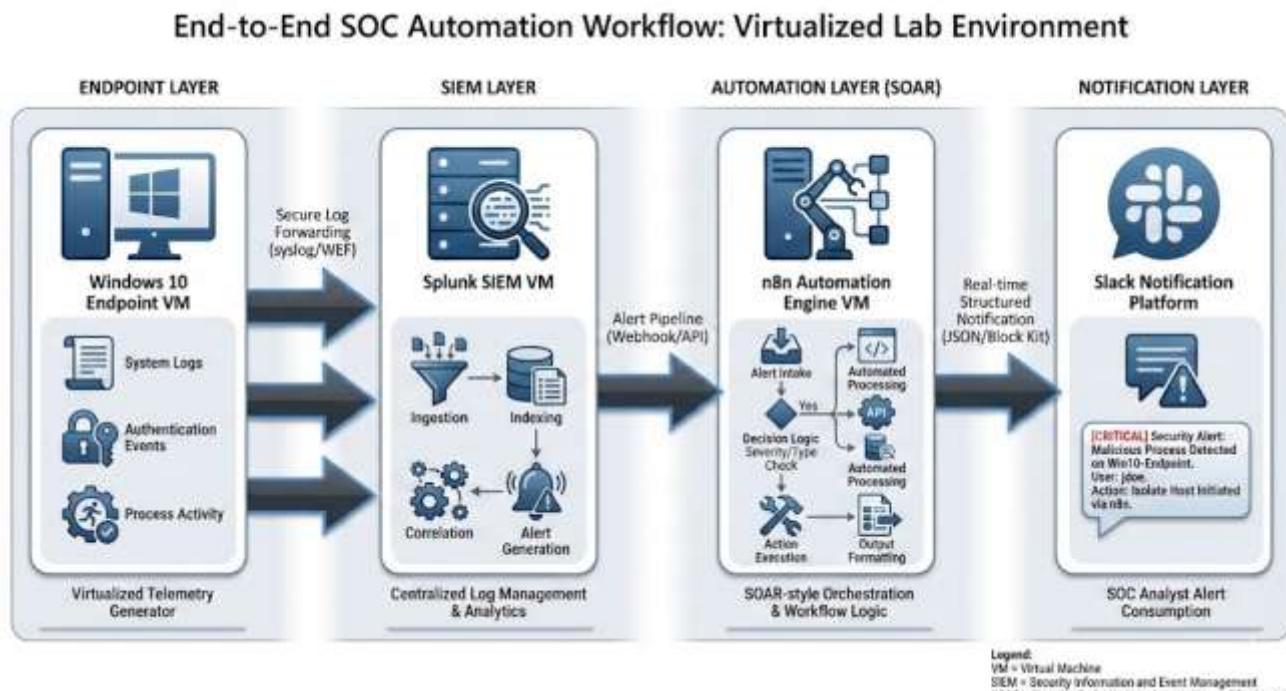


Design and Implementation of an Automated SOC Workflow

Objectives:

The objective of this lab is to demonstrate an end-to-end SOC automation workflow by integrating endpoint telemetry, SIEM detection, and automated response. A Windows 10 system generates security events that are ingested and analyzed by Splunk, where alerts are created based on defined detection logic. These alerts are then forwarded to n8n to trigger automated workflows, resulting in real-time security notifications delivered to Slack. This lab provides practical insight into SIEM–SOAR integration and automated SOC operations.



Step :1 Installing Splunk and Configuring GUI

In this step, Splunk Enterprise is installed on a dedicated virtual machine to serve as the centralized SIEM platform for the lab. After installation, the Splunk web interface is configured to enable access to the management console, verify successful deployment, and prepare the environment for log ingestion, search, and alert configuration in subsequent steps.

The initial phase of the SOC automation workflow begins with provisioning the central SIEM engine. This step involves selecting the Splunk Enterprise 10.2.0 installer specifically for a Linux environment. By choosing the Debian (.deb) package, the setup is optimized for deployment on a Linux-based server (such as Ubuntu), which will serve as the primary instance for data indexing, log analysis, and alert generation.

Choose Your Download

Splunk Enterprise 10.2.0

Index 500 MB/Day. Sign up and download now. After 60 days you can convert to a perpetual free license or purchase a Splunk Enterprise license to continue using the expanded functionality designed for enterprise-scale deployments.

Choose Your Installation Package

Platform	Type	File Extension	Size	Action
Windows				
Linux	64-bit	4.x+, 5.x+, 6.x+ kernel Linux distributions	.tgz	1701.67 MB
			.rpm	1701.65 MB
			.deb	1291.35 MB

Actions: Download Now, Copy wget link, More

The installation is performed on the Linux host by executing the Debian package manager command. This step unpacks the Splunk binaries and prepares the filesystem for the SIEM software deployment.

Command: sudo dpkg -i splunk-10.2.0-d749cb17ea65-linux-amd64.deb

```
safwan@safwan-VirtualBox:~$ sudo dpkg -i splunk-10.2.0-d749cb17ea65-linux-amd64.deb
```

The setup begins by acquiring the Splunk Enterprise 10.2.0 Debian package for Linux, which serves as the central log aggregator. The package is installed via the command line, followed by a transition to the dedicated splunk user environment to initialize the service from the binary directory.

- **Installer:** splunk-10.2.0-linux-amd64.deb (~1.3 GB).
- **Installation:** Executed using sudo dpkg -i to deploy files into the /opt/splunk directory.
- **User Management:** Switched to the service account using sudo -u splunk bash to ensure proper file permissions.

- **Initialization:** Navigated to /opt/splunk/bin and executed ./splunk start to trigger the license agreement and service startup.

```
safwan@safwan-VirtualBox:/opt/splunk$ sudo -u splunk bash
[sudo] password for safwan:
splunk@safwan-VirtualBox:~$ cd /opt/splunk
splunk@safwan-VirtualBox:~$ cd bin
splunk@safwan-VirtualBox:~/bin$ ./splunk start
```

Configuration of the initial administrator username and a secure password (minimum 8 characters) to grant access to the Splunk Web interface.

```
Please enter an administrator username: safwan
Password must contain at least:
 * 8 total printable ASCII character(s).
Please enter a new password:
```

```
safwan@safwan-VirtualBox:/opt/splunk/bin$ sudo ./splunk enable boot-start -user splunk
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
safwan@safwan-VirtualBox:/opt/splunk/bin$
```

