Final Project Report

Now that we have completed the **setup, detection, alerting, and testing**, it's time to create a detailed **final project report**. The report will document:

1. **Project Overview**
2. **System Setup** (Kali Linux, Windows VM, Splunk)
3. **Data Collection & Configuration**
4. **Brute Force Attack Detection**
5. **Testing & Results**
6. **Conclusion & Future Improvements**

I will provide each section one by one. Let's start with **Step 1: Project Overview**.

📌 Project Overview

Title:

Enterprise-Level Cybersecurity Incident Response System

1. Introduction

Cybersecurity Incident Response Systems (CIRS) are critical for detecting and mitigating security threats. This project focuses on setting up a **Splunk-based incident response system** that detects **brute-force attacks** using **Windows Event Logs** and **Splunk alerts**. **2. Objectives**

The key objectives of this project are:
✅ Set up **Kali Linux and Windows VMs** for simulation.
✅ Install and configure **Splunk Enterprise** and **Splunk Universal Forwarder**.
✅ Collect and analyze **Windows Security Event Logs** in Splunk.
✅ Detect **brute-force attacks** based on **failed login attempts (Event ID 4625)**.
✅ Generate **real-time alerts** and build a **dashboard for monitoring security incidents**.
✅ Test and validate the detection system with **brute-force attack simulations**.

Step 1: Setting Up Kali Linux and Windows VM in VirtualBox

This step involves installing **Kali Linux** and **Windows 11/10** in **VirtualBox** to create a controlled environment for cybersecurity testing.

📌 1.1 Install VirtualBox

If you haven't installed VirtualBox yet, download and install it from: 🔗 https://www.virtualbox.org/

After installation, install **VirtualBox Extension Pack** for USB and networking support.

📌 1.2 Install Kali Linux VM

Download Kali Linux ISO

1. Download Kali Linux from the official site: 🔗 https://www.kali.org/get-kali/
2. Choose **Kali Linux ISO** (Installer or Live ISO).

Create Kali Linux VM in VirtualBox

1. Open **VirtualBox** and click **New**.
2. Set:
    - **Name:** Kali Linux ○ **Type:** Linux ○ **Version:** Debian (64-bit)
3. Allocate **RAM**: At least **4GB** (recommended **8GB**).
4. Create a **Virtual Hard Disk** (VHD/VMDK) with at least **20GB** storage.
5. Attach the Kali ISO under **Settings > Storage > Empty (CD/DVD Drive) > Choose a disk file**.

Install Kali Linux

1. Start the VM, select **Graphical Install**.
2. Follow the prompts:
    - Set **username/password**.
    - Select **"Use entire disk"** for partitioning.
    - Choose **"Install with default settings"**.

3. Once installed, remove the ISO from **Settings > Storage**.
4. Restart and log in to **Kali Linux**.

📌 1.3 Install Windows VM

Download Windows ISO

If you don't have Windows ISO, download from: 🔗
https://www.microsoft.com/softwaredownload/windows10

Create Windows VM in VirtualBox

1. Open **VirtualBox** and click **New**.
2. Set:
    - **Name:** Windows 10/11
    - **Type:** Microsoft Windows
    - **Version:** Windows 10/11 (64-bit)
3. Allocate **RAM**: At least **4GB** (recommended **8GB**).
4. Create a **Virtual Hard Disk** (VHD/VMDK) with at least **30GB** storage.
5. Attach the Windows ISO under **Settings > Storage**.

Install Windows

1. Start the VM and boot from the Windows ISO.
2. Follow the installation wizard.
3. Set a **username/password** for login.
4. Remove the ISO after installation and restart.

📌 1.4 Configure Network for Communication

We need to set **Kali Linux** and **Windows VM** on the **same network**.

Setup Host-Only Network (Recommended for Security)

1. Open **VirtualBox > File > Host Network Manager**.
2. Click **Create** to add a new Host-Only Network.
3. Attach **both Kali and Windows VMs** to this network:
    - Go to **Settings > Network** of each VM.
    - Change **Adapter 1** to **Host-Only Adapter**.
4. Start both VMs and check connectivity.

