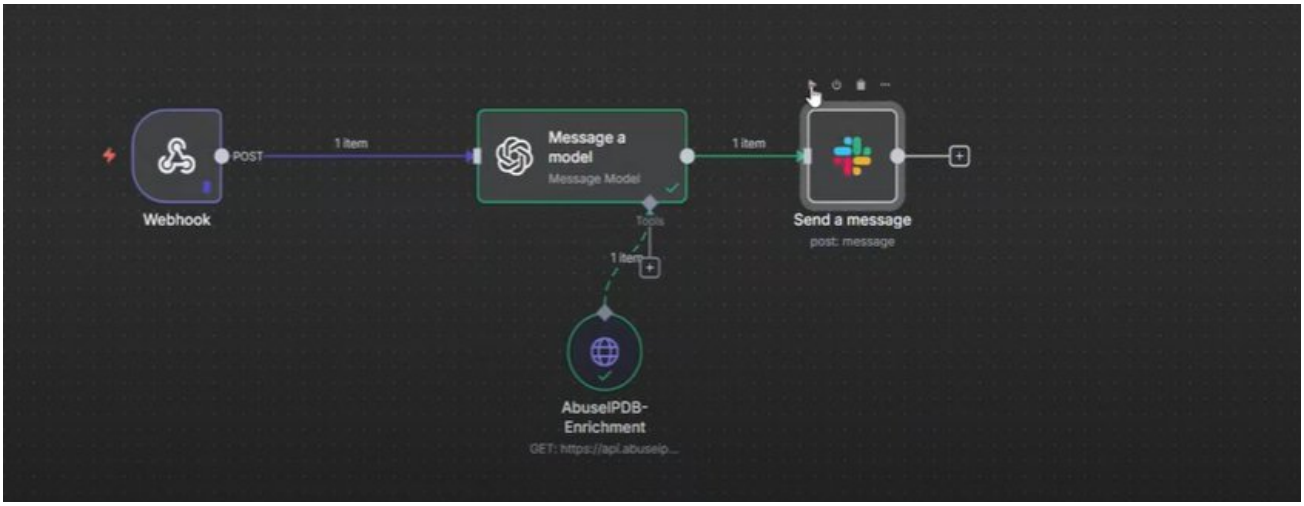
Leonard M. Estos
Cybersecurity Researcher & Technical Author
Edmonton, Alberta, Canada

Project Name: How To Use AI in Your SOC Workflow!

SOAR stands for **Security Orchestration, Automation, and Response**.

It is a platform and a set of technologies that helps **Security Operations Center (SOC)** teams manage the massive volume of security alerts they receive by combining tools, automating repetitive tasks, and standardizing incident response. The Primary goal is to Respond, contain, and remediate threats by automating workflows. If **SIEM** is the eyes and the brain of the SOC (Tells you what happened.), now **SOAR** is the hand and feet of the SOC (Tells you how to fix it.).

N8N Workflow is an **open-source**, low-code platform often described as a workflow automation tool or workflow engine. It allows you to connect different apps and services together to automate complex tasks, sync data, and build custom workflows without having to write extensive code.



Requirement's:

1. Windows Server or Windows 10 VM
2. Splunk VM = SIEM
3. Kali Linux VM
4. N8N instance VM (Ubuntu)= SOAR Workflow
5. DFIR-IRIS VM = Open-Source Incident Response Platform

Splunk installation and configuration:

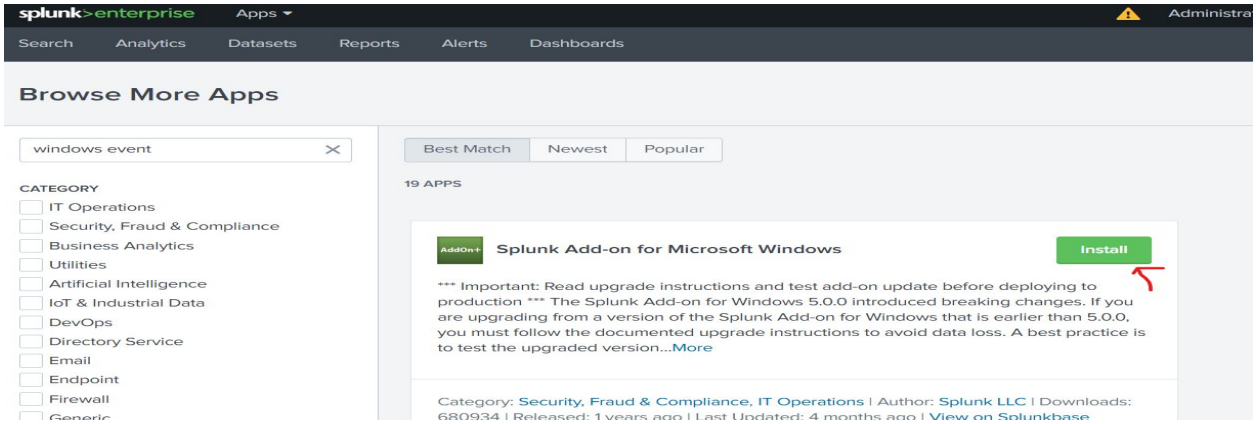
1. **Register** to Splunk https://www.splunk.com or https://www.splunk.com/en_us/download/splunk-enterprise.html

2. **Install** Splunk Server and Agent (Ubuntu or Windows). For me I installed this in my local device Windows 11.

   https://www.braindumps.com/blog/how-to-install-splunk-on-windows-and-linux-a-complete-step-by-step-guide/

3. **Configure** your **Splunk Server Applications**

- In Settings > Forwarding and Receiving > click Configure receiving > New Receiving Port (**9997**) > Click Save
- Click Settings > Indexes > New Index > Name: **lizjames-project**
- Click > Apps > Find more apps > search for Windows event and > **Install Splunk Add-on for Microsoft Windows** > sign-in using your Splunk account > Done
  The purpose of this **Add-on** is to get the complete information from Windows Indexes.



- Configure **your input.conf** in ***C:\Program Files\SplunkUniversalForwarder\etc\system\local***

- How to check > In your Splunk click > Click Search and reporting and type below

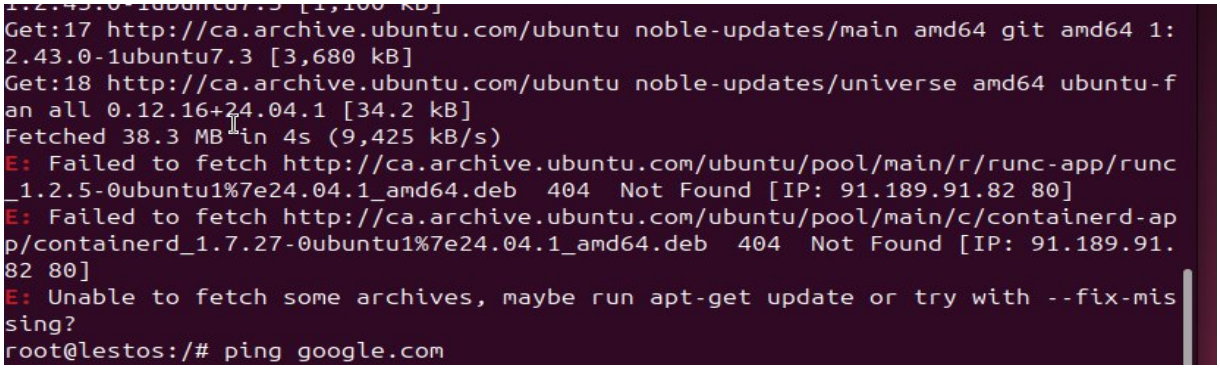
  Failed login

  index=lizjames-project source="WinEventLog:Security" EventCode="4625" host=LizJamesDC1

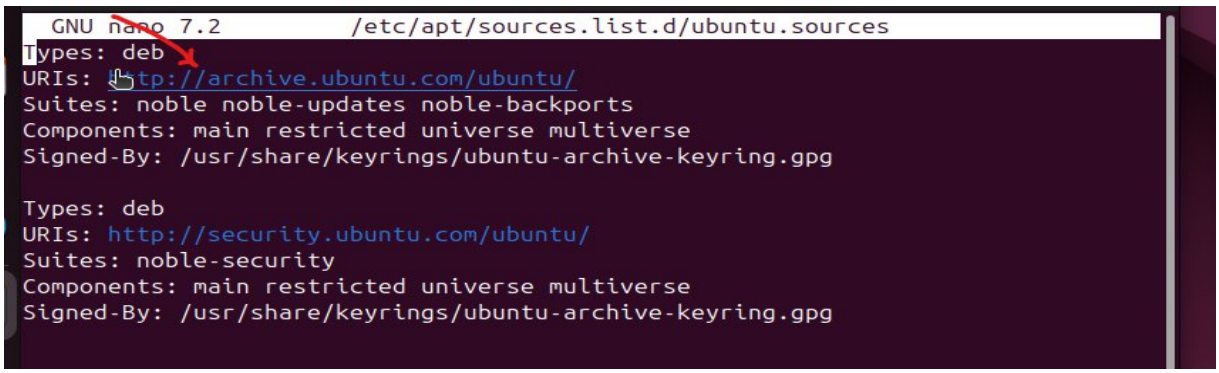
Successful login

index=lizjames-project source="WinEventLog:Security" EventCode="4624" host=LizJamesDC1