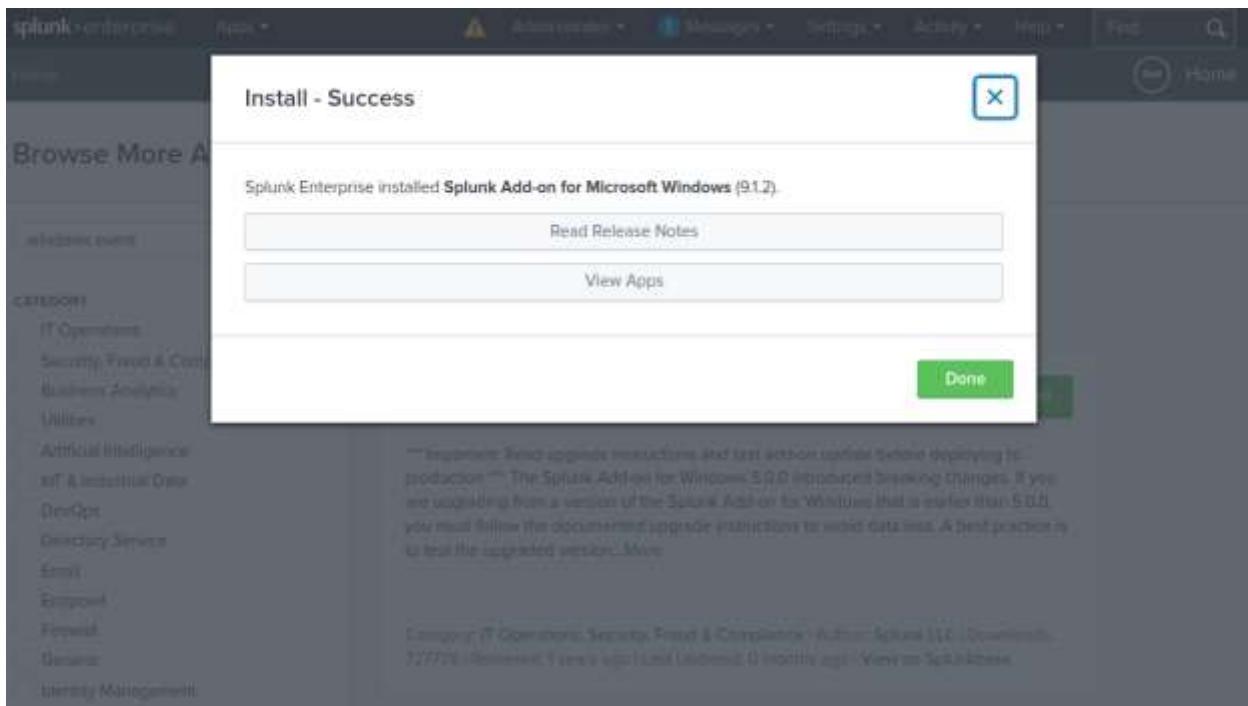
The Splunk Web interface is accessed by navigating to the host's IP address on port 8000 via a web browser. This GUI serves as the primary portal for logging in with the administrative credentials created during installation, enabling the transition from command-line setup to a visual environment for managing security events and automation tasks.



The screenshot shows the Splunk Enterprise home page. At the top, there's a navigation bar with links for 'splunk>enterprise', 'Apps', 'Administrator', 'Messages', 'Settings', 'Activity', 'Help', 'Find', and a search icon. Below the navigation bar, the greeting 'Hello, Administrator' is displayed, along with a link to 'Home page settings'. A horizontal menu bar follows, with 'Bookmarks' being the active tab, followed by 'Dashboard', 'Search history', 'Recently viewed', 'Created by you', and 'Shared with you'. Under the 'Bookmarks' section, there are three expandable categories: 'My bookmarks (0)', 'Shared with my organization (0)', and 'Splunk recommended (16)'. Each category has an 'Add bookmark' button. Below these categories, there are two sections: 'Shared by me' and 'Shared by other administrators'. At the bottom of the page, there are 'Common tasks' and a 'Hide for users' button.

To enable endpoint telemetry ingestion, the **Splunk Add-on for Microsoft Windows** is installed via the Splunk Web GUI. This application provides the necessary knowledge objects to map Windows event logs to the Splunk CIM, ensuring that data from the Windows 10 system is correctly parsed and indexed for analysis.

The screenshot shows the Splunk App Store interface. At the top, there's a navigation bar with links for 'splunk>enterprise', 'Apps', 'Administrator', 'Messages', 'Settings', 'Activity', 'Help', 'Find', and a search icon. Below the navigation bar, the word 'Home' is displayed. The main area features a heading 'Browse More Apps' and a search bar containing the text 'windows event'. To the right of the search bar are buttons for 'Best Match', 'Newest', and 'Popular'. On the left, there's a sidebar titled 'CATEGORY' with a list of checkboxes for various categories: IT Operations, Security, Fraud & Compliance, Business Analytics, Utilities, Artificial Intelligence, IoT & Industrial Data, DevOps, Directory Service, Email, Endpoint, Firewall, Generic, and Identity Management. On the right, there's a list of '5 APPS' with one item visible: 'Splunk Add-on for Microsoft Windows'. This item has a green 'Install' button. Below the app listing, there's a note: '*** Important: Read upgrade instructions and test add-on update before deploying to production *** The Splunk Add-on for Windows 5.0.0 introduced breaking changes. If you are upgrading from a version of the Splunk Add-on for Windows that is earlier than 5.0.0, you must follow the documented upgrade instructions to avoid data loss. A best practice is to test the upgraded version...More'. At the bottom of the app listing, there's a summary: 'Category: IT Operations, Security, Fraud & Compliance | Author: Splunk LLC | Downloads: 727776 | Released: 1 years ago | Last Updated: 0 months ago | View on Splunkbase'.



Part 2: Send Telemetry from Widnows VM to Splunk

In this part, the Windows 10 virtual machine is configured to generate and forward system and security telemetry to the Splunk SIEM. This enables Splunk to ingest endpoint logs, validate data flow, and provide visibility into Windows activity for monitoring and detection purposes.

To enable telemetry collection from the Windows 10 VM, the Splunk Universal Forwarder 10.2.0 is selected for download. The 64-bit Windows .msi installer is chosen to ensure compatibility with modern Windows architectures. This lightweight agent is dedicated to harvesting local system logs and securely forwarding them to the Splunk indexer for analysis.

Choose Your Download

Splunk Universal Forwarder 10.2.0

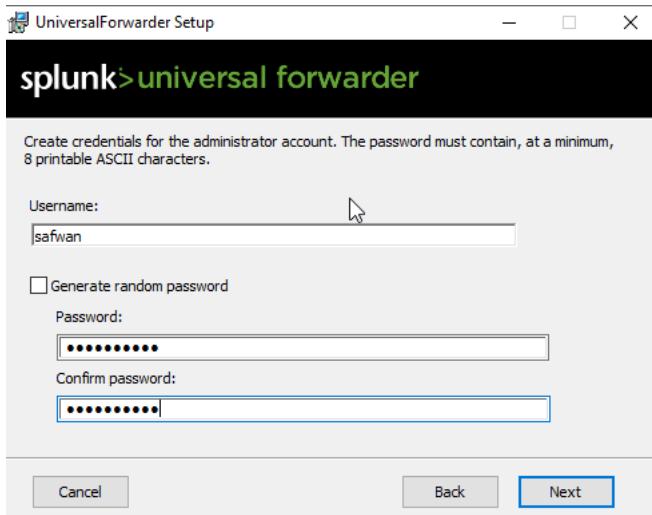
Universal Forwarders provide reliable, secure data collection from remote sources and forward that data into Splunk software for indexing and consolidation. They can scale to tens of thousands of remote systems, collecting terabytes of data.

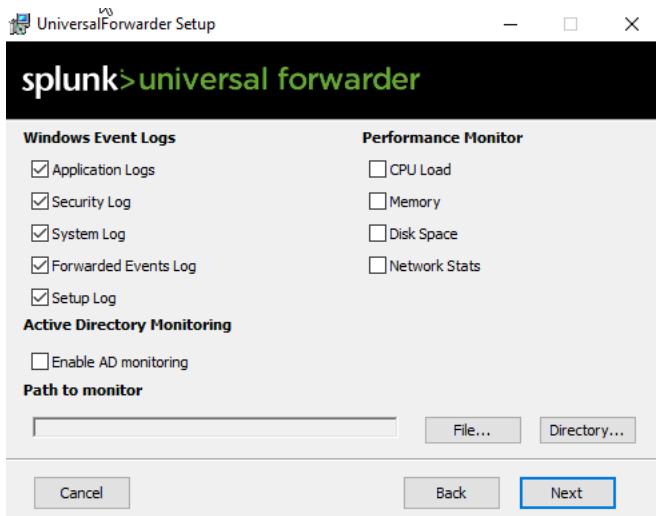
Choose Your Installation Package

The screenshot shows the Splunk download page for the Universal Forwarder. At the top, there are tabs for Windows, Linux, Mac OS, FreeBSD, Solaris, and AIX. Below these, two Windows packages are listed:

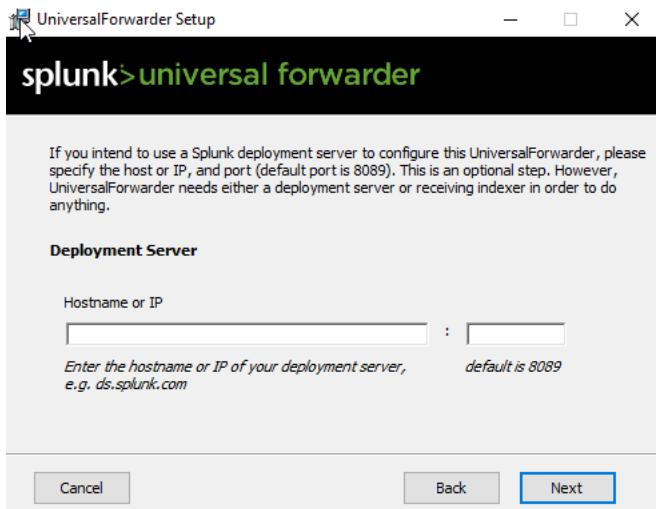
- Windows 10**: 32-bit, .msi, 64.9 MB. Includes "Download Now" and "Copy wget link" buttons.
- Windows 10, 11
Windows Server 2019,
2022, 2025**: 64-bit, .msi, 186.70 MB. This item is highlighted with a red border and includes "Download Now" and "Copy wget link" buttons.

After downloading , installation process starts. In this follow the exact same steps as per below figures displayes for poper configuration.



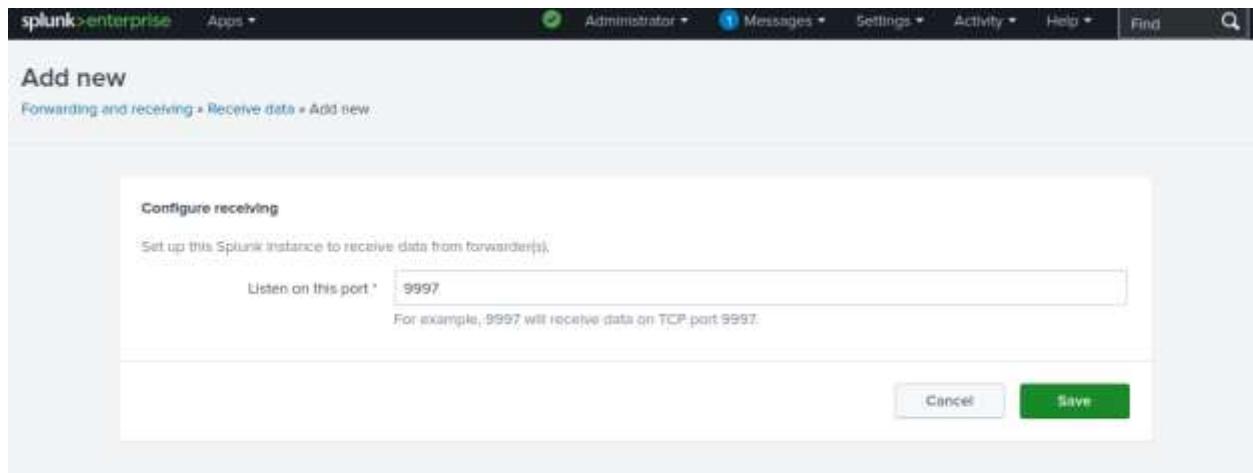


After this step Receiver Server option will come, so on that add the Splunk VM IP Address with 9997 port, as the Splunk listens for the telemetry at port 9997.



To prepare the SIEM for incoming endpoint telemetry, a receiver is configured within the Splunk Web interface. By navigating to Settings > Forwarding and receiving, a new receiving port is defined to listen for traffic from remote forwarders.

The standard **port 9997** is specified and saved, enabling the Splunk indexer to ingest data streams from the Windows VM. This configuration is vital for establishing the communication channel required for real-time log analysis and subsequent SOC automation.



In this part add the IP Address of the Windows VM in the Source Type Description.

Input Settings

Optionally set additional input parameters for this data input as follows:

Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Source Type	<input type="button" value="Select"/> <input type="button" value="New"/>
	<input type="text" value="windowslog"/>
Source Type Category	<input type="button" value="Custom"/>
Source Type Description	<input type="text"/>

New Index

General Settings

Index Name Set index name (e.g., INDEX_NAME). Search using index=INDEX_NAME.

Index Data Type Events Metrics
The type of data to store (event-based or metrics).

Home Path Hot/warm db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/db).

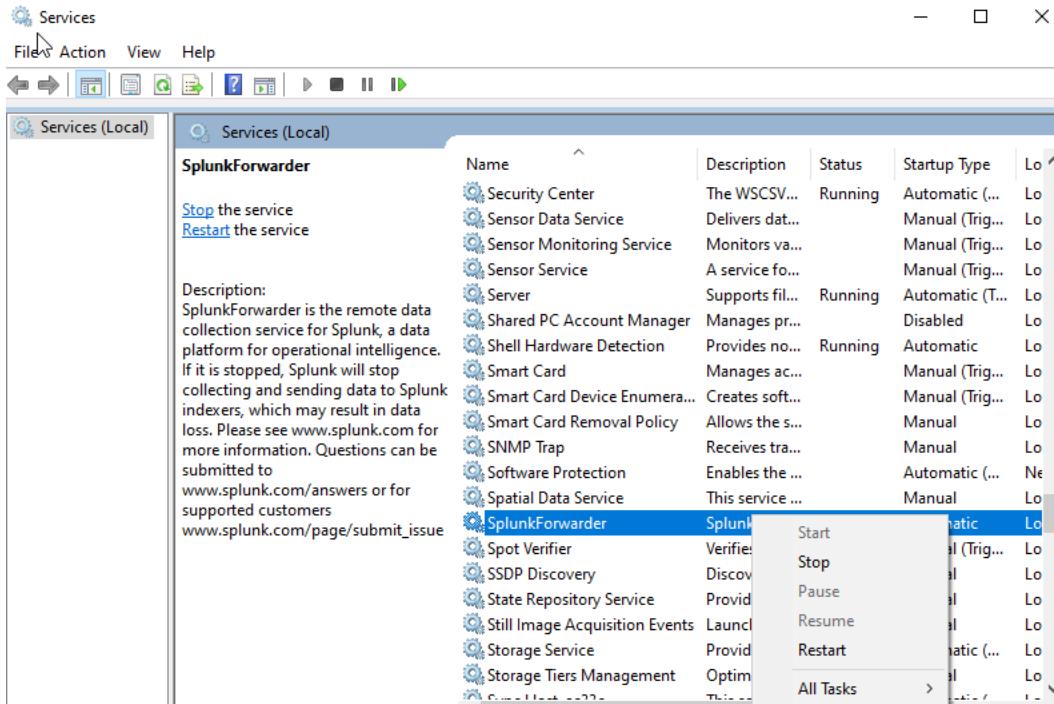
Cold Path Cold db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/colddb).

Thawed Path Thawed/resurrected db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/thaweddb).

Data Integrity Check Enable Disable
Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.

Save **Cancel**

Then go to the services in Windows and search for SplunkForwarder. Right click on it and restart it and ensure it should run smoothly. As this is a very crucial part of the process, if this is not configured properly the telemetry won't be sent over to the Splunk GUI.



The final stage of the telemetry setup involves verifying that the Windows event logs are successfully reaching the Splunk indexer. By executing a global search for `index=*`, the Search & Reporting interface confirms the ingestion of real-time data from the Windows 10 endpoint.

- **Verified Source:** Telemetry is being received from the WinEventLog:Security source.
- **Event Confirmation:** Successful capture of critical security events, such as EventCode 4672 (Special Logon) and EventCode 4624 (An account was successfully logged on).
- **Host Identification:** The data is correctly mapped to the target Windows machine (DESKTOP-6VKP16F), indicating a healthy connection between the Universal Forwarder and the SIEM.