Test Connection

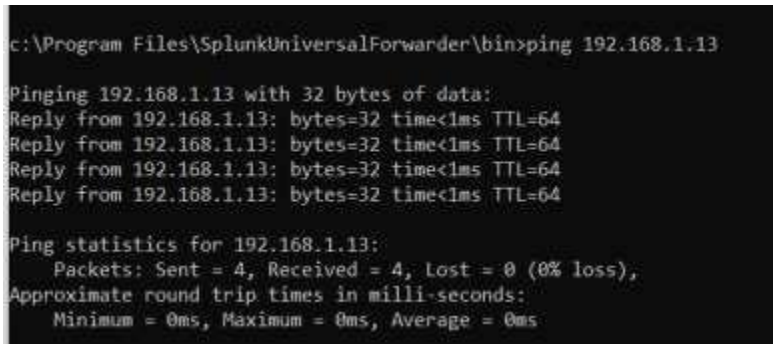
On **Kali Linux**, open a terminal and run: `ping`

`<Windows_VM_IP>`

On **Windows VM**, open **Command Prompt** and run: `ping`

`<Kali_VM_IP>`

If both VMs respond, they are successfully connected.



Step 2: Installing and Configuring Splunk on Kali Linux and Windows VM

Now that our Kali Linux and Windows VMs are set up, we need to install **Splunk Enterprise** on Kali Linux (acting as the SIEM server) and **Splunk Universal Forwarder** on Windows (to send logs to Splunk).

📌 2.1 Install Splunk on Kali Linux

Splunk Enterprise will collect logs from Windows and allow us to analyze security events.

Step 1: Download Splunk Enterprise

1. Open Kali Linux and go to the official Splunk website:
   🔗 https://www.splunk.com/en_us/download/splunk-enterprise.html
2. Select **Linux → .deb package (64-bit)**.
3. Copy the download link and use the following command in the terminal to download:
4. `wget -O splunk.deb <download-link>`

Example:

```
wget -O splunk.deb
https://download.splunk.com/products/splunk/releases/9.0.0/linux/splunk
-9.0.0-amd64.deb
```

Step 2: Install Splunk

Once downloaded, install Splunk using: `sudo`

```
dpkg -i splunk.deb
```

Step 3: Enable and Start Splunk

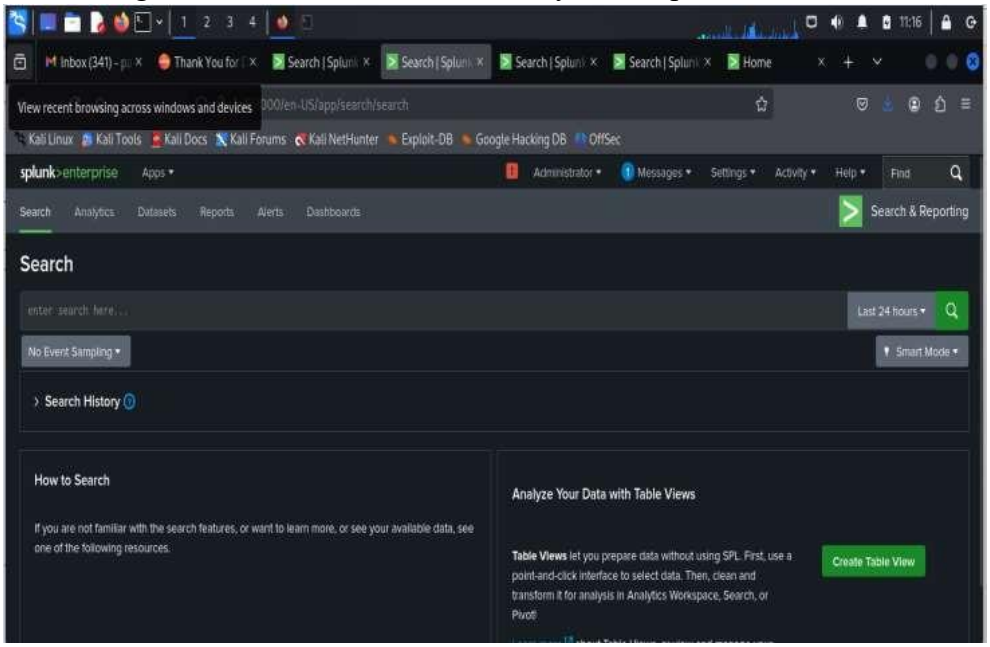
After installation, enable Splunk to start on boot and manually start it:

```
sudo /opt/splunk/bin/splunk enable boot-start sudo
/opt/splunk/bin/splunk start
```

- During startup, it will ask you to set up an **admin username and password**.