4. **Install Docker-Compose in Ubuntu machine.**
    - Type below
      $ **Docker-compose** = to check the command availability
      $ **Sudo apt install docker-compose**
    - Now, if there is an error below like Failed to fetch



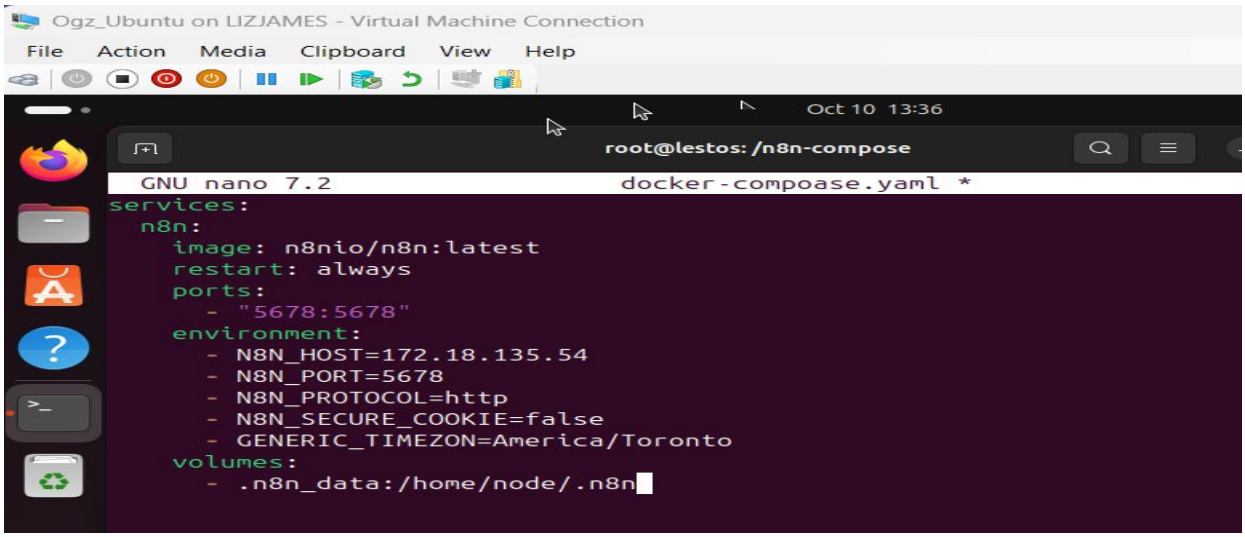
- Run = # **sudo nano /etc/apt/sources.list.d/ubuntu.sources**
- Edit the URIs: http//ca.archive.ubuntu.com/ubuntu and remove the **ca**



Run $ **sudo apt-get update** and $ **sudo apt-get upgrade -y** and type $ **docker** again. Now run the docker installation again $ **Sudo apt install docker.io and Sudo apt install docker-compose**

5. **Install the N8N in Ubuntu machine.** The **root** is **/lestos@lestos:** and file is **docker-compose.yaml** if you want to edit the IP.

$ **mkdir n8n-compose**
$ **cd n8n-compose/**
$ **sudo nano docker-compose.yaml** and provide the configuration below.
Services:
N8n:
Image:n8nio/n8n:latest
Restart: always
Ports:
-5678:5678
Environment:
-N8N_HOST= 172.18.135.54 **(your Ubuntu IP)**
-N8N_PORT=5678
-N8N_PROTOCOL=http
-N8N_SECURE_COOKIE=false
-GENERIC_TIMEZONE=America/Toronto
Volumes:
-./n8n_data:/home/node/ .n8n



Ctrl - X + Y to save the configuration

- Start to verify your configuration: $ sudo nano **docker-compose.yaml**

  Run $ **apt install docker-compose**
  Run $ **sudo docker-compose pull**

  Note: If encountered an **error** run the **setup tools package**

  Run $ **sudo apt-get install python3-distutils**
  Run $ **apt install python3-pip**

  Continue:

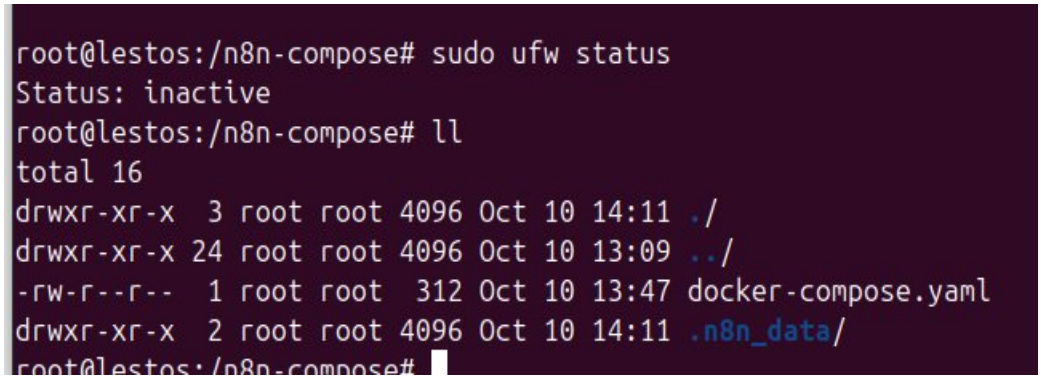
  Run $ **sudo docker-compose pull**
  Run $ **sudo docker-compose up -d**

- Now, try to access the URL type IP (Device):5678 e.g. **172.18.135.54:5678**

  Note: If encountered an **error** run bleow

  $ sudo ufw status = if result is inactive

  $ ll = the issue is that .n8n_data/ have no permission and permission is in root not at lestos.



  We will now provide the permission
  $ **sudo chown -R 1000:1000 .n8n_data/**
  $ **ll** = to check that permission is now lestos

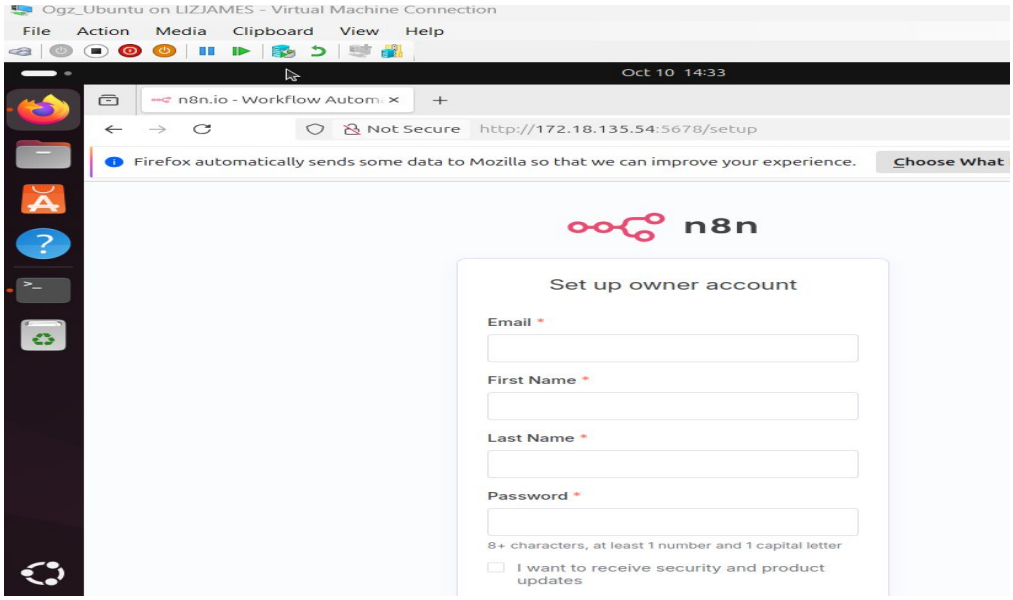
```
total 16
drwxr-xr-x  3 root   root   4096 Oct 10 14:11 ./
drwxr-xr-x 24 root   root   4096 Oct 10 13:09 ../
-rw-r--r--  1 root   root    312 Oct 10 13:47 docker-compose.yaml
drwxr-xr-x  2 lestos lestos 4096 Oct 10 14:11 .n8n_data/
root@lestos:/n8n-compose#
```