The screenshot shows the Splunk Enterprise search interface. The search bar at the top contains the query `index=*`. Below the search bar, a message indicates "779 of 779 events matched". The main area displays a histogram of event times and a list of 20 events. Two specific events are highlighted in green:
Event 1: Time: 2/7/26 1:03:07 AM, Event: 02/06/2026 12:03:07, 187 PM, LogName:Security, EventCode:4672, EventType:0, ComputerName:DESKTOP-6VKP16F, host = DESKTOP-6VKP16F, source = WinEventLog:Security, sourcetype = WinEventLog.
Event 2: Time: 2/7/26 1:03:07 AM, Event: 02/06/2026 12:03:07, 187 PM, LogName:Security, EventCode:4624, EventType:0.

Part 3: Installing n8n

In this part, the n8n automation platform is installed on a dedicated virtual machine. This sets up the environment for creating automated workflows that can process alerts from Splunk and trigger security responses or notifications.

Make separate directory for n8n configuration

```
safwan@safwan-VirtualBox:~$ mkdir n8n
safwan@safwan-VirtualBox:~$ cd n8n
safwan@safwan-VirtualBox:~/n8n$ sudo nano docker-compose.yaml
```

The installation of n8n is managed through Docker Compose, providing a consistent and isolated environment for the automation platform. By utilizing a YAML configuration file, the service is defined with persistent storage and specific networking parameters required for SOC orchestration.

- **Docker Image:** Utilization of the n8nio/n8n:latest image to ensure the platform remains up-to-date with the latest security features and nodes.
- **Port Mapping:** Binding the container's internal service to port 5678 on the host machine, enabling web browser access to the workflow editor.
- **Environment Variables:** Configuration of critical host details, including the Host Machine IP, protocol (HTTP), and timezone (America/Toronto) to ensure accurate log timestamps in automated responses.
- **Data Persistence:** Mapping a local volume (./n8n_data) to the container's internal storage to preserve workflows, credentials, and configuration data across service restarts.

```
GNU nano 7.2                                     docker-compose.yaml *
services:
  n8n:
    image: n8nio/n8n:latest
    restart: always
    ports:
      - "5678:5678"
    environment:
      - N8N_HOST=(Host Machine IP)
      - N8N_PORT=5678
      - N8N_PROTOCOL=http
      - N8N_SECURE_COOKIE=false
      - GENERIC_TIMEZONE=AMERICA/TORONTO
    volumes:
      - ./n8n_data:/home/node/.n8n
```

The sudo docker-compose pull command is executed to download the specific n8n image layers defined in the configuration file. This step ensures all necessary dependencies and the latest platform binaries are present on the local host before the container is officially started.

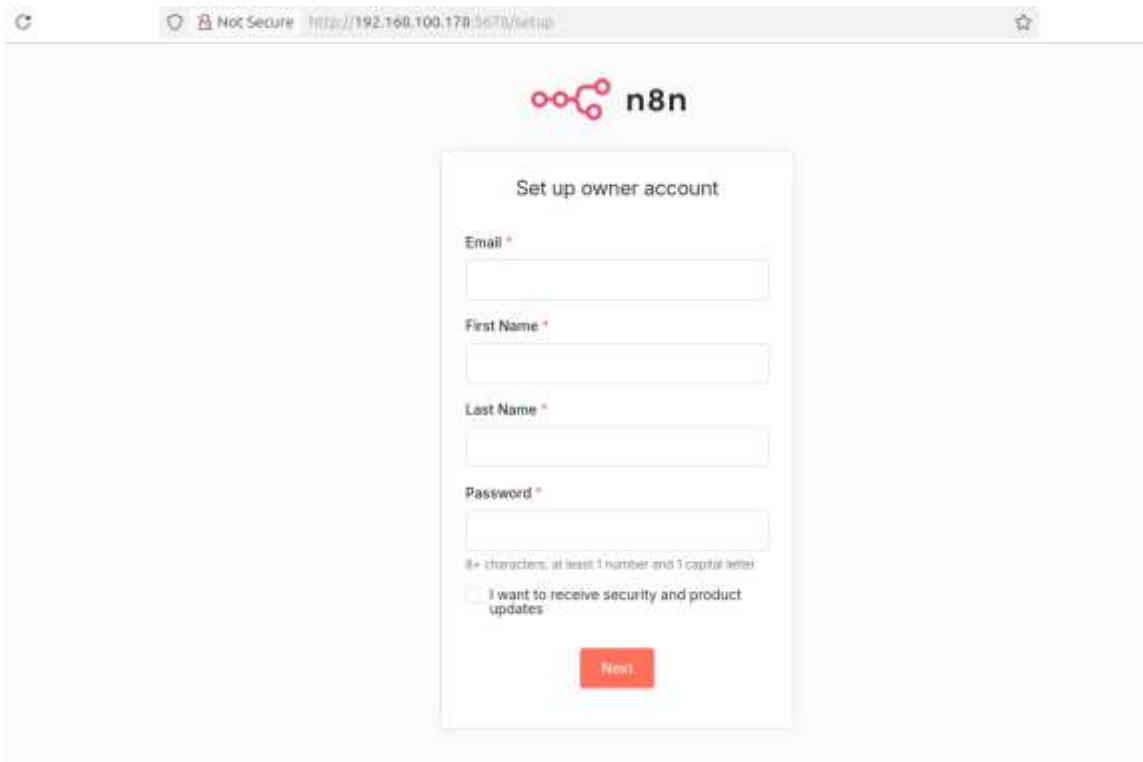
```
safwan@safwan-VirtualBox:~/n8n$ sudo docker-compose pull
```

The final step in the containerized deployment is the execution of the sudo docker-compose up -d command. This instruction initializes the n8n container in detached mode, allowing the automation platform to run in the background as a persistent service. Once active, the environment is ready for web-based access to begin workflow orchestration.

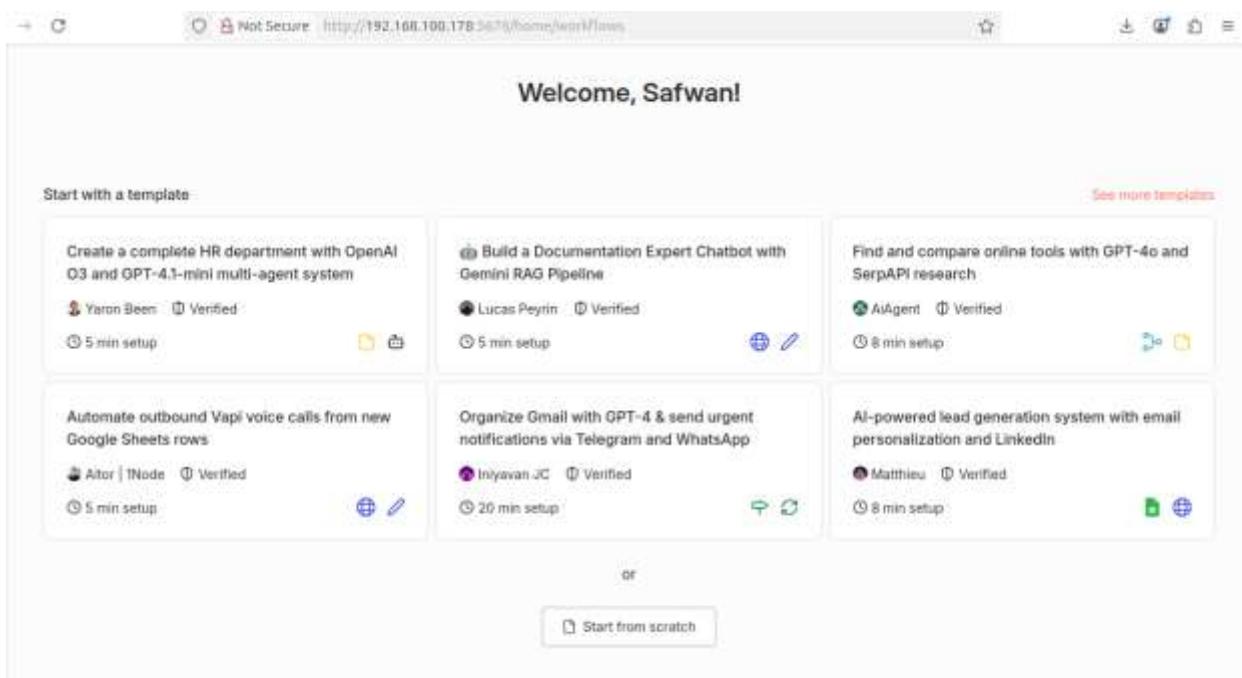
```
safwan@safwan-VirtualBox:~/n8n$ sudo docker-compose up -d
```

Once the container is running, the n8n web interface is accessed by navigating to the Host Machine IP on port 5678 via a web browser. This launches the initial setup wizard, which is the final step in preparing the automation environment for workflow creation.

