Author: P. Rajeev Date: 28-10-2025

Task 01 – Log Analysis, SIEM Dashboard Creation

1.Objective

This report details a task focused on log analysis, security event documentation, and monitoring dashboard setup.

The primary objectives are:

- Understand the fundamentals and operations of a Security Operations Center (SOC).
- Learn key SOC processes such as monitoring, detection, and incident response.
- Perform hands-on log analysis and alert configuration.
- Build a mini SOC lab using tools like Elastic SIEM, Wazuh, and Snort.
- Develop skills to detect, analyze, and document security events effectively.

2.Introduction

This task focuses on understanding the core concepts of SOC operations, including security monitoring, log management, and incident response workflows. Through both theoretical learning and practical exercises, learners will gain experience in using tools like **Elastic SIEM, Wazuh, and Snort** to analyze logs, configure alerts, and simulate real-world attack detection within a mini SOC environment.

3.Target and Attacker Description

- Attacker: Host A (Kali Linux Machine)
- Target: Host B (Windows)

4.Tools and Setup

- Windows Event Viewer
- Sqlite3
- ELK Stack
- Wazuh Dashboard

5.Log Analysis

Log analysis is the process of collecting, reviewing, and interpreting system, network, and application logs to identify patterns, detect anomalies, and investigate security incidents.Logs are digital records automatically generated by devices, operating systems, applications, or security tools that document events and activities.

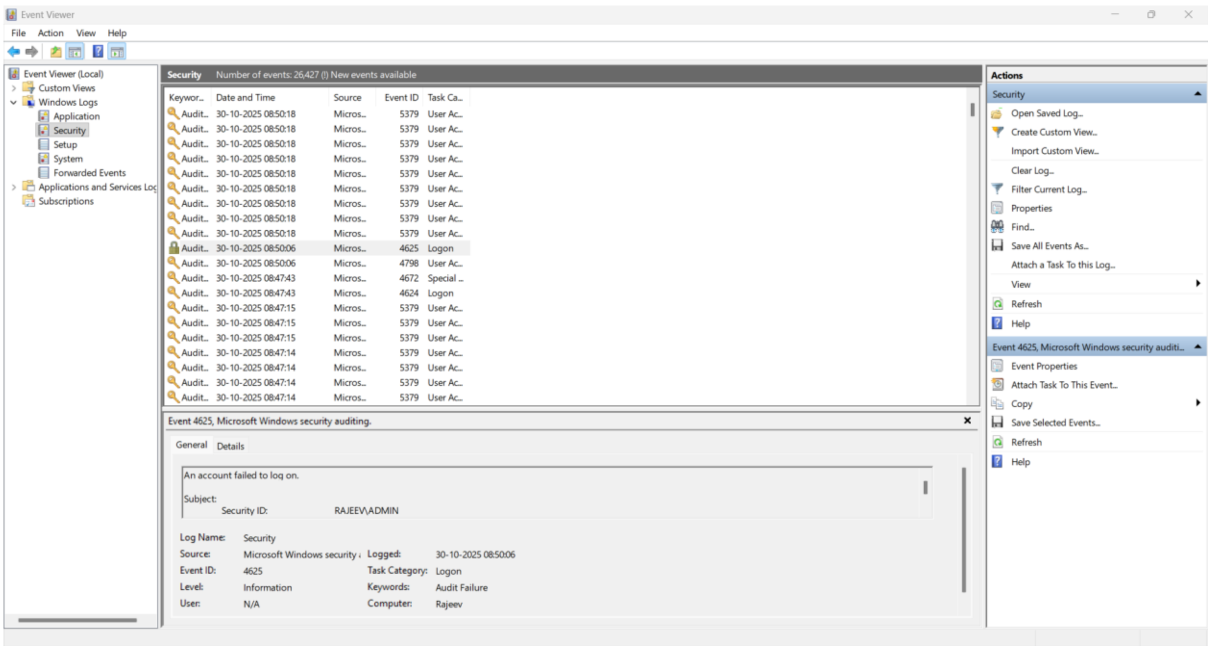
5.1 Windows Event Viewer

Windows Event Viewer is a built-in tool that lets administrators and users view and analyze event logs generated by the Windows operating system and its applications.

It records all significant actions and changes that happen on the computer — such as:

- ✓ User logins and logouts
- ✓ System errors and warnings
- ✓ Application crashes
- ✓ Security events (e.g., failed logins, privilege use)

These logs help in **system troubleshooting**, **performance monitoring**, and **security auditing**.



Security Number of events: 26,427 (!) New events available

Keywor...	Date and Time	Source	Event ID	Task Ca...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:18	Micros...	5379	User Ac...
Audit...	30-10-2025 08:50:06	Micros...	4625	Logon
Audit...	30-10-2025 08:50:06	Micros...	4798	User Ac...
Audit...	30-10-2025 08:47:43	Micros...	4672	Special ...
Audit...	30-10-2025 08:47:43	Micros...	4624	Logon
Audit...	30-10-2025 08:47:15	Micros...	5379	User Ac...
Audit...	30-10-2025 08:47:15	Micros...	5379	User Ac...
Audit...	30-10-2025 08:47:15	Micros...	5379	User Ac...
Audit...	30-10-2025 08:47:14	Micros...	5379	User Ac...
Audit...	30-10-2025 08:47:14	Micros...	5379	User Ac...
Audit...	30-10-2025 08:47:14	Micros...	5379	User Ac...
Audit...	30-10-2025 08:47:14	Micros...	5379	User Ac...