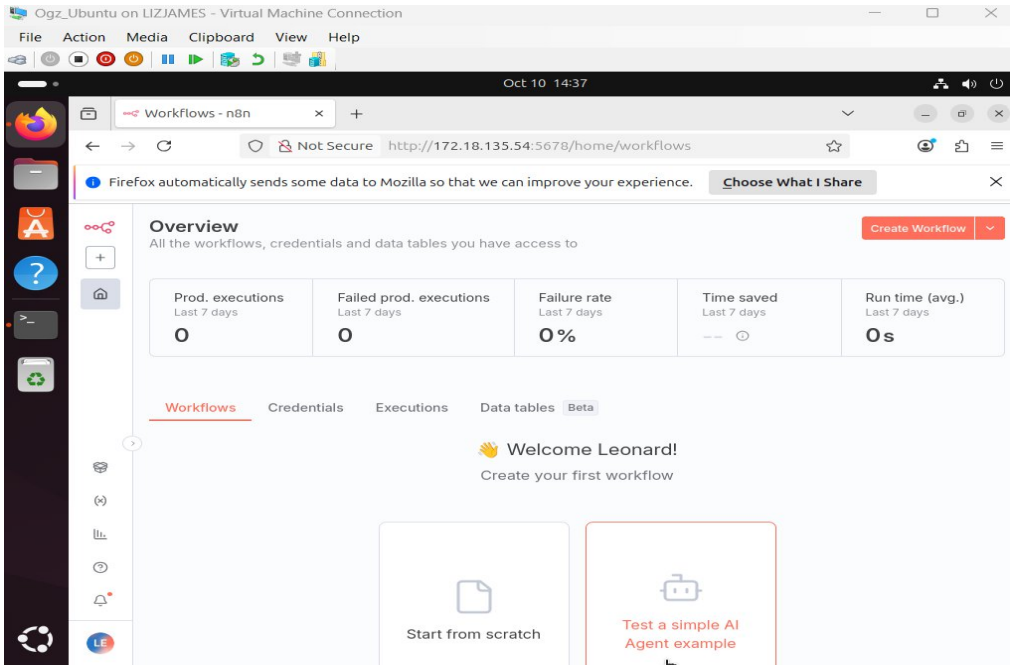
$ **clear**

$ **sudo docker-compose down**

$ **sudo docker-compose up -d**

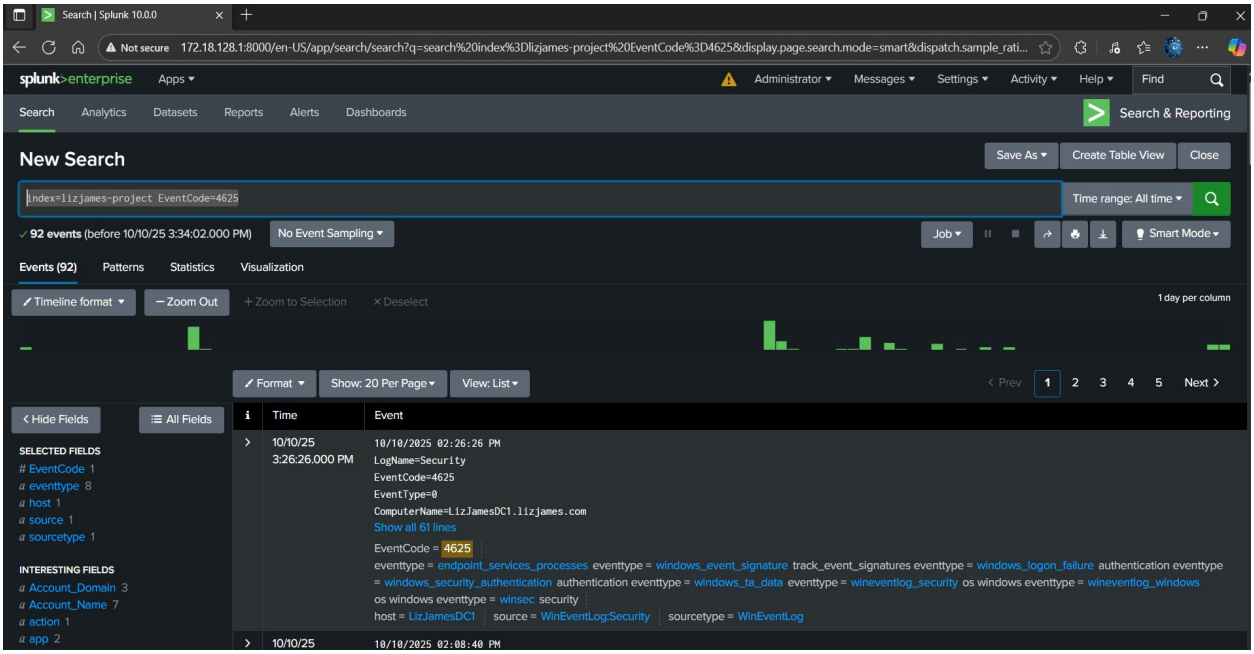
6. The results, **perfect!** Now, continue to sign up



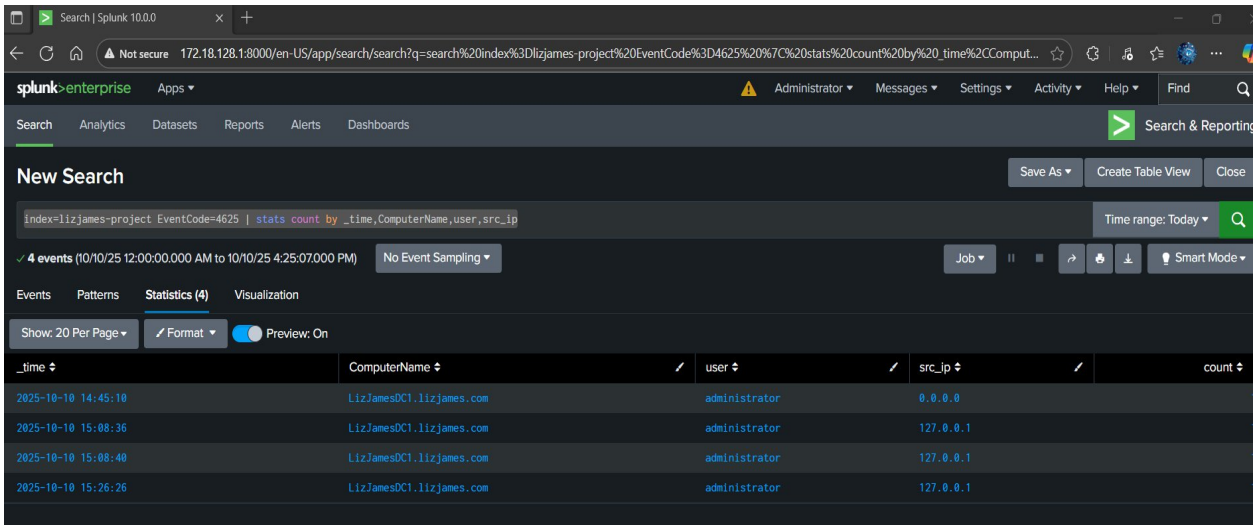


7. Now, go back to your SPLUNK and try to do a Search and Reporting

   index=lizjames-project EventCode=4625



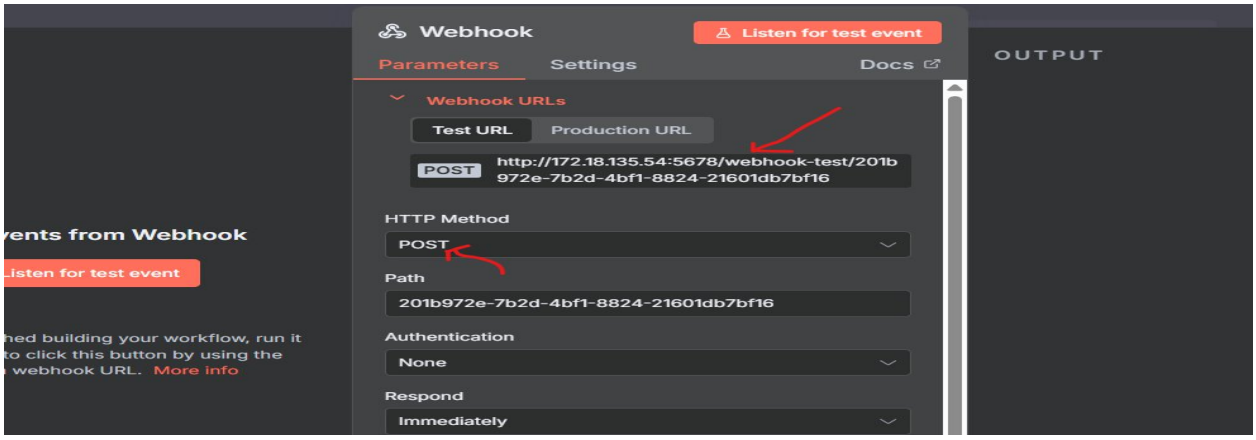
8. Now, we will create an alert for every **FAILED login**. Provide the query below. **index=lizjames-project EventCode=4625 | stats count by _time,ComputerName,user,src_ip**