- **Initial Access:** Navigating to http://<Host_IP>:5678 connects the browser to the n8n service running inside the Docker container.
- **Owner Account Creation:** The "Set up owner account" screen requires the registration of an administrative user with a valid email, name, and a secure password (minimum 8 characters with numbers and capital letters).



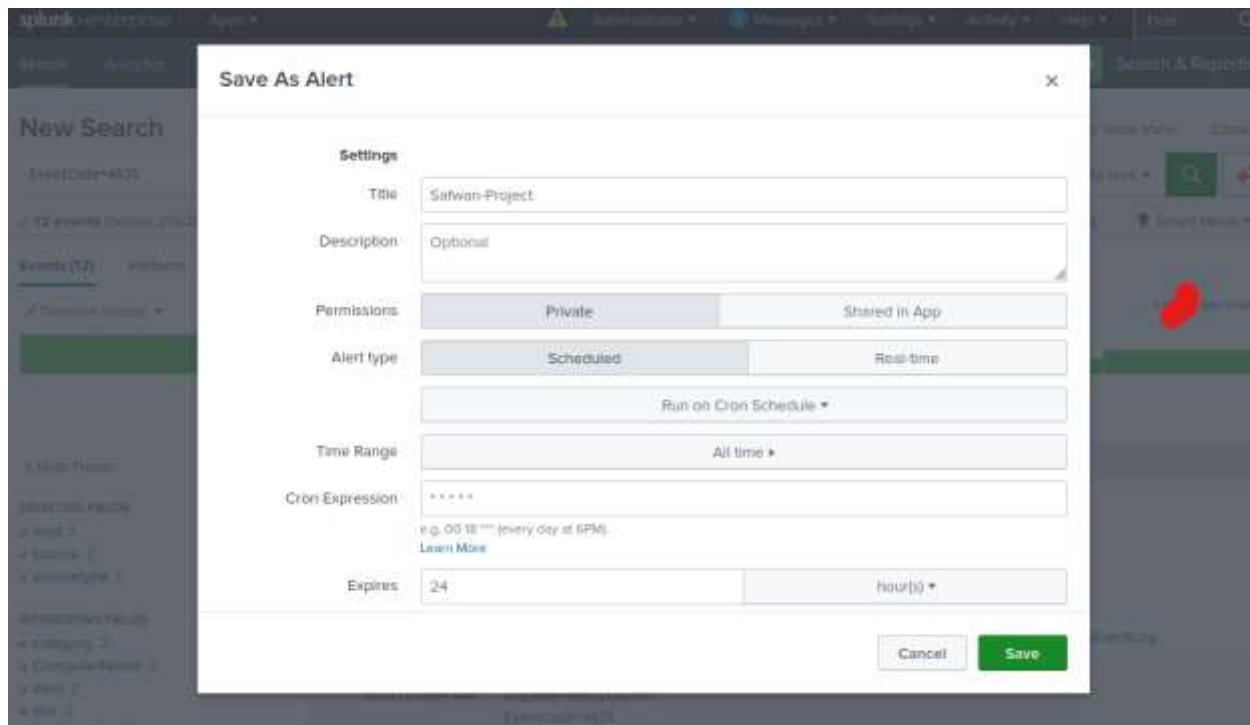
After the successful account creation the default GUI of n8n opens.



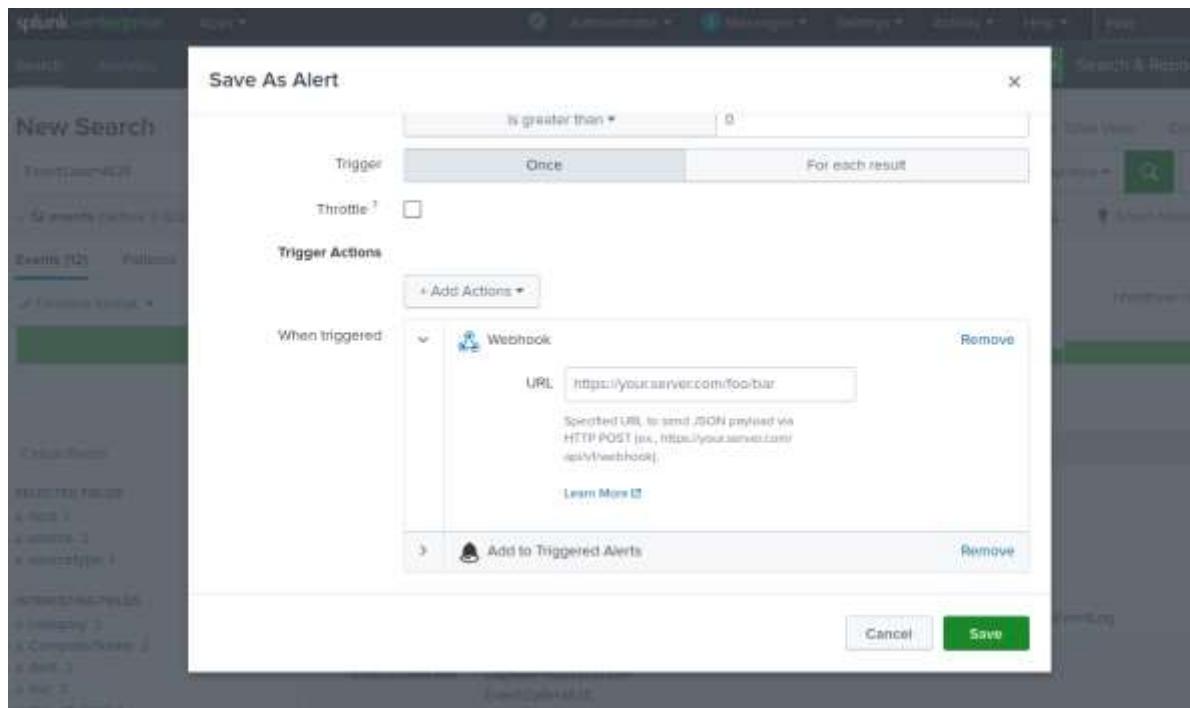
Part 4: Building Automation Workflow

In this part, an automated workflow is created in n8n to process alerts received from Splunk. The workflow defines the logic for handling security events and triggers corresponding actions, such as sending notifications to Slack, simulating a real-world SOC response process.