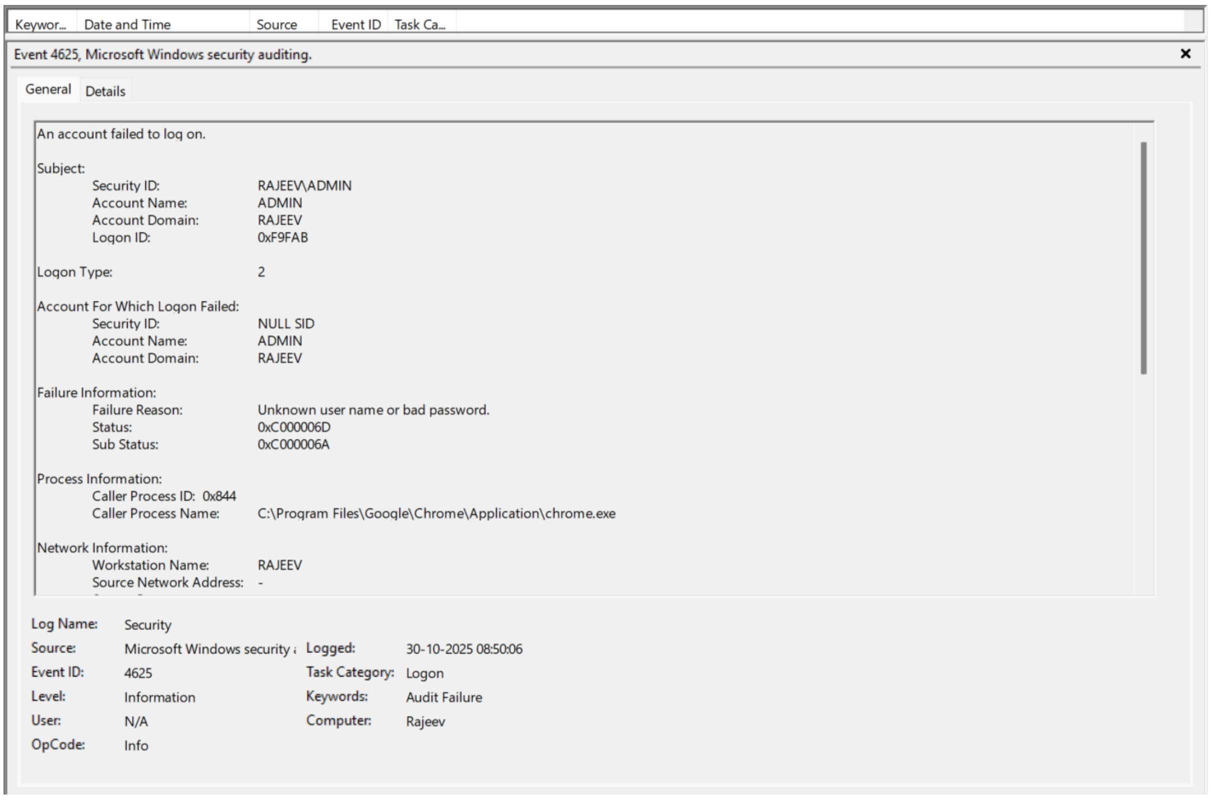
Event 4625, Microsoft Windows security auditing.

General Details

```
An account failed to log on.

Subject:
        Security ID:            RAJEEV\ADMIN

Log Name:    Security
Source:      Microsoft Windows security a   Logged:         30-10-2025 08:50:06
Event ID:    4625                           Task Category:  Logon
Level:       Information                    Keywords:       Audit Failure
User:        N/A                            Computer:       Rajeev
```

5.1.1 Brute-Force Attack Identification

A **brute-force attack** is when an attacker systematically tries many possible credentials (passwords, PINs, API keys, session tokens) until one works. It's noisy and common — attackers automate it with bots, password lists, or credential stuffing (using leaked username/password pairs from other breaches).



```
Keywor...   Date and Time   Source   Event ID   Task Ca...
Event 4625, Microsoft Windows security auditing.
General   Details

An account failed to log on.

Subject:
        Security ID:            RAJEEV\ADMIN
        Account Name:           ADMIN
        Account Domain:         RAJEEV
        Logon ID:               0xF9FAB

Logon Type:                     2

Account For Which Logon Failed:
        Security ID:            NULL SID
        Account Name:           ADMIN
        Account Domain:         RAJEEV

Failure Information:
        Failure Reason:         Unknown user name or bad password.
        Status:                 0xC000006D
        Sub Status:             0xC000006A

Process Information:
        Caller Process ID:  0x844
        Caller Process Name:    C:\Program Files\Google\Chrome\Application\chrome.exe

Network Information:
        Workstation Name:       RAJEEV
        Source Network Address: -

Log Name:   Security
Source:     Microsoft Windows security a   Logged:          30-10-2025 08:50:06
Event ID:   4625                           Task Category:   Logon
Level:      Information                    Keywords:        Audit Failure
User:       N/A                            Computer:        Rajeev
OpCode:     Info
```

5.1.2 Creating a Template with fields

Date/Time	Source IP	Event ID	Description	Action Taken
29-10-2025 17:29:48	192.168.1.10	4625	Failed logon attempt detected — possible incorrect credentials or brute-force attempt.	Investigated login failure; verified user activity and account status.

30-10-2025 08:50:06	192.168.1.10	4625	Repeated failed logon detected; indicates ongoing unauthorized login attempts.	Added IP to temporary block list and enforced password reset policy.

