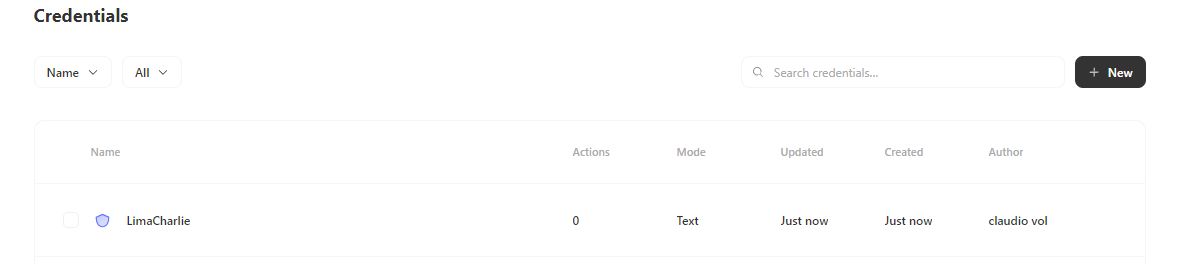
We need to add LimaCharlie API to get isolation status from it. Go to credentials page again in Tines and add manually the credentials.



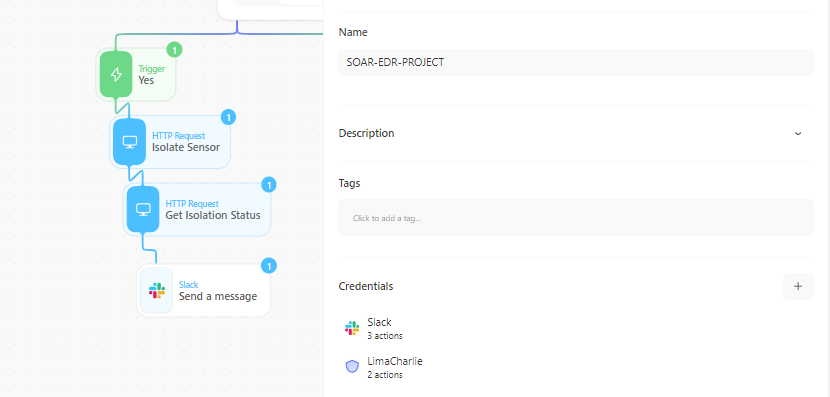
The API value can be obtained from LimaCharlie.

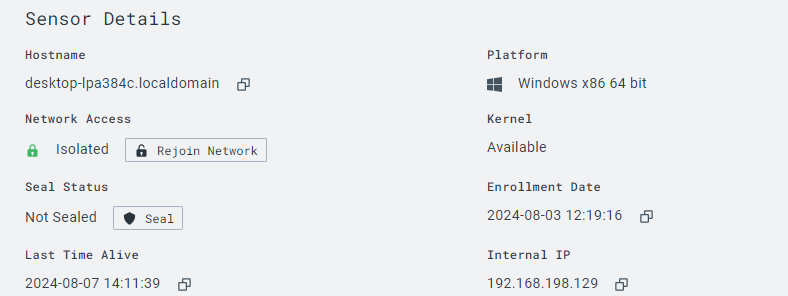


The API credential is added.

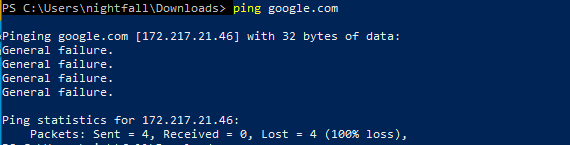


Similarly, if user selects Yes the machine is isolated by LimaCharlie and the isolation message is sent to slack.





The machine will be isolated and not able to connect outside.



1. **Summary**

In the project we use LimaCharlie to detect any malicious action on our endpoint and then automated the process of isolating the endpoint using Tines. Tines will also send detection messages to slack in particular channel for informing the team or dedicated person while also sending email about detection.