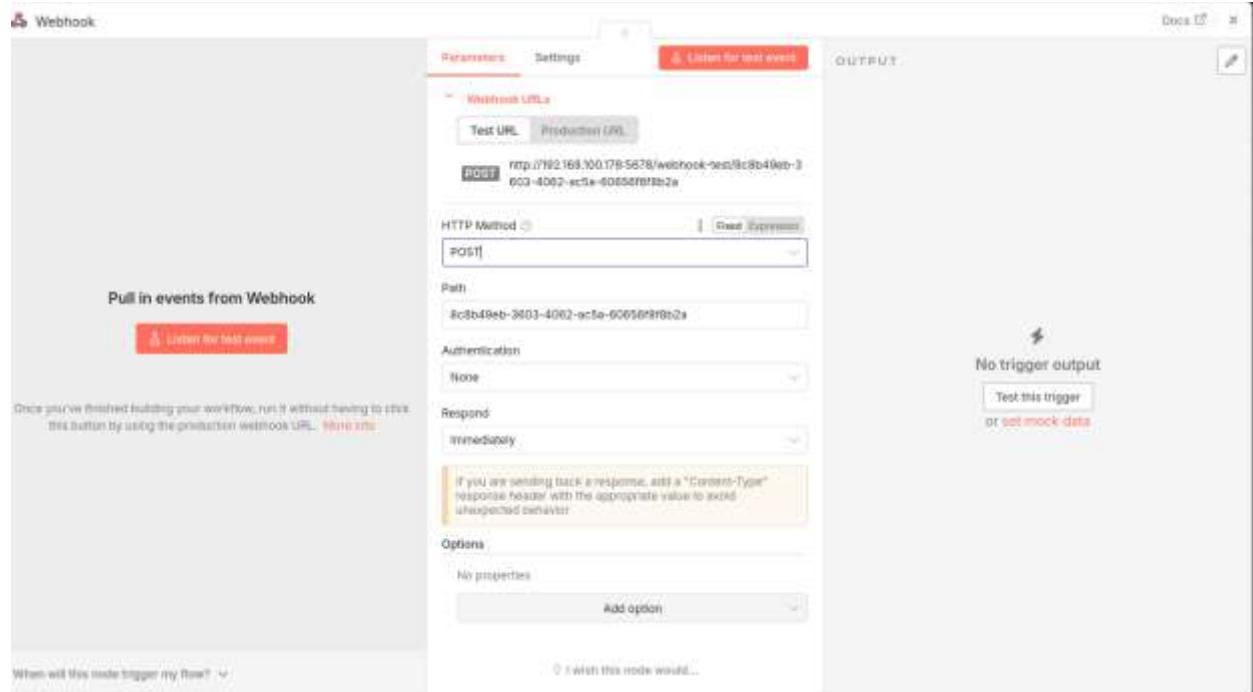
The first step in the automation workflow involves creating a scheduled alert in Splunk to detect specific security events. In the **Save As Alert** dialog, the search for EventCode=4625 (failed login attempts) is saved with the title "Safwan-Project". The alert is configured to run on a **Cron Schedule** of * * * * * (every minute) and is set to expire after 24 hours, ensuring continuous monitoring of the endpoint.



To receive these alerts, a **Webhook** node is configured within the n8n interface. The HTTP Method is set to **POST**, and the node generates a unique **Test URL** (e.g., [http://192.168.100.178:5678/webhook-test/...](http://192.168.100.178:5678/webhook-test/)). This URL serves as the listener that will accept the JSON payload sent from Splunk whenever the alert is triggered.



To receive these alerts, a Webhook node is configured within the n8n interface. The HTTP Method is set to POST, and the node generates a unique Test URL (e.g., <http://192.168.100.178:5678/webhook-test/>). This URL serves as the listener that will accept the JSON payload sent from Splunk whenever the alert is triggered.



The final integration step connects the two platforms. Back in the Splunk alert settings under Trigger Actions, the "Webhook" option is selected. The unique Test URL generated by n8n is then pasted into the URL field. This configuration ensures that every time the alert fires, Splunk automatically sends the event data to the n8n automation flow for processing.

Safwan-Project

Enabled: Yes, Disable

App: search

Permissions: Private, Owned by safwan, Edit

Modified: Feb 6, 2026 12:47:21 PM

Alert Type: Scheduled, Cron Schedule, Edit

Trigger Condition: Number of Results is > 0, Edit

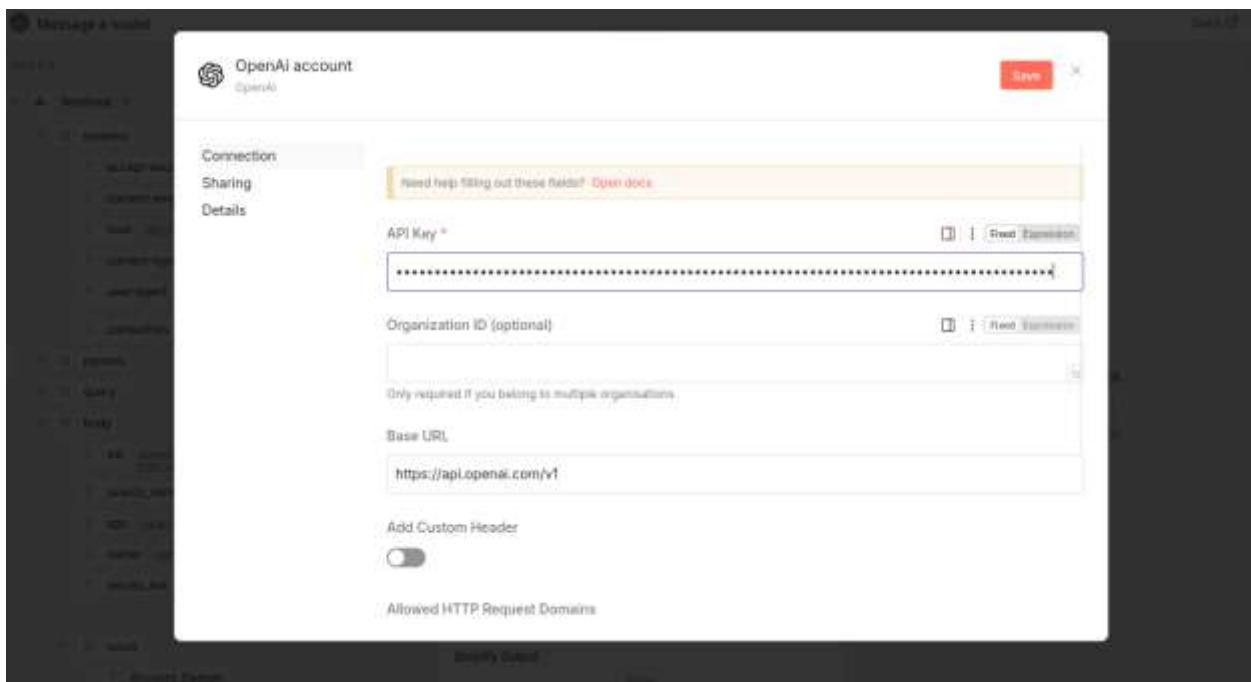
Actions: 2 Actions, Edit

- Add to Triggered Alerts
- Webhook

Edit ▾

This screenshot shows the n8n interface for a project named 'Safwan-Project'. It displays various configuration details such as trigger conditions and actions. A prominent 'Edit' button is visible in the top right corner.

This step establishes the secure connection between the n8n automation platform and the OpenAI API. A new OpenAI account credential is created within the workflow, where the secret API Key is input and saved. This authentication allows the workflow to programmatically access GPT models for analyzing security alerts.



The OpenAI node is configured to define the AI's operational role within the SOC workflow. Under the Messages section, an "Assistant" role is created with a specific system prompt: *"You are an automated cybersecurity analysis engine integrated into an event-driven workflow."* This instruction sets the context, ensuring the model interprets subsequent data as a security analyst rather than a general-purpose chatbot.