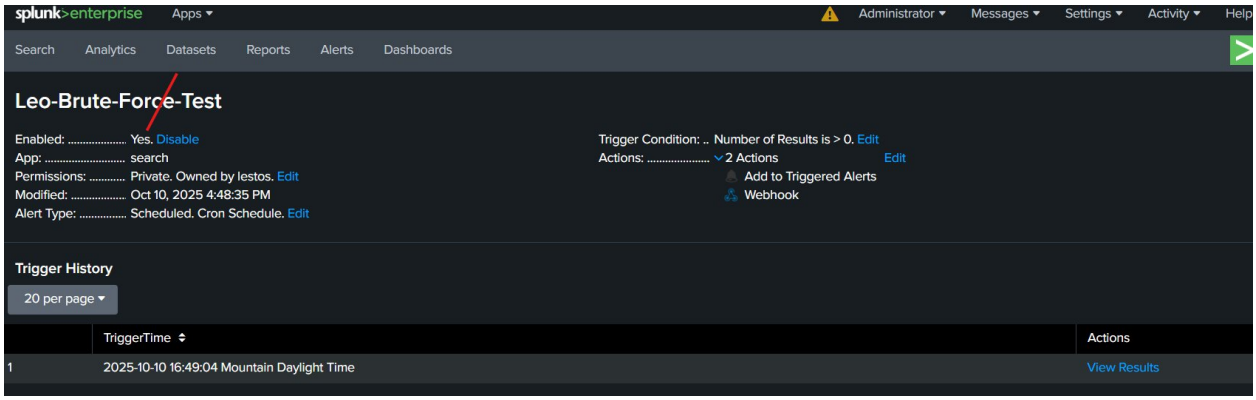
Click Save As > **Alert > Title: > Run on Cron Schedule > Last 24 hours > Change all Cron Expression to ***** > Click Trigger Action > Add Actions > Click Add to Triggered Alerts > Click Add Actions again > Webhook > Now, for the URL we will use our N8N URL**

9. Now, **we will start with our N8N configuration**.

- Click start from Scratch > Add first step > search for Webhook
- Change the **HTTP Method** for GET to **POST**
- Note look the URL below it will change from **GET to POST**
- Now, **get the URL and paste this to SPLUNK Webhook Triggered Alert**



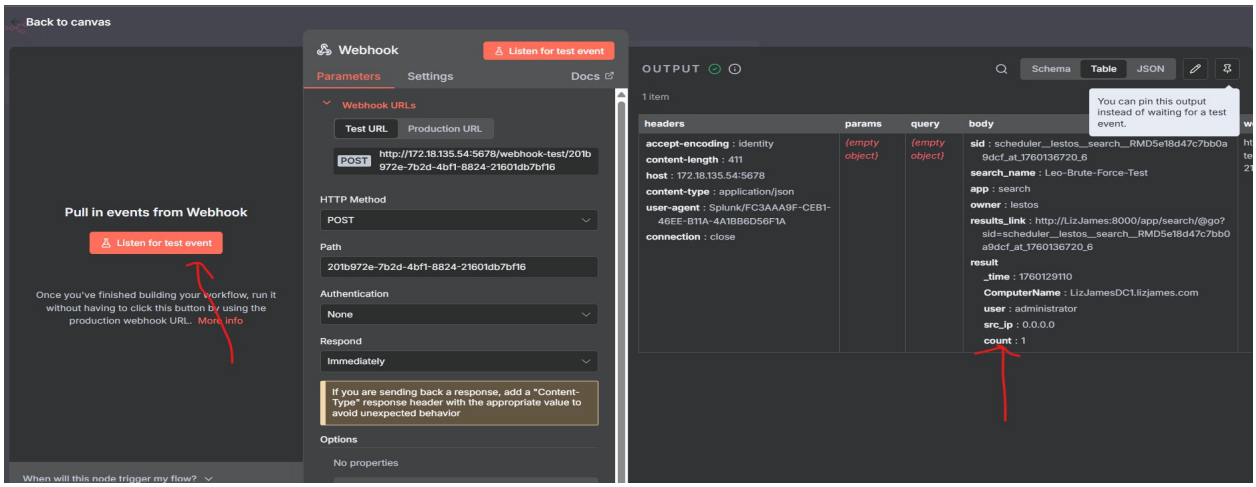
- Result in **SPLUNK**



Make sure that you will disable the alert the keep as triggered every minute after your testing. **Go back to SPLUNK and click Disabled**. > **click Settings > All configurations > Find your Alert or type in search Leo and in Status click disabled or enable.**

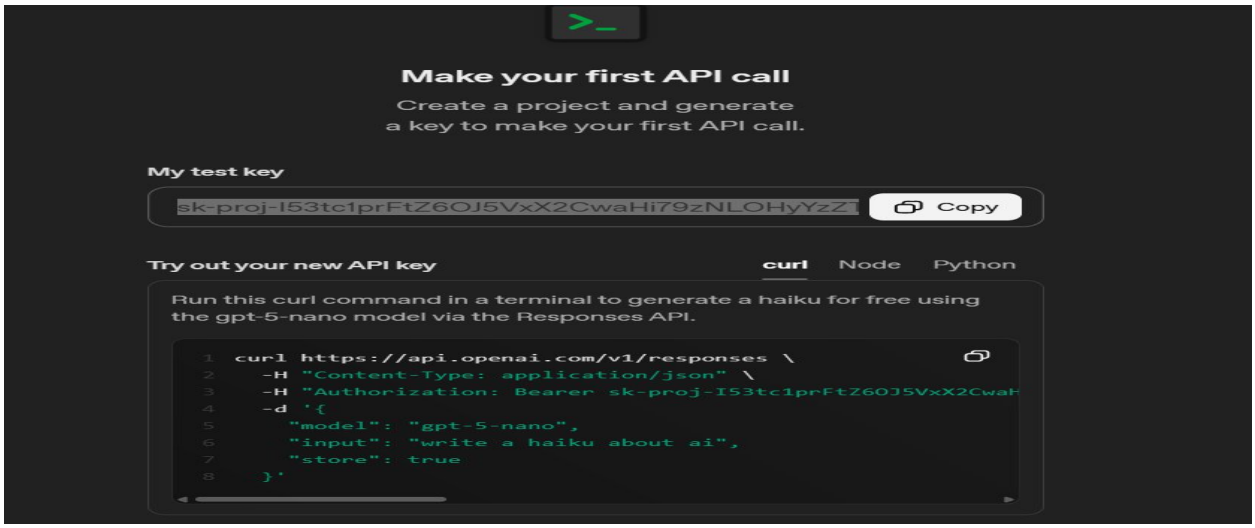
- Now, go back to your **N8N > click Listen for test event**

Results:

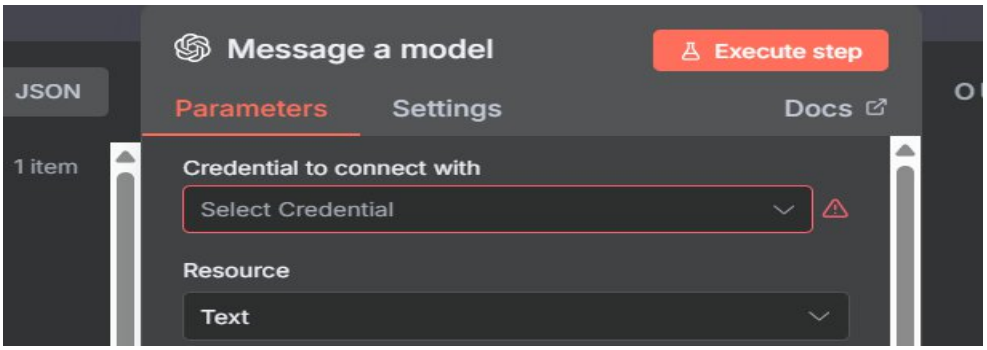


10. **Building Automation Workflow**: We will create an Automation inside the N8N, note that we will need an **API credit for an OpenAI** to make this work.
    - Now, go back to your N8N and click > **Back to Canvas > right click the Webhook and click PIN**
    - In the **top icon + click and search chatgpt > click OpenAI > Click Message a model >** Now, it will need a new credential a need an **API key**
    - Create an account to OpenAI (Chat GPT) > Go to https://openai.com/ > click **Login > API Platform > Create your account**
    - Click **START Building > Organization Name: S2S SOC Project >** Provide the email you will invite