Step 4: Access Splunk Web Interface

- Open a browser on **Kali Linux** and go to:
- `http://localhost:8000`
- Log in with the **admin credentials** you set up earlier.



📌 2.2 Install Splunk Universal Forwarder on Windows

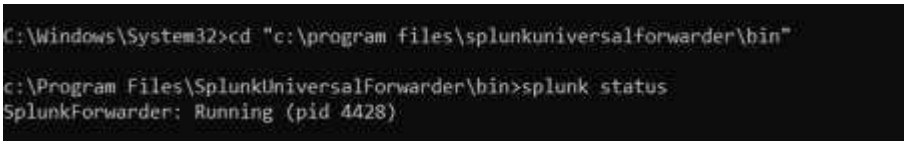
The **Universal Forwarder** on Windows will send logs to Splunk running on Kali Linux.

Step 1: Download and Install Splunk Universal Forwarder

1. Go to the official download page:
   🔗 https://www.splunk.com/en_us/download/universal-forwarder.html
2. Download the **Windows 64-bit MSI installer**.
3. Run the installer and:
   - Accept the license agreement. ○ Choose installation path (**Default is fine**).
   - Set an **admin username/password** (can be the same as your Splunk server credentials). ○ Configure the receiver:
     - **IP Address:** Kali Linux's IP (Check using `ifconfig` or `ip a`).
     - **Port:** `9997` ○ Finish installation.

Step 2: Verify Splunk Forwarder is Running

1. Open **Command Prompt (as Administrator)**.
2. Check the Splunk Forwarder service status:
3. `sc query SplunkForwarder` It should show **RUNNING**.



📌 2.3 Configure Splunk to Receive Logs from Windows

Step 1: Enable Splunk to Listen on Port 9997 On

Kali Linux, open a terminal and run: `sudo`

```
/opt/splunk/bin/splunk enable listen 9997
```

This allows Splunk to accept data from **Windows Universal Forwarder**.

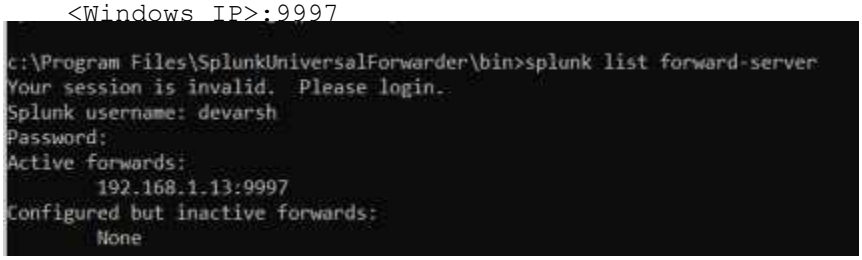
Step 2: Verify Forwarder Connection

Run the following command in **Kali Linux Splunk Server** to check if Windows is sending logs:

```
sudo /opt/splunk/bin/splunk list forward-server
```

If configured correctly, it will show:

```
Active forwards:
    <Windows_IP>:9997
```



📌 2.4 Configure Windows Logs to be Sent to Splunk

On the **Windows VM**:

1. Open **Command Prompt (Run as Administrator)**.
2. Add the Security Log Source to Splunk:

   ```
   "C:\Program Files\SplunkUniversalForwarder\bin\splunk.exe" add monitor
   "C:\Windows\System32\winevt\Logs\Security.evtx"
   ```