The screenshot shows the "Message a model" interface. On the left, the "INPUT" panel displays a webhook payload in JSON format. The payload includes headers (accept-encoding: identity, content-length: 3900, host: 192.168.100.178:5878, content-type: application/json, user-agent: Splunk/9C0B7F4B-FD8E-45C1-BB3D-16AACC3881A7, connection: close), parameters, query (search_name: Safwan-Project), body (id: scheduler_safwan_search_RMD5b612ed9a97393_ae1770364140_2, search_name: Safwan-Project, app: search, owner: safwan, results_link: http://safwan:VirtualBox:8000/app/search/2007-02-07/cache/scheduler_safwan_search_RMD5b612ed9a97393_ae1770364140_2), and a result section. On the right, the "Parameters" tab is selected, showing fields for "Role" (OpenAI account) and "Prompt". The prompt template is: "You are an automated cyber security analysis engine integrated into an event-driven workflow. Alert : {{ \$json.body.search_name }}". The "OUTPUT" panel shows a message step with a "Text" type and a role of "Assistant". A note indicates "No output data" and provides options to "Execute step" or "Set mock data".

The workflow logic is further refined by mapping incoming Splunk data to the AI's input prompt. A User role is added to the message parameters, utilizing a dynamic expression: Alert : {{ \$json.body.search_name }}. This configuration ensures that the specific title of the triggered Splunk alert (e.g., "Safwan-Project") is automatically injected into the prompt, telling the AI exactly which security event it is analyzing.

This screenshot shows the same "Message a model" interface after updating the prompt. The "Parameters" tab now includes a "Role" field set to "User". The "Prompt" field contains the updated template: "Alert : {{ \$json.body.search_name }}". The "Result" field shows the value "Alert : Safwan-Project". The "OUTPUT" panel remains the same, showing a message step with a "Text" type and a role of "Assistant". A note indicates "No output data" and provides options to "Execute step" or "Set mock data".

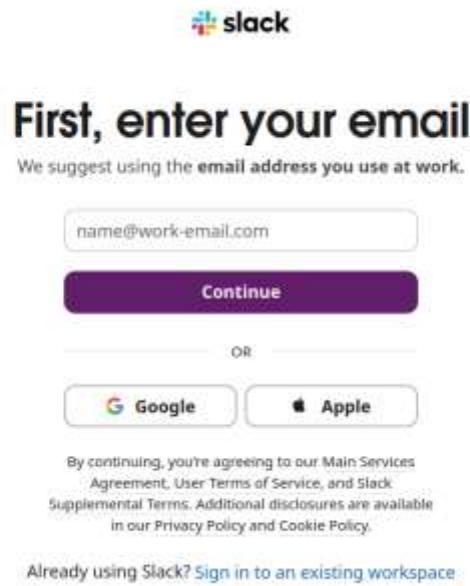
The final image demonstrates the construction of the full data payload passed to the AI model. An n8n Expression is used to stringify the entire JSON result from Splunk: `{JSON.stringify($json.body.result,null,2)}`. The preview window confirms that critical telemetry—including EventCode 4625 (failed login), ComputerName (DESKTOP-6VKP16F), and timestamp data—is correctly formatted and ready to be processed by the AI for incident analysis.

```

Search previous nodes: Results
Expression
Anything inside {{ }} is JavaScript. Learn more
Result
Item: 
X
Text HTML Markdown
connection
params
query
body
  Alert : ({{ $json.body.search_name }})
  Alert_Details : ({{ JSON.stringify($json.body.result,null,2) }})
  Alert : "Safwan-Project"
  Alert_Details : {
    "Account_Domain": "...",
    "Account_Name": "...",
    "Authentication_Package": "...",
    "Caller_Computer_NetBIOS_Name": "...",
    "Caller_Domain": "...",
    "Caller_Logon_ID": "...",
    "Caller_Machine_NetBIOS_Name": "...",
    "Caller_Process_NetBIOS_Name": "...",
    "Caller_User_Name": "...",
    "CategoryString": "...",
    "Change_Type": "...",
    "Client_Address": "...",
    "Client_Breakaway": "...",
    "Client_Logon_ID": "...",
    "Client_Machine_NetBIOS_Name": "...",
    "Client_User_Name": "...",
    "ComputerName": "DESKTOP-6VKP16F",
    "Creator_Process_NetBIOS_Name": "...",
    "Description": "...",
    "Domain": "...",
    "Error_Code": "...",
    "EventCode": "4625",
    "EventType": "...",
    "Failure_Reason": "...",
    "File_Name": "...",
    "File_Path": "...",
    "Group_Domain": "...",
    "User_NetBIOS_Name": "..."
  }
  results_link: http://Redwan-VirtualBox:8000/_app/search/_search?_id=<redacted>_safwan_search_RH0D3b6157e8ba2e87203.a37703943402
result
  Account_Domain
  Account_Name
  Authentication_Package
  Caller_Computer_NetBIOS_Name
  Caller_Domain
  Caller_Logon_ID

```

Configuration of Slack:



Click on Channels to add new channel for the smooth configuration of lab workflow.

The screenshot shows the Slack desktop application interface. On the left, there's a sidebar with various icons and sections like Home, Direct messages, and Apps. A dropdown menu is open over a channel named '# new-channel'. The main workspace shows a message about channels, three cards for inviting teammates, welcoming the team, and connecting apps, and a message from a user joining the channel. At the bottom, there's a message input field and a toolbar.