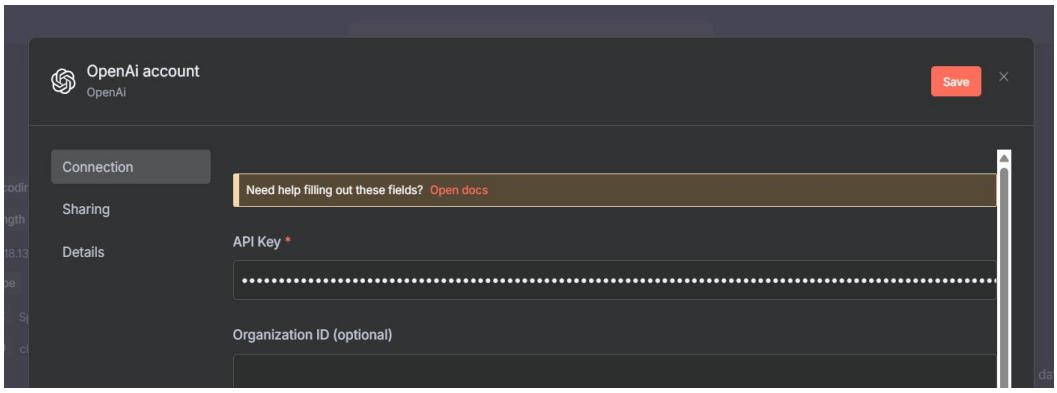
- **API Name: S2S SOC Project > Project name: Default project > Click generate API**



- Now, copy your OpenAI API key and go back to your N8N > **Click Credential to connect with > click create a new credential**



- Paste your OpenAI API key and click Save



- However, we need to purchase some credits. In your OpenAI > **click settings and click Billing > Add payment details 5$** is enough for this project.

- Now, once purchase completed **go back to your N8N** automation and configure the Message a model below > click **Model: GPT-4-1-Mini > Role: Assistant > Prompt:**

Act as a Tier 1 SOC Analyst assistant. When provided with a security alert or incident details (Including indicators of compromise, logs, or metadata), perform the following steps:

Summarize the alert - Provide a clear summary of what triggered the alert, which systems/users are affected, and the nature of the activity (e.g., suspicious login, malware detection, lateral movement).

Enrich with threat intelligence - Correlate any IOC's (IP addresses, domains, hashes) with known threat intel sources. Highlight if the indicators are associated with known malware or threat actors.

Assess severity - Based on MITRE ATT&CK mapping, identify tactics/technique, and provide an initial severity rating (Low, Medium, High, Critical).

Recommend next actions - Suggest investigation steps and potential containment actions.

Role: Assistant

- **Add another MESSAGE:**

  PROMPT:

  Format output clearly - Return findings in a structured format (Summary, IOC Enrichment, Serverity Assesment, Reccomendation Actions).

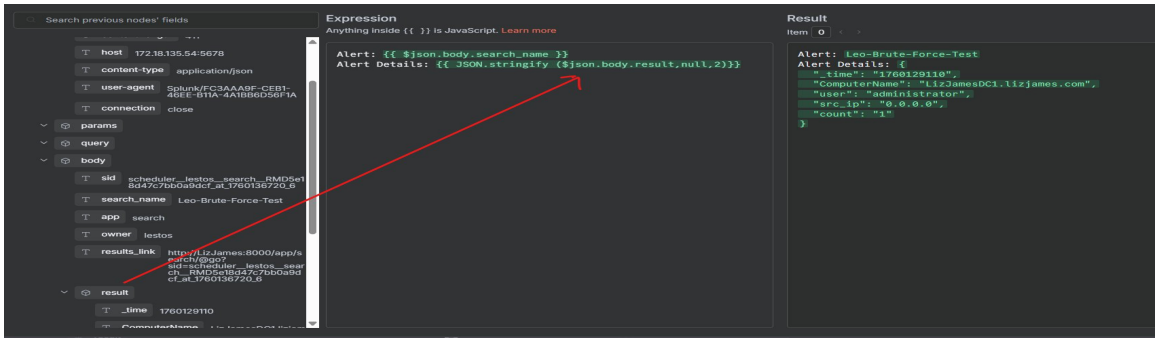
  ROLE: System

- **Add another MESSAGE:**

  PROMPT:

  Alert: Drag the **search_name** to the prompt, see sample output below.

  Alert Details: Alert: {{ $json.body.search_name }}
  Alert Details: {{ JSON.stringify ($json.body.result,null,2)}}



What we did in alert details: {{JSON.stringify > (drag the result,null,2)}}

ROLE: User

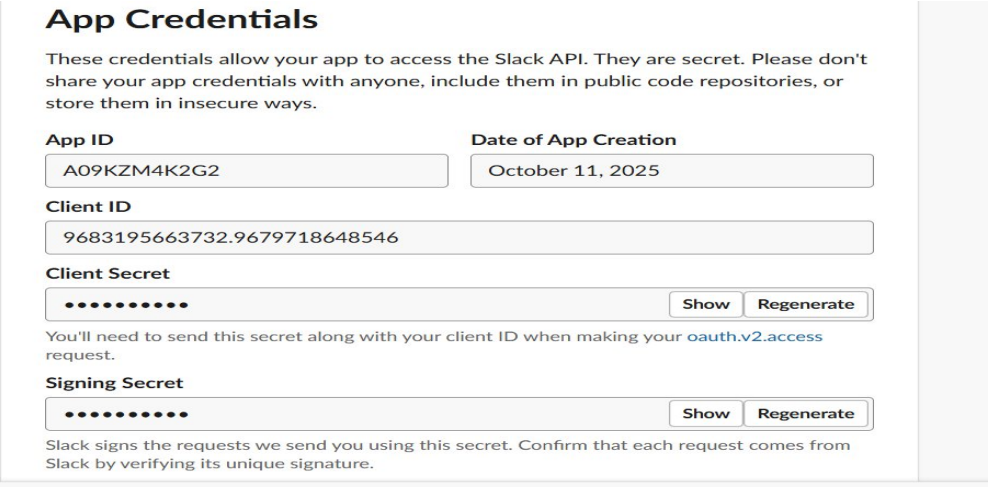
11. Connect our workflow to SLACK.
    - Create an account to slack
    - Click Create Workgroup > Company name S2S > Provide your name > Provide your picture > Provide email invite > click Start with the **Limited Free Version**
    - Right click Channels > New Channel > Next > Channel name: **alerts** > Click Create

12. Now, go back to your **N8N.**
    - in the right-hand side click > **+** > In the search type **Slack** > choose **Slack** > Go down and find the Message Action and choose Send a message

    - In the Send a message > credential to connect with > Create a new credential
    - Now, we don't know how to do it > Click Open Docs Slack credentials | n8n Docs > find the Using API access token

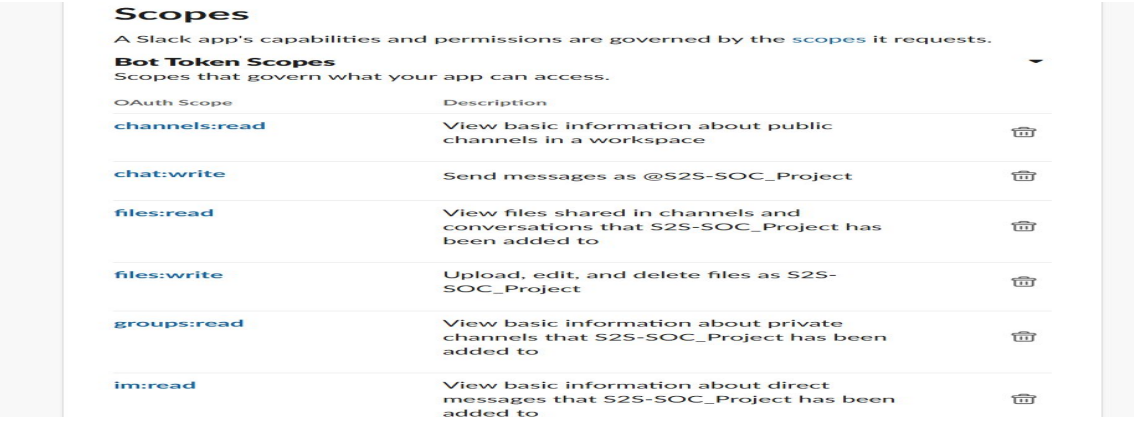
Using API access token

1. Open your **Slack API Apps** = https://api.slack.com/apps page.
2. Select **Create New App > From scratch**.
3. Enter an **App Name**.
4. Select the **Workspace** where you'll be developing your app.
5. Select **Create App**. The app details open.