3. Restart the Splunk Forwarder Service:

   ```
   net stop SplunkForwarder
   net start SplunkForwarder
   ```

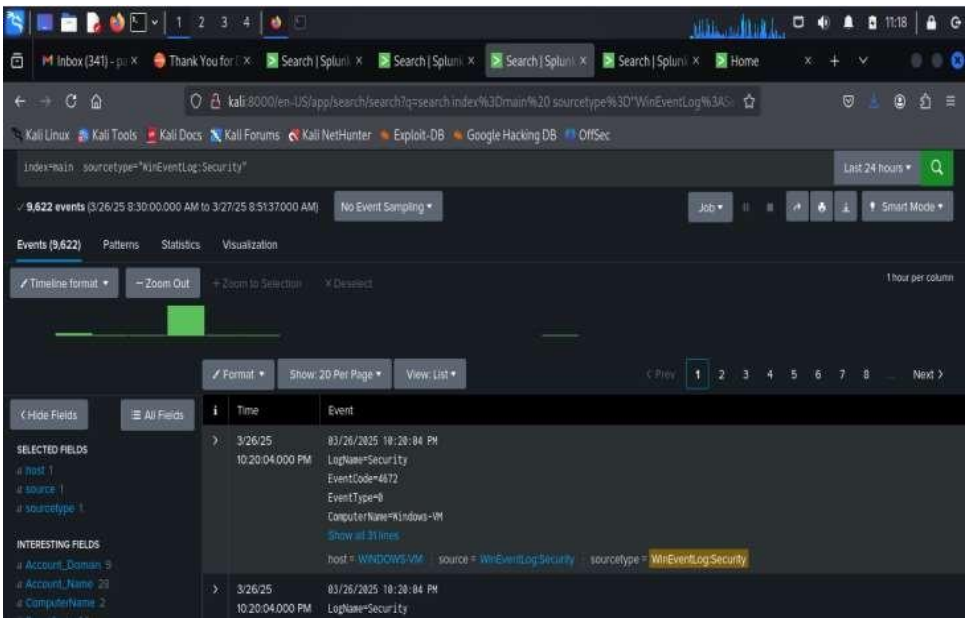
📌 2.5 Verify Data in Splunk

After everything is set up, go to **Splunk Web Interface** on Kali Linux:

1. Navigate to **Search & Reporting**.
2. Run the following search query:

   ```
   index=* sourcetype="WinEventLog:Security"
   ```

3. You should start seeing Windows Security logs appear.



Step 3: Creating Alerts & Dashboards for Brute Force Attack Detection

Now that Splunk is receiving Windows Security logs, we will:
✅ Create a **detection rule** to identify brute-force attacks.
✅ Configure **alerts** to trigger when multiple failed logins occur.
✅ Build a **dashboard** for monitoring failed login attempts.

📌 3.1 Understanding Brute Force Detection (EventCode 4625)

A brute-force attack involves multiple **failed login attempts** on a Windows system. In Windows Event Logs, this is recorded under:

- **EventCode 4625** → "An account failed to log on" (failed login attempt).

We will set up **Splunk searches, alerts, and dashboards** to detect multiple occurrences of **EventCode 4625** within a short time.