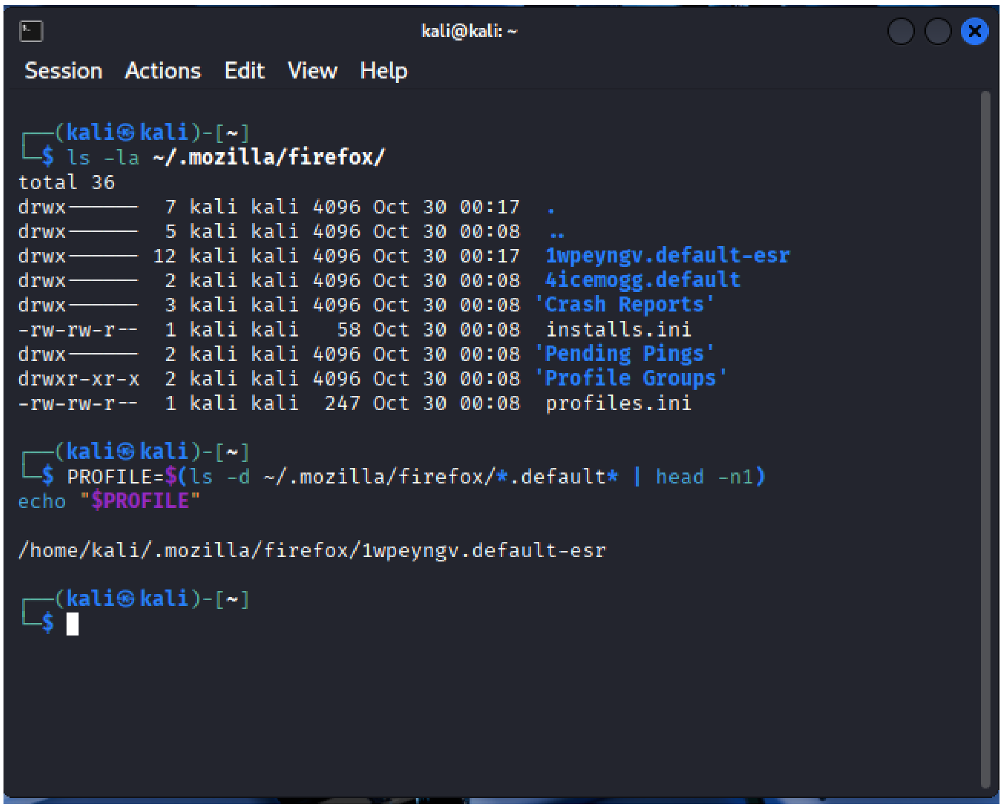
5.2 Browser History Analysis

Browser history analysis is the process of examining data stored by web browsers to understand a user's online activities — such as visited websites, downloads, searches, cookies, and sessions.

It is a key part of **digital forensics**, **incident response**, and **threat investigation**, as browsers often contain crucial evidence of user behavior or compromise.

5.2.1 Find Profile Path

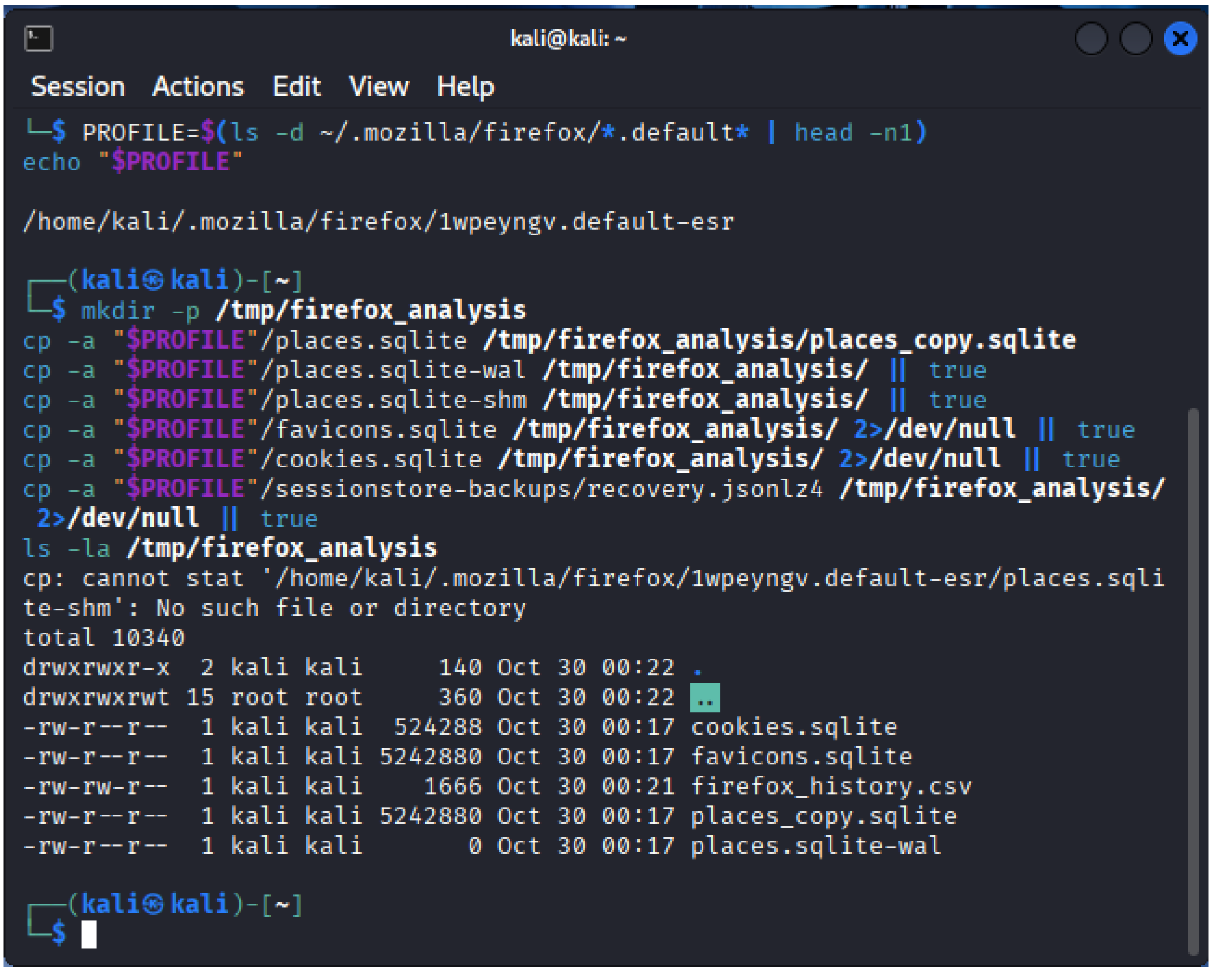
First find the Firefox profile path and select the profile folder using the commands



```
┌──(kali㉿kali)-[~]
└─$ ls -la ~/.mozilla/firefox/
total 36
drwx------  7 kali kali 4096 Oct 30 00:17  .
drwx------  5 kali kali 4096 Oct 30 00:08  ..
drwx------ 12 kali kali 4096 Oct 30 00:17  1wpeyngv.default-esr
drwx------  2 kali kali 4096 Oct 30 00:08  4icemogg.default
drwx------  3 kali kali 4096 Oct 30 00:08 'Crash Reports'
-rw-rw-r--  1 kali kali   58 Oct 30 00:08  installs.ini
drwx------  2 kali kali 4096 Oct 30 00:08 'Pending Pings'
drwxr-xr-x  2 kali kali 4096 Oct 30 00:08 'Profile Groups'
-rw-rw-r--  1 kali kali  247 Oct 30 00:08  profiles.ini

┌──(kali㉿kali)-[~]
└─$ PROFILE=$(ls -d ~/.mozilla/firefox/*.default* | head -n1)
echo "$PROFILE"

/home/kali/.mozilla/firefox/1wpeyngv.default-esr

┌──(kali㉿kali)-[~]
└─$ 
```