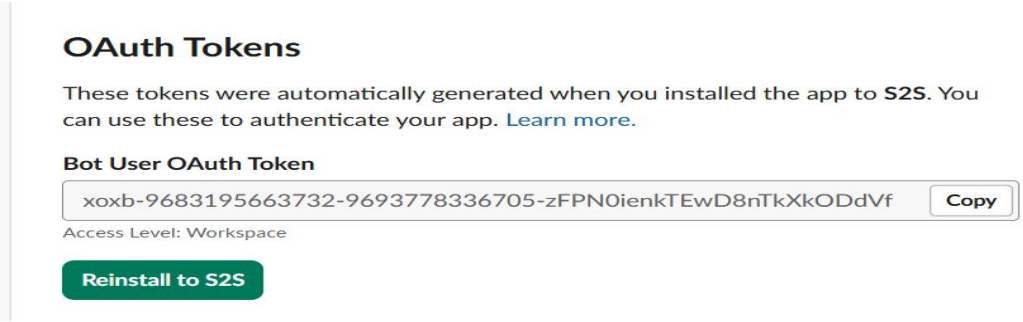


Now, follow the other instruction:

6. In the left menu under Features, select OAuth & Permissions.
7. In the Scopes section, select appropriate scopes for your app. Refer to Scopes for a list of recommended scopes.
8. Go to Scopes and click **Add an OAuth Scopes > Click Channels Read so on and so forth (Add all the scopes in Add an OAuth Scopes**, after you've added scopes, **go up to the OAuth Tokens section and select Install S2S** to Workspace > Click **Allow**. You must be a Slack workspace admin to complete this action.

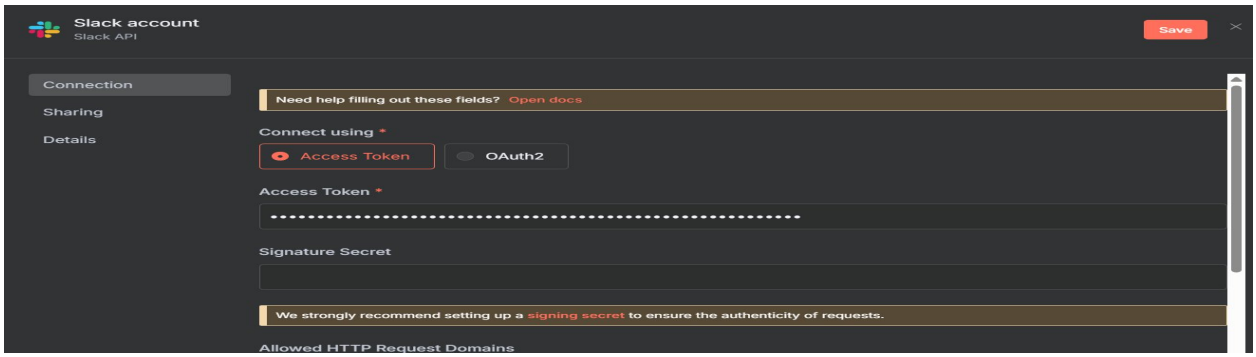


9. Select Allow.
10. Copy the Bot User OAuth Token and enter it as the Access Token in your n8n credential.

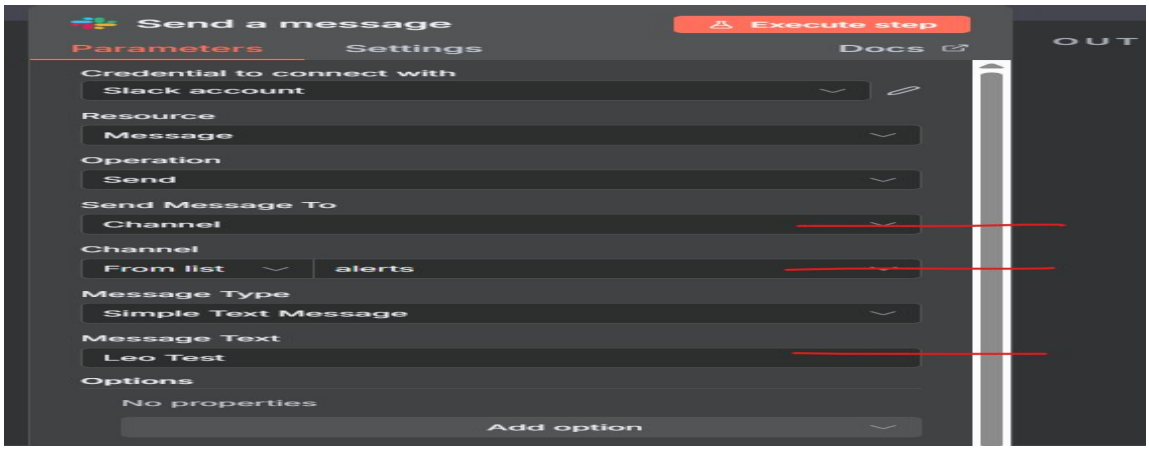


11. If you're using this credential for the Slack Trigger, follow the steps in Slack Trigger configuration to finish setting up your app.

13. Now, go back to your N8N and copy the Bot User OAuth Token > Click Save.

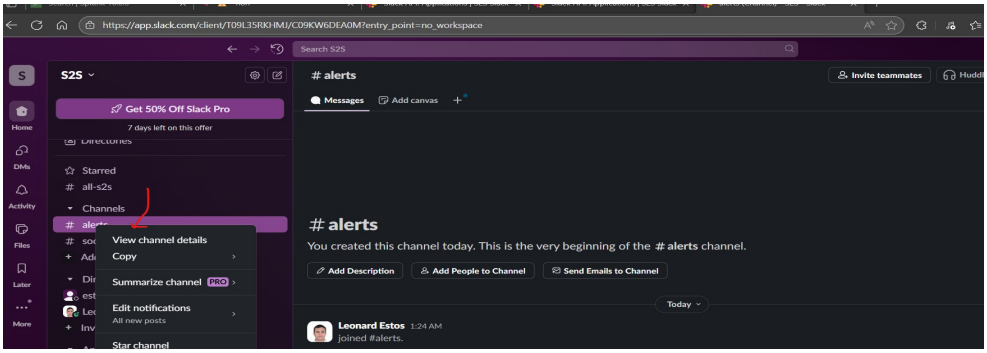


- Under the **Send Message To: Channel**
- Under the from **List: Alert**
- **Message Text: Leo Test > Click Execute Steps**

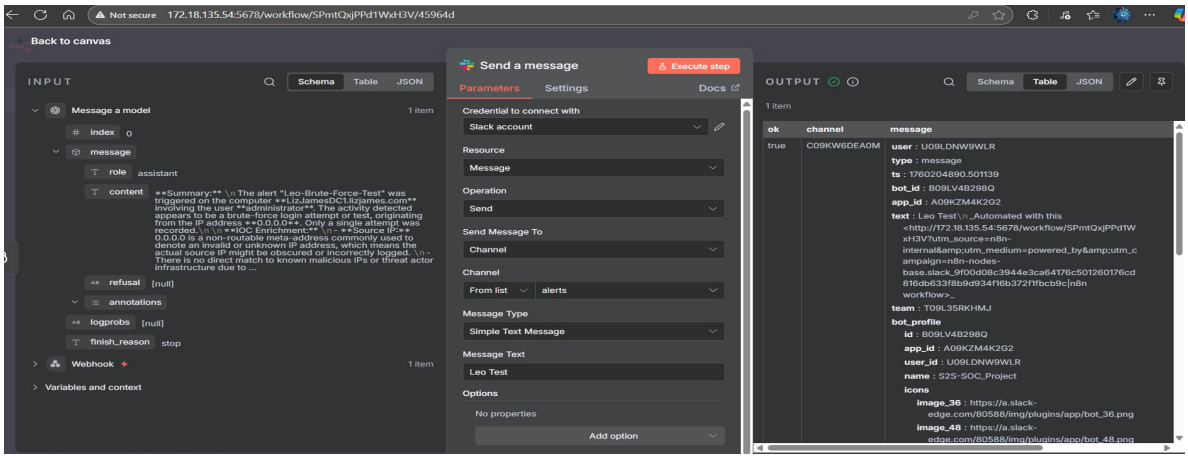


- Now, you will encounter **error** "**Problem in node 'Send a message'** Slack error response: "not_in_channel"