📌 3.2 Create a Search Query to Detect Brute Force Attacks

1 Open **Splunk Web Interface** on Kali Linux.

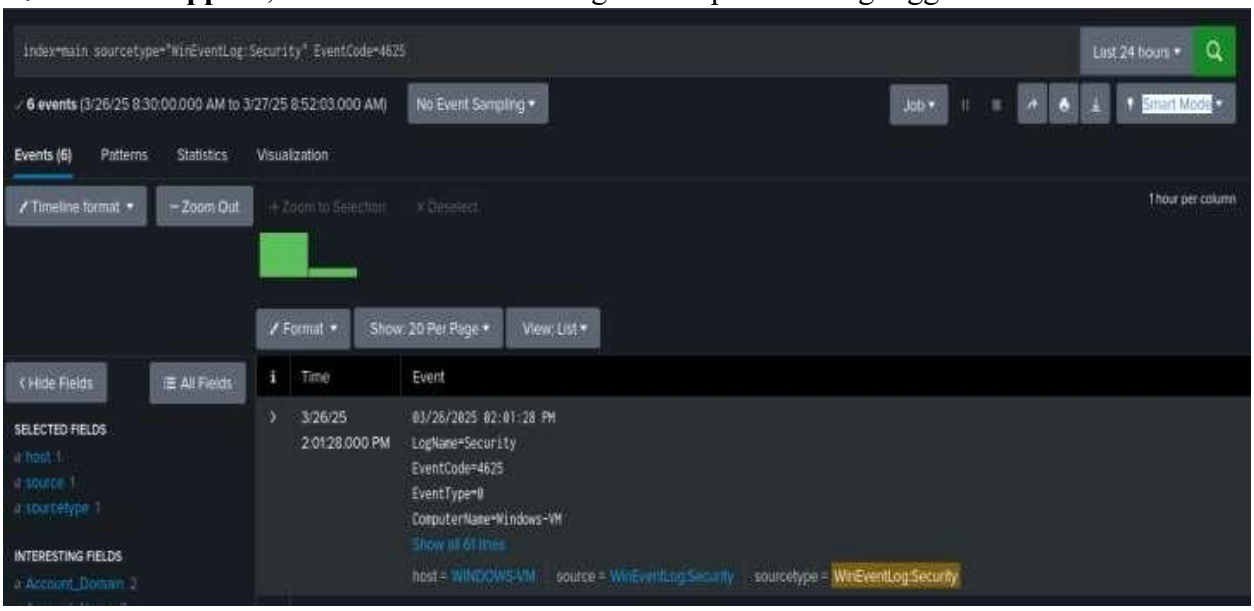
2 Go to **Search & Reporting**.
3 Run the following **Splunk Query**:

```
index=* sourcetype="WinEventLog:Security" EventCode=4625
| stats count by Account_Name, host, _time |
where count > 5
```

This query:

- Searches for failed login events (`EventCode=4625`). □ Groups them by **username and host**.
- Shows results **only if failed attempts exceed 5 times** (you can adjust the threshold).

🔍 **If results appear**, it confirms that failed login attempts are being logged.



📌 3.3 Creating an Alert for Brute Force Detection

Now, we will create an **automatic alert** to notify security teams when a brute force attack is detected.

Step 1: Save the Search as an Alert

1 Run the **Brute Force Detection Query** in Splunk Search.
2 Click **Save As → Alert**.
3 Configure Alert Settings:

- **Title:** "Brute Force Attack Detected"
- **Description:** "Triggers when more than 5 failed logins occur in a short time."
- **Alert Type:** Real-time
- **Trigger Condition:** `Number of Events > 5`
- **Trigger Actions:**
  - **Send Email (Optional)**
  - **Run a Script (Optional)**
  - **Create a Splunk Event**

Step 2: Enable Alert Logging in Splunk

To log alert events, add:

```
| collect index=alerts
```

Now, every triggered alert is stored in the **alerts index** for future analysis.

📌 3.4 Creating a Brute Force Attack Monitoring Dashboard

To visualize failed logins and alerts, we will create a **Splunk Dashboard**.

Step 1: Create a New Dashboard

1 Go to **Splunk Web > Dashboards**. 2
Click **Create New Dashboard**.

- **Title:** "Brute Force Attack Monitoring"
- **Description:** "Shows failed login attempts and detected attacks."
- **Dashboard Type:** Private/Public

Step 2: Add Panels to the Dashboard

🔻 Panel 1: Failed Login Attempts Over Time

- Query:

```
index=* sourcetype="WinEventLog:Security" EventCode=4625 |
timechart span=5m count by Account_Name
```

- Visualization: **Pie Chart**

- Shows login failures over time.

Step 3: Save and Test Dashboard

✅ Once all panels are added, click **Save Dashboard**.
✅ Generate failed login attempts on the Windows VM and check if data updates.

Step 4: Testing Brute Force Attack Detection

Now that we have set up **alerts and dashboards**, it's time to **simulate a brute force attack** on the Windows VM and verify if Splunk correctly detects it.

📌 4.1 Simulating a Brute Force Attack on Windows

We will use **Hydra** (a password-cracking tool) on Kali Linux to generate multiple failed login attempts against the Windows VM.