5.2.2 Firefox Data Extraction and Analysis Setup

Now creates a temporary folder (/tmp/firefox_analysis) and copies key Firefox data files from your profile—like history, bookmarks, cookies, and session data—into it for analysis. The cp -a option preserves file details, || true prevents errors if files are missing, and the final ls -la lists all copied files.



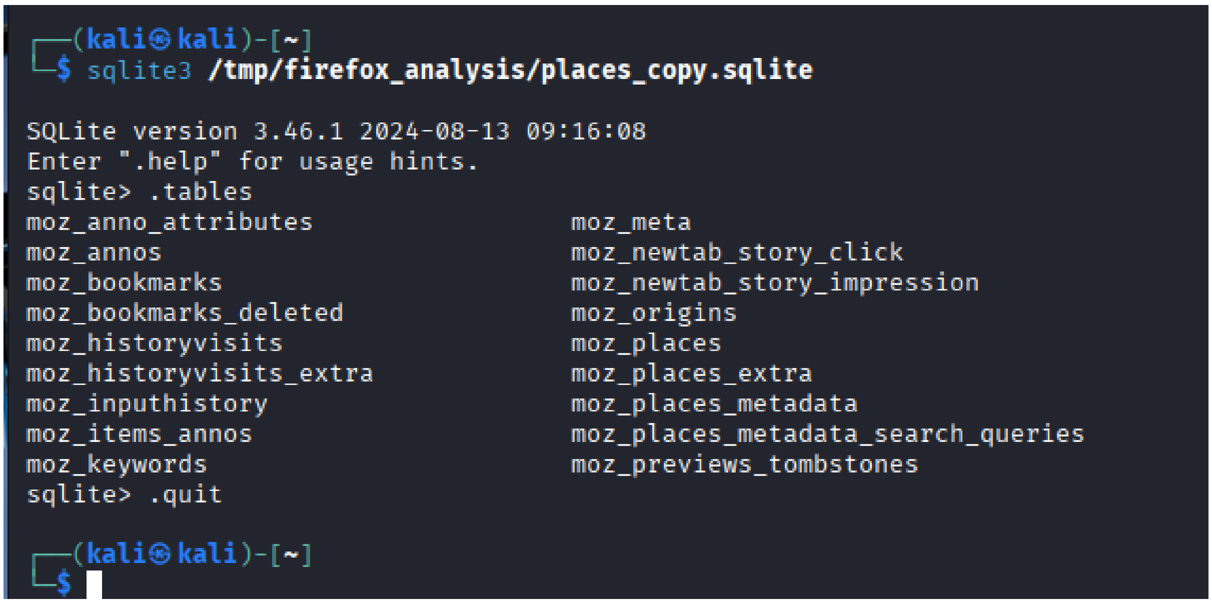
```
└─$ PROFILE=$(ls -d ~/.mozilla/firefox/*.default* | head -n1)
echo "$PROFILE"

/home/kali/.mozilla/firefox/1wpeyngv.default-esr

┌──(kali㉿kali)-[~]
└─$ mkdir -p /tmp/firefox_analysis
cp -a "$PROFILE"/places.sqlite /tmp/firefox_analysis/places_copy.sqlite
cp -a "$PROFILE"/places.sqlite-wal /tmp/firefox_analysis/ || true
cp -a "$PROFILE"/places.sqlite-shm /tmp/firefox_analysis/ || true
cp -a "$PROFILE"/favicons.sqlite /tmp/firefox_analysis/ 2>/dev/null || true
cp -a "$PROFILE"/cookies.sqlite /tmp/firefox_analysis/ 2>/dev/null || true
cp -a "$PROFILE"/sessionstore-backups/recovery.jsonlz4 /tmp/firefox_analysis/
 2>/dev/null || true
ls -la /tmp/firefox_analysis
cp: cannot stat '/home/kali/.mozilla/firefox/1wpeyngv.default-esr/places.sqli
te-shm': No such file or directory
total 10340
drwxrwxr-x  2 kali kali     140 Oct 30 00:22 .
drwxrwxrwt 15 root root     360 Oct 30 00:22 ..
-rw-r--r--  1 kali kali  524288 Oct 30 00:17 cookies.sqlite
-rw-r--r--  1 kali kali 5242880 Oct 30 00:17 favicons.sqlite
-rw-rw-r--  1 kali kali    1666 Oct 30 00:21 firefox_history.csv
-rw-r--r--  1 kali kali 5242880 Oct 30 00:17 places_copy.sqlite
-rw-r--r--  1 kali kali       0 Oct 30 00:17 places.sqlite-wal

┌──(kali㉿kali)-[~]
└─$ 
```

Notes:

- places.sqlite contains history & bookmarks.
- favicons.sqlite stores site icons.
- cookies.sqlite stores cookies.
- recovery.jsonlz4 contains the current session (open tabs/windows) — needs decompression.