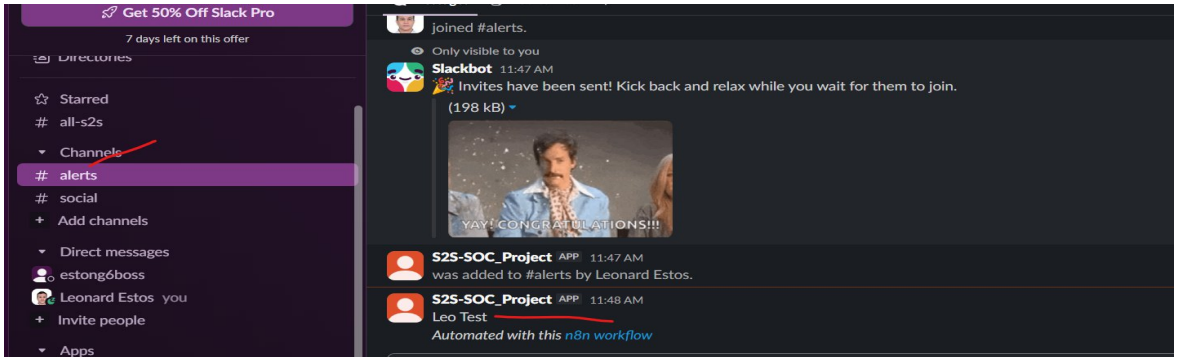
- Now, go back to your **SLACK > Right click the # alerts > view channel details**



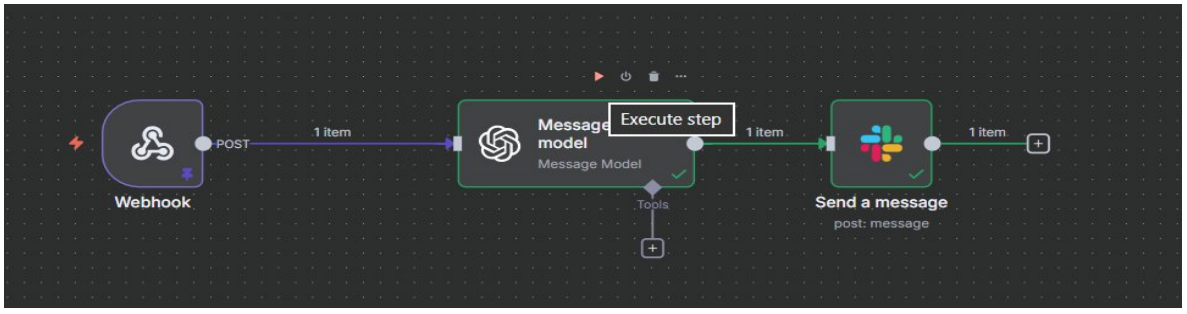
- Click **Integrations > Select Add Apps > Select S2S-SOC_Project**
- Now, go back to your **N8N and Click Execute Steps.** Result it's now working.



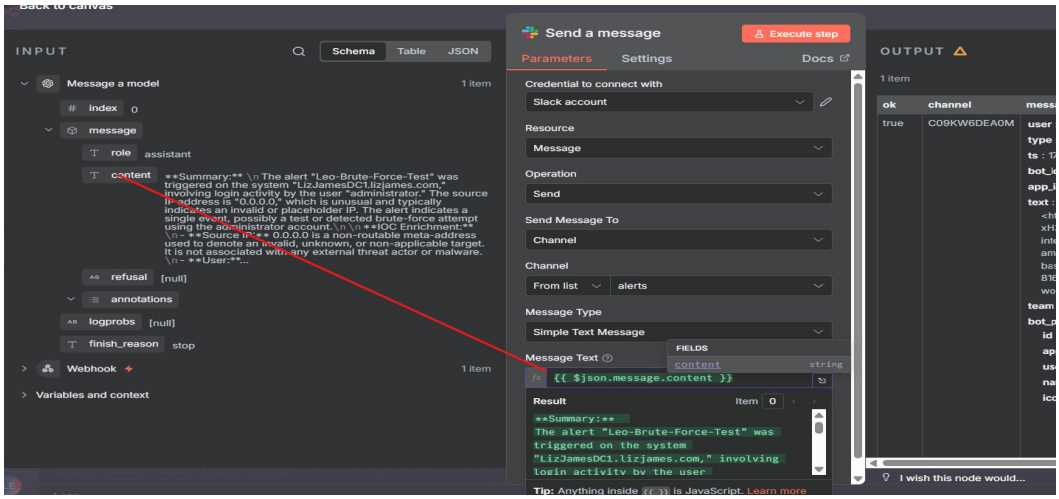
- Now, go back to **SLACK again and click > # Alert** > check if Leo Test is there.



14. Now, the Moment of Truth, we will now Test the workflow automation > Go back to your N8N > click Back to Canvas
    - Click the Message a model > Click Play button seen below.
    - Now, click the Slack > Click Play button.



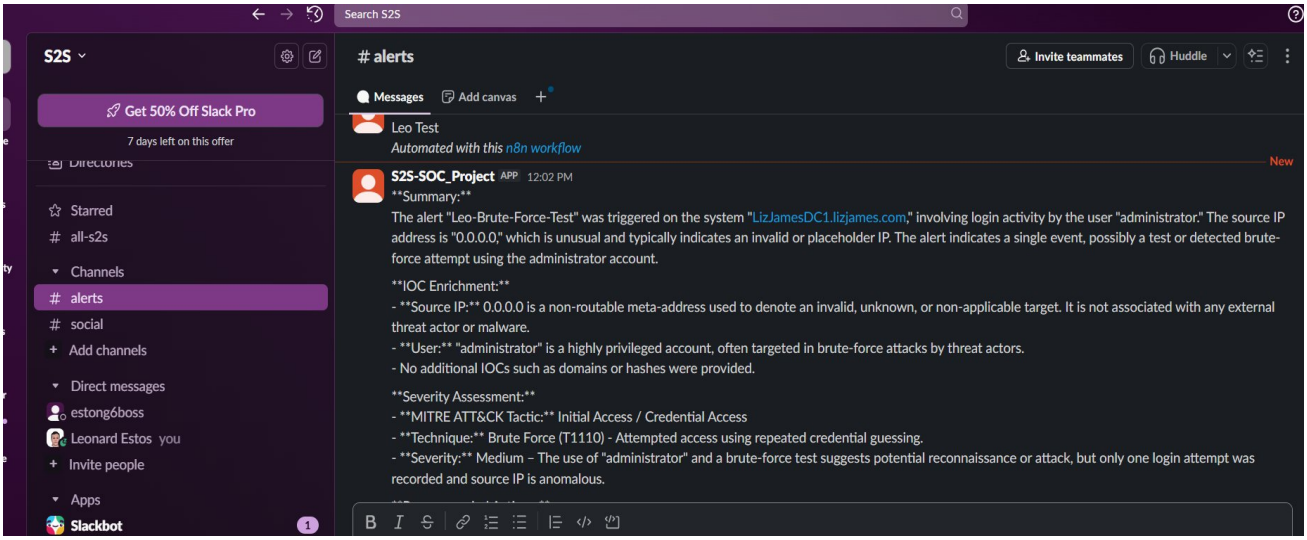
    - Check the **Slack and #Alert** will be available there.
    - Now we need to put the **Content Message in the N8N > In N8N > double click the SLACK and configuration will open again > Drag the content to Message Text > click Execute Step**



- Now, go back to SLACK again > In **# Alert check the results**

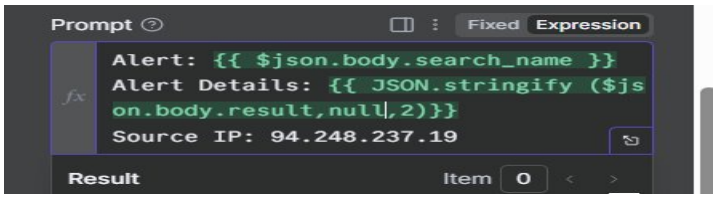
Now for the SOC, this is a nice POC (Proof of Concept)

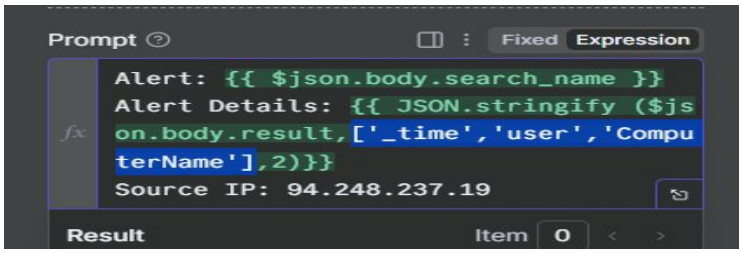
- **POC,** a small-scale project or demonstration that proves an idea's feasibility and viability before committing significant resources. It shows that a concept can be successfully implemented in the real world, validate assumptions, and provide stakeholders with evidence that the project is worth pursuing. A POC is not a final product.