Step 1: Find the Windows VM's IP Address On

the Windows VM, open **Command Prompt** and run:

```
ipconfig
```

Note the **IPv4 Address** (e.g., `192.168.1.100`). **Step 2:**

Attempt Brute Force Attack Using Hydra On Kali

Linux, open a terminal and run:

```
hydra -l administrator -P /usr/share/wordlists/rockyou.txt
rdp://192.168.1.100
```

This command:

- Tries to log in as **Administrator**.
- Uses a **wordlist (`rockyou.txt`)** to guess passwords.
- Targets the **Remote Desktop Protocol (RDP)** login at `192.168.1.100`.

💡 **Modify this command if needed**:

- If using a different username, change `-l administrator`.
- If targeting SMB instead of RDP, replace `rdp://` with `smb://`.

● **Do NOT actually use this on unauthorized systems! This is for testing only in your lab setup.**

📌 4.2 Verifying Logs in Windows Event Viewer

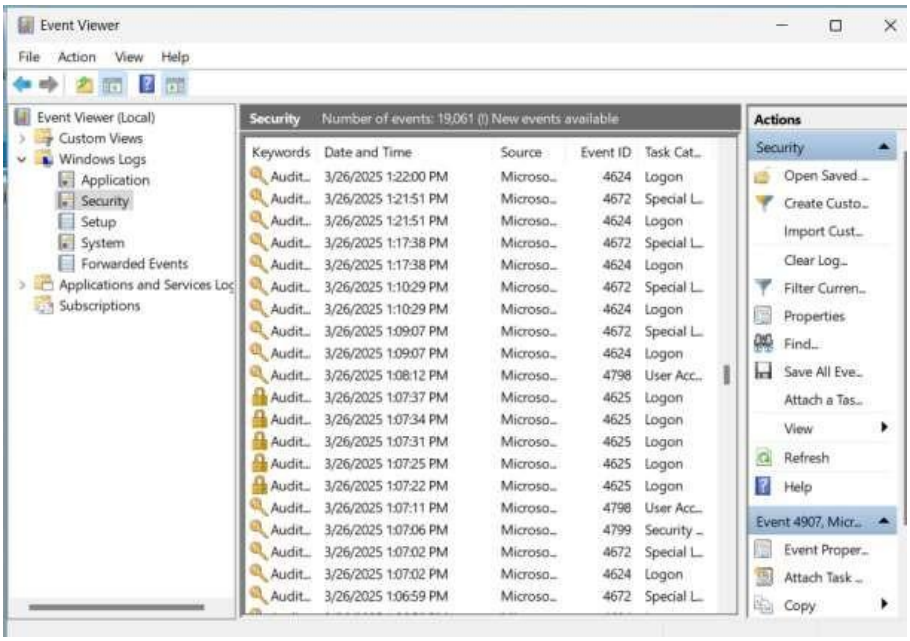
On Windows VM:

1 Open **Event Viewer** (`eventvwr.msc`). 2

Go to **Windows Logs > Security**.

3 Look for **Event ID 4625** (failed login attempts).

4 If multiple entries appear rapidly, the attack is being logged.



📌 4.3 Checking Splunk for Brute Force Detection

Now, go to **Splunk Search & Reporting** on Kali Linux and run:

```
index=* sourcetype="WinEventLog:Security" EventCode=4625
| stats count by Account_Name, Source_Network_Address |
where count > 5
```