5.2.3 Quick Database Check Using SQLite3

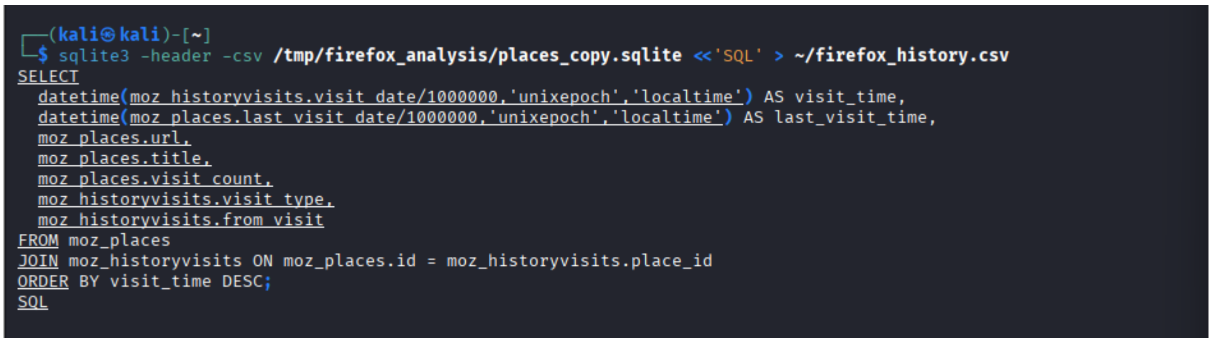
Now opens the copied Firefox database (places_copy.sqlite) with SQLite3 to quickly inspect its contents. Using .tables lists all tables, and .quit exits. You can also run SQL queries directly from the terminal using a heredoc for faster, one-off analysis.



```
┌──(kali㉿kali)-[~]
└─$ sqlite3 /tmp/firefox_analysis/places_copy.sqlite

SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .tables
moz_anno_attributes              moz_meta
moz_annos                        moz_newtab_story_click
moz_bookmarks                    moz_newtab_story_impression
moz_bookmarks_deleted            moz_origins
moz_historyvisits                moz_places
moz_historyvisits_extra          moz_places_extra
moz_inputhistory                 moz_places_metadata
moz_items_annos                  moz_places_metadata_search_queries
moz_keywords                     moz_previews_tombstones
sqlite> .quit

┌──(kali㉿kali)-[~]
└─$ 
```

5.2.4 Exporting Firefox Browsing History to CSV

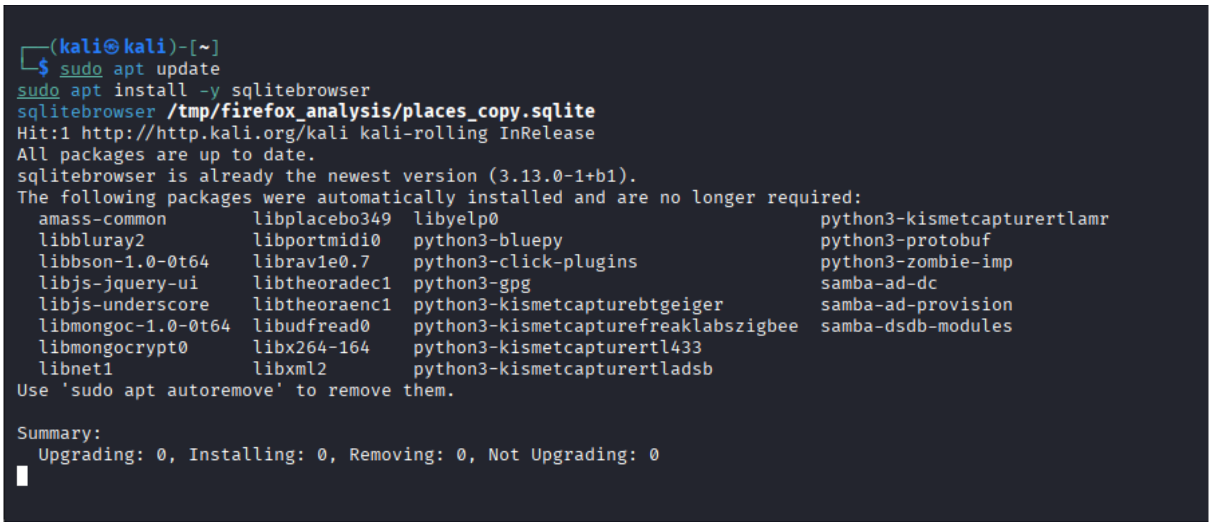
Now runs an SQL query on the copied Firefox database to extract browsing history details—like URLs, titles, visit times, and counts—and saves them in CSV format (firefox_history.csv). It joins relevant tables, converts timestamps to local time, and outputs the results in a readable format for easy analysis



```
┌──(kali㉿kali)-[~]
└─$ sqlite3 -header -csv /tmp/firefox_analysis/places_copy.sqlite <<'SQL' > ~/firefox_history.csv
SELECT
  datetime(moz_historyvisits.visit_date/1000000,'unixepoch','localtime') AS visit_time,
  datetime(moz_places.last_visit_date/1000000,'unixepoch','localtime') AS last_visit_time,
  moz_places.url,
  moz_places.title,
  moz_places.visit_count,
  moz_historyvisits.visit_type,
  moz_historyvisits.from_visit
FROM moz_places
JOIN moz_historyvisits ON moz_places.id = moz_historyvisits.place_id
ORDER BY visit_time DESC;
SQL
```