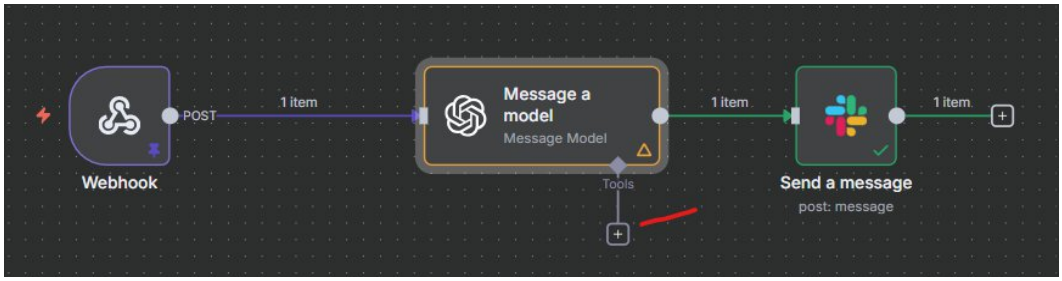
15. Now, we will test this with outside IP address basically those IP's that are used for abuse. Go to www.abuseipdb.com > get some test IP.

- Now, in your N8N > Click the OpenAI (ChatGPT) Message a Model > Go down and go to PROMPT > Add the Source IP: 94.248.237.19 came from abuseipdb.
- Now, change the **Null** with **['_time','user','ComputerName']**



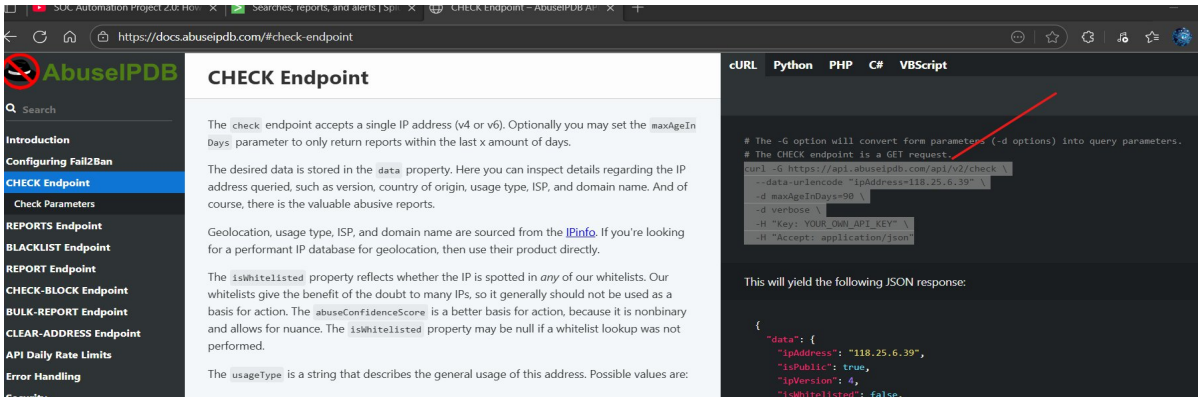


- Go back to your N8N > Click Back to Canvas > Now you can see a Tool below the Message a Model > Double click the + > Click HTTP request Tool

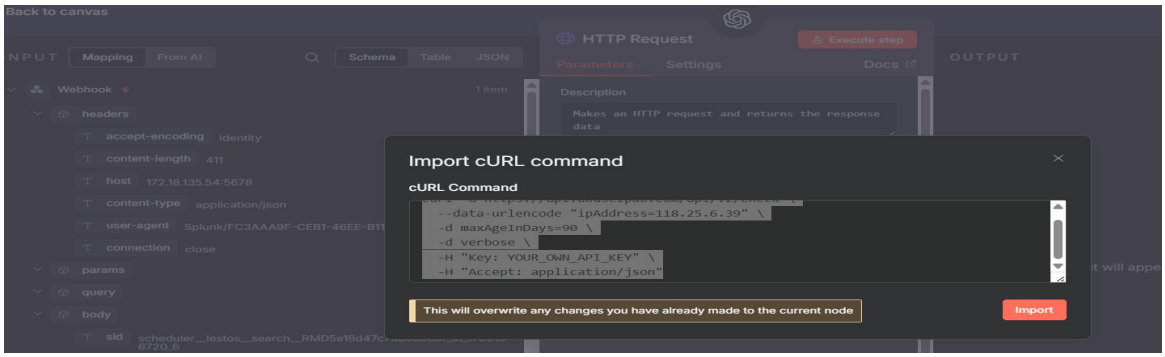


- Now, we need to sign-up in the www.abuseipdb.com to get the API Key. > Login to abuseipdb > Click API > Click Create key > Click CREATE

- Now click > Check out [our manual](#) to learn how to use our API. > Click CHECK Endpoint > Copy the Curl Code, see below

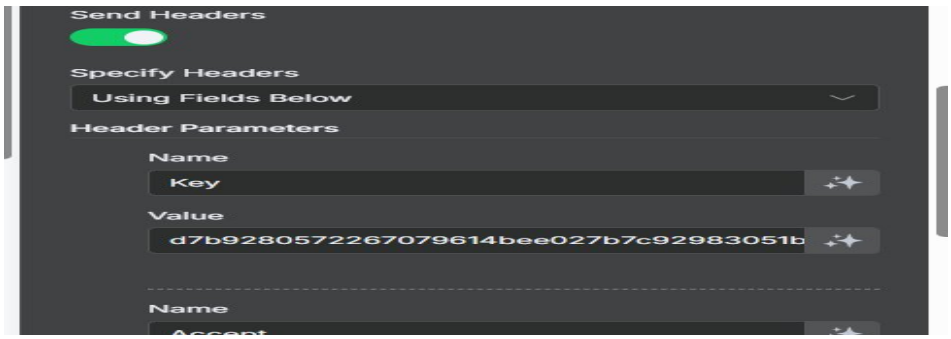


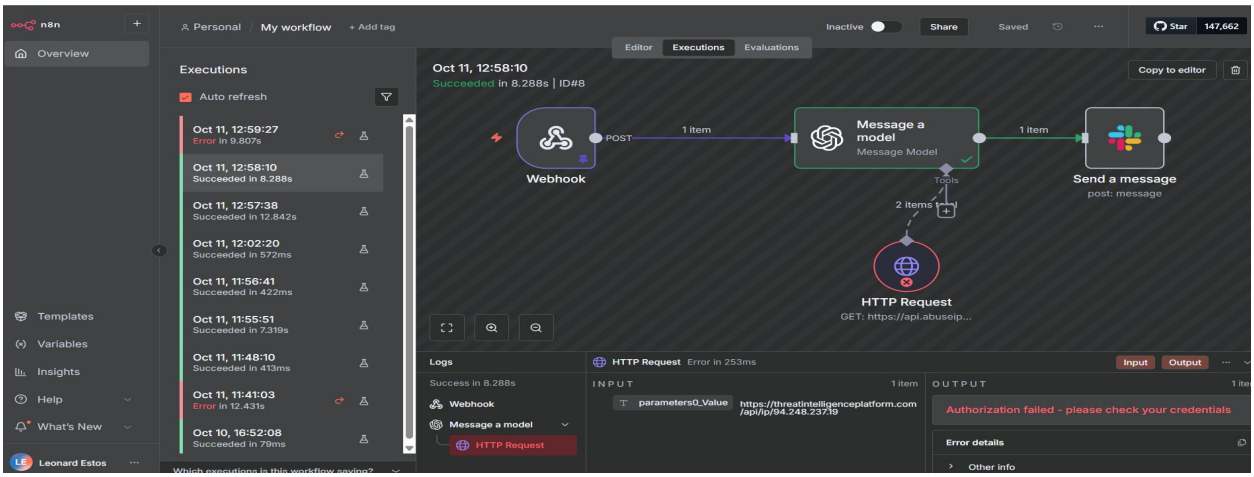
- Now go back to your N8N > click IMPORT and paste the curl code.



- Once imported > **Go down to the Query Parameters > Value removes the hardcoded IP > Change the IP with your test** www.abuseipdb.com **=** 94.248.237.19

- Going down again, see the **Specify Headers/Header Parameters** > in Value: Paste your **abuseipdp API Key**





" When you train Smarter, you defend Stronger"
Leonard Estos