Create a channel

X

Name

safwan-project

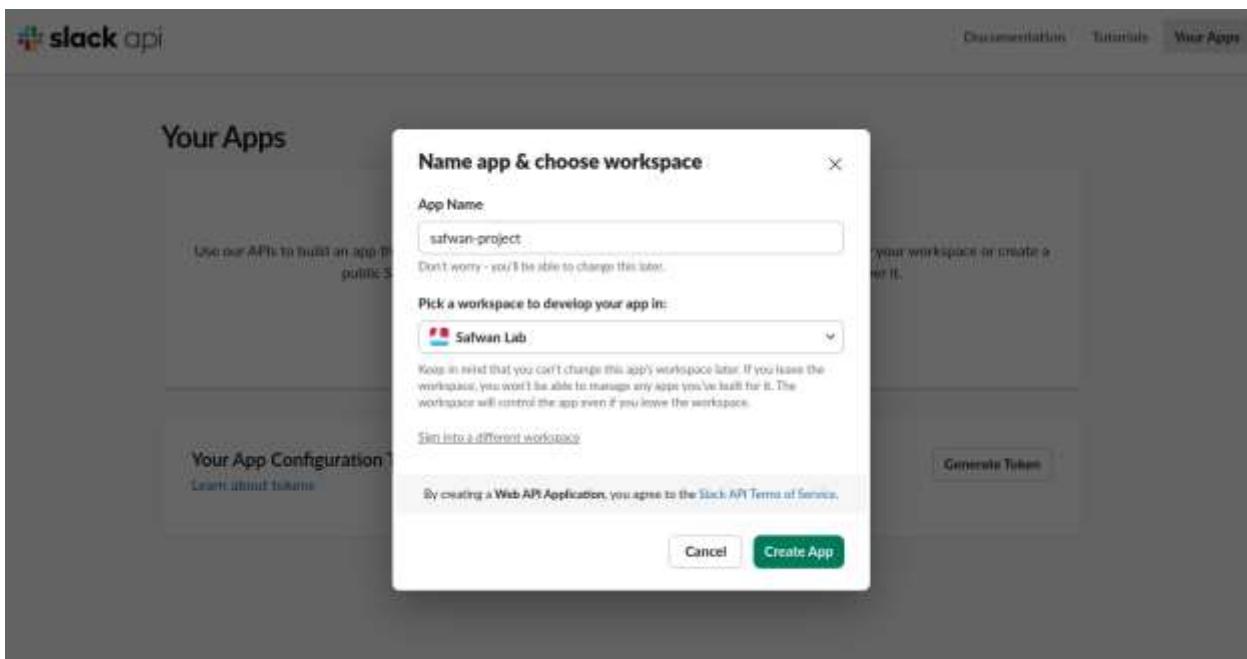
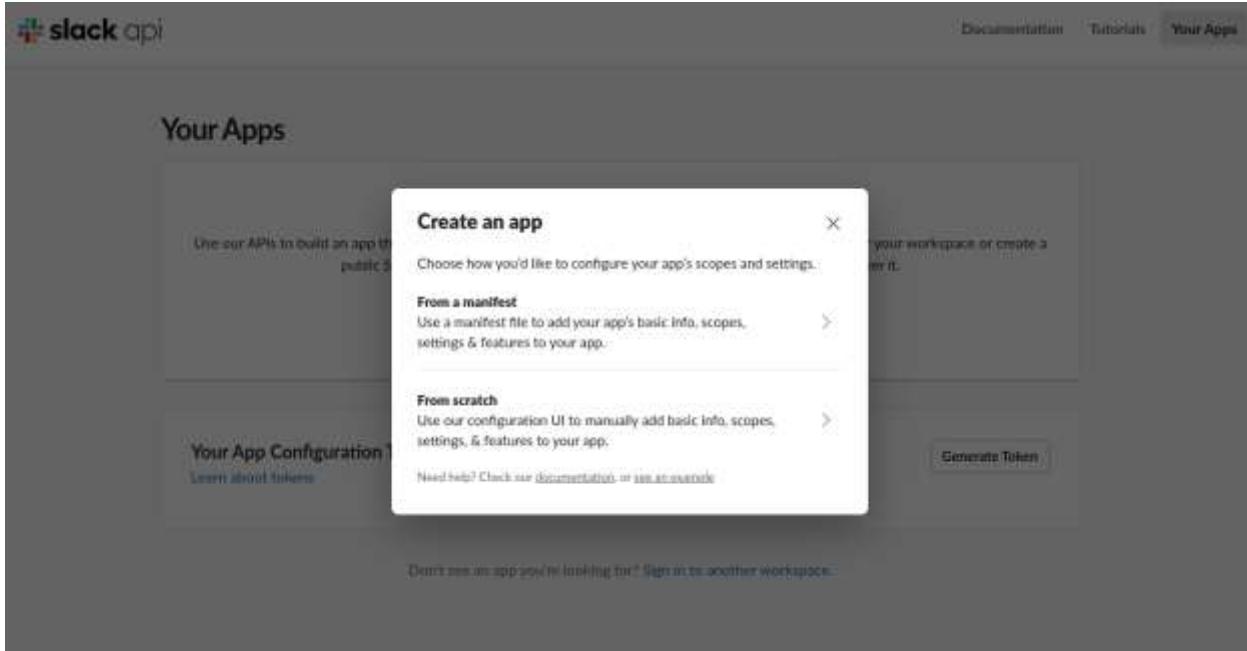
66

Channels are where conversations happen around a topic. Use a name that is easy to find and understand.

Step 1 of 2

Next

The Slack API dashboard by selecting "**Create New App**" from scratch. The application is given a name, such as "SIEM-Alerts," and linked to the active project workspace. This creates the bot identity that will be responsible for delivering automated notifications from the SIEM.



Go to Scopes and then add all of the below scopes in the Add and OAuth Scope section.

Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	⋮
channels:read	View basic information about public channels in a workspace	
Add an OAuth Scope		

User Token Scopes

Scopes that access user data and act on behalf of users that authorize them.

OAuth Scope	Description	⋮
You haven't added any OAuth Scopes for your User token.		
Add an OAuth Scope		

chat:read	View basic information about public channels in a workspace	
chat:write	Send messages as @slack-project	
im:read	View basic information about direct messages that "slack-project" has been added to	
im:write	View basic information about group direct messages that "slack-project" has been added to	
reactions:read	View emoji reactions and their associated content in channels and conversations that "slack-project" has been added to	
reactions:write	Add and edit emoji reactions	
usersgroups.read	View user groups in a workspace	
usersgroups.write	Create and manage user groups	
users.profile.read	View profile details about people in a workspace	
users.read	View people in a workspace	
files:read	View files shared in channels and conversations that "slack-project" has been added to	
files:write	Upload, edit, and delete files in "slack-project"	
groups:read	View basic information about private channels that "slack-project" has been added to	

Upon Saving the previous work the new instance appear. In this click on the Install to “Safwan” Lab (Your case would be change) to get the API Key.

The screenshot shows the 'OAuth & Permissions' section of the Slack developer console. On the left, there's a sidebar with 'Basic Information', 'Collaborators', 'Socket Mode', 'Install App', 'Manage Distribution', 'Features' (with sub-options like 'App Home', 'Agents & AI Apps', 'Work Object Previews...', 'Workflow Steps', 'Org Level Apps', 'Incoming Webhooks', 'Interactivity & Shortcuts', 'Slash Commands', 'Steps from Apps', and 'OAuth & Permissions'), and 'OAuth & Permissions' which is currently selected. The main content area has a heading 'Advanced token security via token rotation' with a note about redirect URLs and a 'Opt In' button. Below that is a section for 'OAuth Tokens' with a note about automatic generation and an 'Install to Safwan Lab' button.

API Key that was taken from the previous step will be putted in the Slack node of n8n.

The screenshot shows the 'Slack account' configuration screen in n8n. It includes tabs for 'Connection', 'Sharing', and 'Details'. Under 'Connection', there's a note about filling out fields, a 'Save' button, and a 'Need help filling out these fields? Open docs' link. The 'Connect using' section has two options: 'Access Token' (selected) and 'OAuth2'. The 'Access Token' field contains a redacted value. The 'Signature Secret' field is also redacted. A note at the bottom of the 'Access Token' field says 'We strongly recommend setting up a signing secret to ensure the authenticity of requests.' The 'Allowed HTTP Request Domains' dropdown is set to 'All'.

Send a message

INPUT

Schema Table JSON Parameters Settings **Execute step**

Credential to connect with: Slack account

Resource: Message

Operation: Send

Send Message To: Channel

From list: alt-satwan-tab

Message Type: Simple Text Message

Message Text: Text

Options: No properties

Add option

OUTPUT

No output data

Execute step or set mock data

Send a message

INPUT

Schema Table JSON Parameters Settings **Execute step**

Credential to connect with: Slack account

Resource: Message

Operation: Send

Send Message To: Channel

From list: alt-satwan-tab

Message Type: Simple Text Message

Message Text: Text

Options: No properties

Add option

OUTPUT

Items

ok	channel	message
true	COADTB7P5T7	user: U0ADEFRNF008 type: message ts: 1730375829.212839 bot_id: B0A0B35GCJH app_id: ADACZ2B4XCP text: Test!...Automated with this. http://192.168.100.57:5078/workflow/LuAGhDreyq0f8k-HULPo7fm_aeuicam-internal&amp;utm_medium=powered_by&amp;gt;+Inbox base_url: https://52.253.89.90:84204<75157>/n/workflow/ team: T0AE8N00998 bot_profile: id: B0A0B35GCJH app_id: ADACZ2B4XCP user_id: U0ADEFRNF008 name: satwan-project icons: image_32 : https://a.slack-edge.com/plugins/app/bot_32.png image_48 : https://a.slack-edge.com/plugins/app/bot_48.png image_72 : https://a.slack-edge.com/

This screenshot demonstrated the proper working of the complete Lab workflow. The Test shown in the Slack which was created in the slack n8n node.s

all-safwan-lab

Messages Add canvas +

safwanawan099 2:58 PM joined #all-safwan-lab.

Today ▾

Only visible to you

Slackbot LEGACY 3:27 PM Invites have been sent! Kick back and relax while you wait for them to join. (198 KB) ▾


YAY! CONGRATULATIONS!!!

safwan-project APP 3:27 PM was added to #all-safwan-lab by safwanawan099.

safwan-project APP 3:27 PM Test
Automated with this n8n workflow