5.2.5 Browsing History

Now step lets you explore the Firefox database visually without typing SQL commands. First, install the graphical tool using sudo apt update and sudo apt install -y sqlitebrowser, then open the copied database with sqlitebrowser /tmp/firefox_analysis/places_copy.sqlite. In the application, go to the **"Browse Data"** tab and select the **moz_places** or **moz_historyvisits** table to view the records. You can easily click on columns, apply filters, and sort results — providing a simple and user-friendly way to examine browsing data.



```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
sudo apt install -y sqlitebrowser
sqlitebrowser /tmp/firefox_analysis/places_copy.sqlite
Hit:1 http://http.kali.org/kali kali-rolling InRelease
All packages are up to date.
sqlitebrowser is already the newest version (3.13.0-1+b1).
The following packages were automatically installed and are no longer required:
  amass-common        libplacebo349  libyelp0                          python3-kismetcapturertlamr
  libbluray2          libportmidi0   python3-bluepy                    python3-protobuf
  libbson-1.0-0t64    libravle0.7    python3-click-plugins             python3-zombie-imp
  libjs-jquery-ui     libtheoradec1  python3-gpg                       samba-ad-dc
  libjs-underscore    libtheoraenc1  python3-kismetcapturebtgeiger     samba-ad-provision
  libmongoc-1.0-0t64  libudfread0    python3-kismetcapturefreaklabszigbee  samba-dsdb-modules
  libmongocrypt0      libx264-164    python3-kismetcapturertl433
  libnet1             libxml2        python3-kismetcapturertladsb
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
█
```