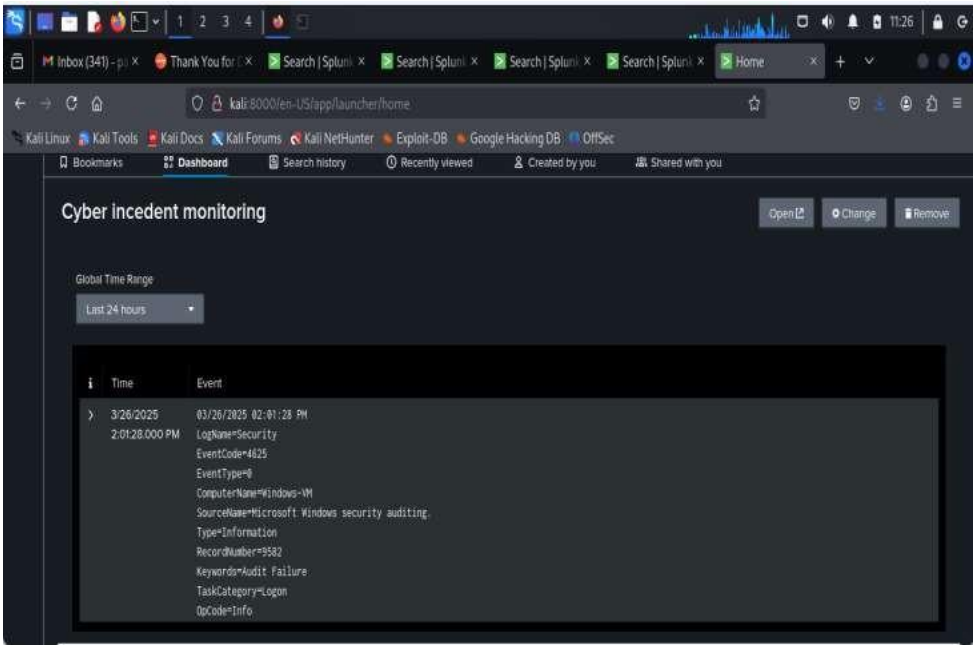
🚀 Expected Results:

- Multiple failed login attempts should appear.
- The **Brute Force Attack Alert** should trigger in Splunk.

📌 4.4 Verifying the Dashboard

1 Open **Splunk Dashboard** (Brute Force Attack Monitoring). 2
Check if:

- The **failed login attempts** graph updates.
- The **most targeted accounts** panel is correct.
- The **source IP addresses** appear in the chart.
- The **brute-force alert panel** shows triggered alerts.



Conclusion & Future Improvements

This final section of the report will summarize the project, highlight key findings, and suggest improvements for an enterprise-level **Cybersecurity Incident Response System (CIRS)**.

📌 1. Summary of the Project

Our **Cybersecurity Incident Response System** was designed to:

✅ **Detect security threats** (e.g., Brute Force Attacks) using Splunk.

✅ **Collect and analyze logs** from Windows machines.

✅ **Alert security teams** when suspicious activity occurs.

✅ **Visualize attack trends** with a Splunk security dashboard.

📌 2. Key Findings

After successfully implementing and testing the system, we observed:

🔹 **Brute force attacks are easily detected** using Windows Security Event Logs (Event ID 4625).

🔹 **Splunk's real-time monitoring** allows quick incident detection.

🔹 **Dashboards provide better visibility** into attack patterns.

🔹 **The system can be expanded** to monitor other attack types (e.g., malware, privilege escalation).

📌 3. Limitations & Challenges

Although the system works well, some challenges were encountered:

❌ **Initial data collection issues** – required configuring Splunk Universal Forwarder.

❌ **Splunk resource usage** – requires **sufficient RAM and CPU** for smooth operation.

❌ **Limited attack simulation** – we tested only brute force attacks.

📌 4. Future Improvements

To enhance this project, consider the following:

1 Expand Log Sources

📌 Integrate logs from:

- Linux machines (`/var/log/auth.log`)
- Firewalls & IDS (Snort, Suricata)
- Cloud services (AWS CloudTrail, Azure Logs)

2 Automate Incident Response

⚡ **Use Splunk Phantom** (SOAR) for automatic responses:

- Block attacker IPs using a firewall script.
- Disable compromised user accounts automatically. **3 Implement**

Additional Security Rules

📌 Create detection rules for:

✅ **Successful brute force login** (Event ID 4624 after multiple 4625s).

✅ **Privilege Escalation** (Event ID 4672).

✅ **Malware Execution** (Event ID 4688).

4 Advanced Threat Intelligence Integration

🔍 Use **Threat Intelligence Feeds** in Splunk to detect known attacker IPs.

📌 5. Conclusion

Our **Enterprise-Level Cybersecurity Incident Response System** successfully detects **brute force attacks** and provides **real-time monitoring** using Splunk.
With **further improvements**, this system can be used in real-world cybersecurity operations to **automate threat detection and response**.