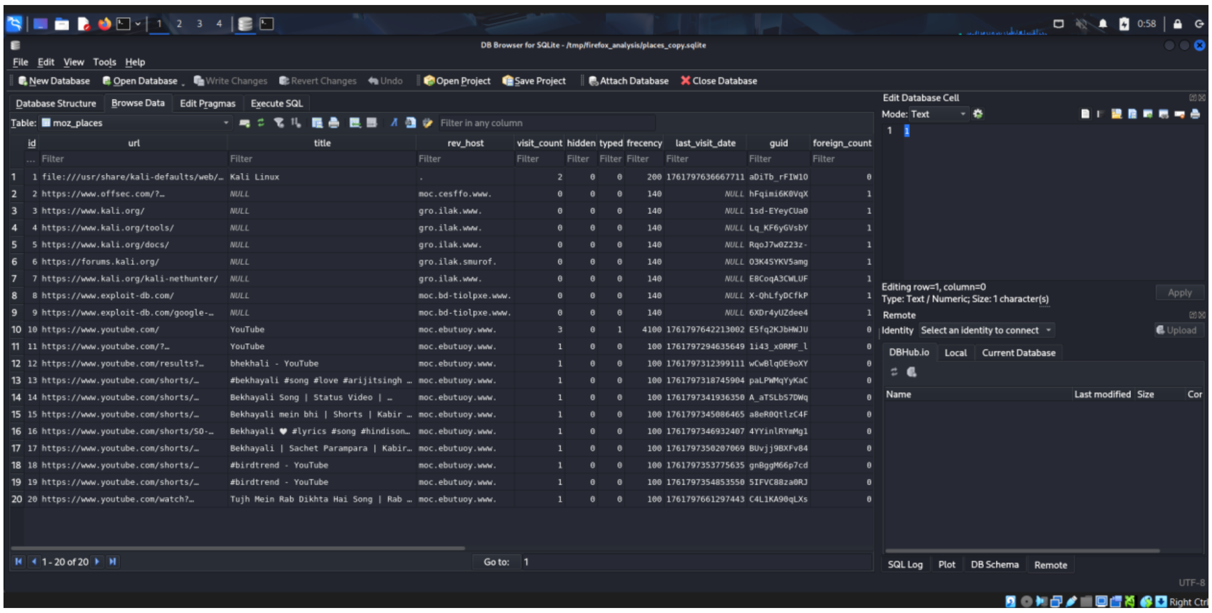


Table: moz_places

	id	url	title	rev_host	visit_count	hidden	typed	frecency	last_visit_date	guid	foreign_count
1	1	file:///usr/share/kali-defaults/web/…	Kali Linux	.	2	0	0	200	1761797636667711	aDiTb_rFIW1O	0
2	2	https://www.offsec.com/?…	NULL	moc.cesffo.www.	0	0	0	140	NULL	hFqimi6K0VqX	1
3	3	https://www.kali.org/	NULL	gro.ilak.www.	0	0	0	140	NULL	1sd-EYeyCUa0	1
4	4	https://www.kali.org/tools/	NULL	gro.ilak.www.	0	0	0	140	NULL	Lq_KF6yGVsbY	1
5	5	https://www.kali.org/docs/	NULL	gro.ilak.www.	0	0	0	140	NULL	RqoJ7w0Z23z-	1
6	6	https://forums.kali.org/	NULL	gro.ilak.smurof.	0	0	0	140	NULL	O3K4SYKV5amg	1
7	7	https://www.kali.org/kali-nethunter/	NULL	gro.ilak.www.	0	0	0	140	NULL	E8CoqA3CWLUF	1
8	8	https://www.exploit-db.com/	NULL	moc.bd-tiolpxe.www.	0	0	0	140	NULL	X-QhLfyDCfkP	1
9	9	https://www.exploit-db.com/google-…	NULL	moc.bd-tiolpxe.www.	0	0	0	140	NULL	6XDr4yUZdee4	1
10	10	https://www.youtube.com/	YouTube	moc.ebutuoy.www.	3	0	1	4100	1761797642213002	E5fq2KJbHWJU	0
11	11	https://www.youtube.com/?…	YouTube	moc.ebutuoy.www.	1	0	0	100	1761797294635649	1i43_x0RMF_l	0
12	12	https://www.youtube.com/results?…	bhekhali - YouTube	moc.ebutuoy.www.	1	0	0	100	1761797312399111	wCwBlqOE9oXY	0
13	13	https://www.youtube.com/shorts/…	#bekhayali #song #love #arijitsingh …	moc.ebutuoy.www.	1	0	0	100	1761797318745904	paLPWMqYyKaC	0
14	14	https://www.youtube.com/shorts/…	Bekhayali Song | Status Video | …	moc.ebutuoy.www.	1	0	0	100	1761797341936350	A_aTSLbS7DWq	0
15	15	https://www.youtube.com/shorts/…	Bekhayali mein bhi | Shorts | Kabir …	moc.ebutuoy.www.	1	0	0	100	1761797345086465	a8eR0QtlzC4F	0
16	16	https://www.youtube.com/shorts/SO-…	Bekhayali ♥ #lyrics #song #hindison…	moc.ebutuoy.www.	1	0	0	100	1761797346932407	4YYinlRYmMg1	0
17	17	https://www.youtube.com/shorts/…	Bekhayali | Sachet Parampara | Kabir…	moc.ebutuoy.www.	1	0	0	100	1761797350207069	BUvjj9BXFv84	0
18	18	https://www.youtube.com/shorts/…	#birdtrend - YouTube	moc.ebutuoy.www.	1	0	0	100	1761797353775635	gnBggM66p7cd	0
19	19	https://www.youtube.com/shorts/…	#birdtrend - YouTube	moc.ebutuoy.www.	1	0	0	100	1761797354853550	5IFVC88za0RJ	0
20	20	https://www.youtube.com/watch?…	Tujh Mein Rab Dikhta Hai Song | Rab …	moc.ebutuoy.www.	1	0	0	100	1761797661297443	C4L1KA90qLXs	0

1 - 20 of 20   Go to: 1

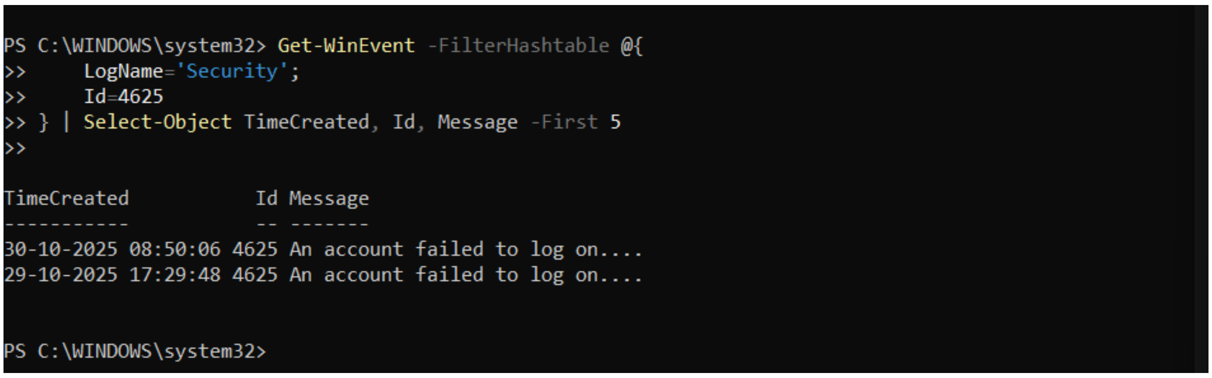
Mode: Text
Editing row=1, column=0
Type: Text / Numeric; Size: 1 character(s)
Identity: Select an identity to connect

6.1 Mock Event

A **mock event** is a **simulated security incident** created for practice or testing purposes. It helps analysts or students learn how to detect, analyze, and respond to real-world threats using security tools.

6.1.1 Check for Login Failure Events in Windows Logs

Right-click on the Start menu → **Windows PowerShell (Admin)**.

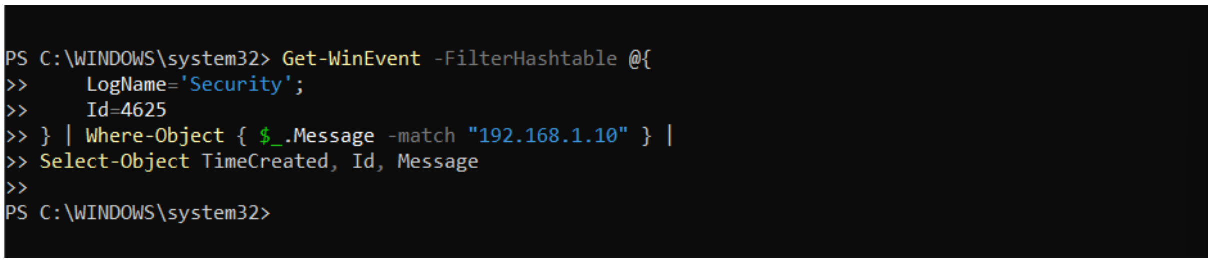


```
PS C:\WINDOWS\system32> Get-WinEvent -FilterHashtable @{
>>     LogName='Security';
>>     Id=4625
>> } | Select-Object TimeCreated, Id, Message -First 5
>>

TimeCreated           Id Message
-----------           -- -------
30-10-2025 08:50:06 4625 An account failed to log on....
29-10-2025 17:29:48 4625 An account failed to log on....


PS C:\WINDOWS\system32>
```

This shows the latest **failed login attempts** (Event ID 4625).

6.1.2 Filter Events by Source IP (Simulating "192.168.1.10")

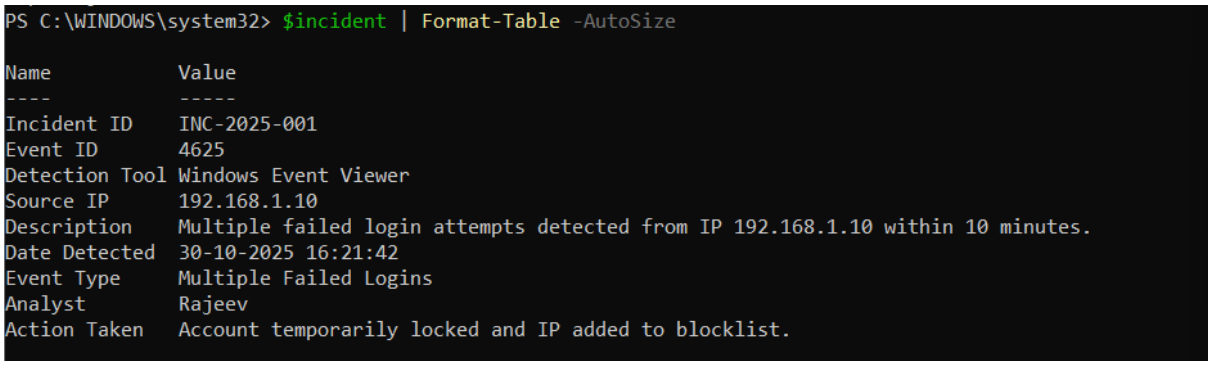
The command is used to **filter Windows Security log events** that match a specific **source IP address (192.168.1.10)**, simulating a scenario such as multiple failed login attempts from the same host. It uses the Get-WinEvent cmdlet to retrieve events from the **Security log** with the **Event ID 4625**, which corresponds to **failed logon attempts**. The Where-Object condition then filters these events by checking if the message contains the IP address "192.168.1.10". Finally, the output displays only the **time of occurrence**, **event ID**, and **message details**, making it easier to analyze and document suspicious activities associated with that IP. This technique is commonly used in mock event analysis or SOC practice to simulate and detect brute-force or unauthorized access attempts.



```
PS C:\WINDOWS\system32> Get-WinEvent -FilterHashtable @{
>>     LogName='Security';
>>     Id=4625
>> } | Where-Object { $_.Message -match "192.168.1.10" } |
>> Select-Object TimeCreated, Id, Message
>>
PS C:\WINDOWS\system32>
```

6.1.3 Create a Table View in PowerShell

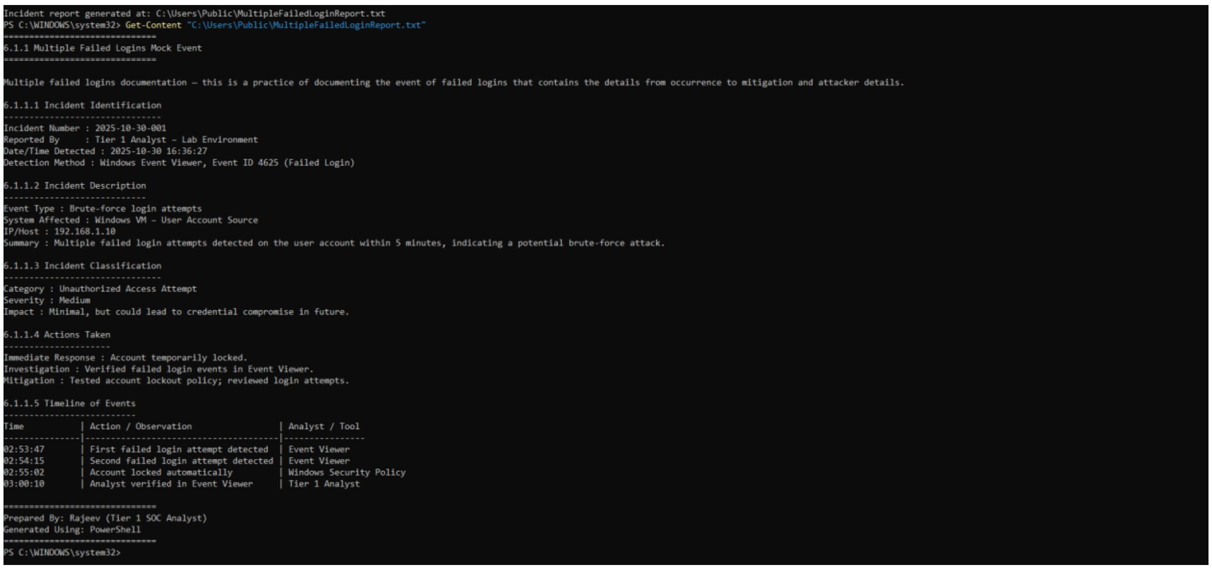
To **display the contents of the $incident variable** in a well-formatted table. The Format-Table cmdlet organizes the data into **neatly aligned columns**, and the -AutoSize parameter automatically adjusts the column widths based on the content to ensure everything fits properly. This makes the output **more readable and professional**, especially when reviewing or documenting mock security incidents, event logs, or analysis results. It's commonly used in SOC reporting scripts to present filtered event data in a clear, structured format.



```
PS C:\WINDOWS\system32> $incident | Format-Table -AutoSize

Name           Value
----           -----
Incident ID    INC-2025-001
Event ID       4625
Detection Tool Windows Event Viewer
Source IP      192.168.1.10
Description    Multiple failed login attempts detected from IP 192.168.1.10 within 10 minutes.
Date Detected  30-10-2025 16:21:42
Event Type     Multiple Failed Logins
Analyst        Rajeev
Action Taken   Account temporarily locked and IP added to blocklist.
```

6.1.4 Multiple Failed Login Mock Events

Multiple Failed Login Mock Events simulate repeated unsuccessful login attempts from a specific IP (like 192.168.1.10) to test detection of brute-force attacks. They help SOC analysts practice identifying, analyzing, and responding to suspicious login activities in a controlled environment.



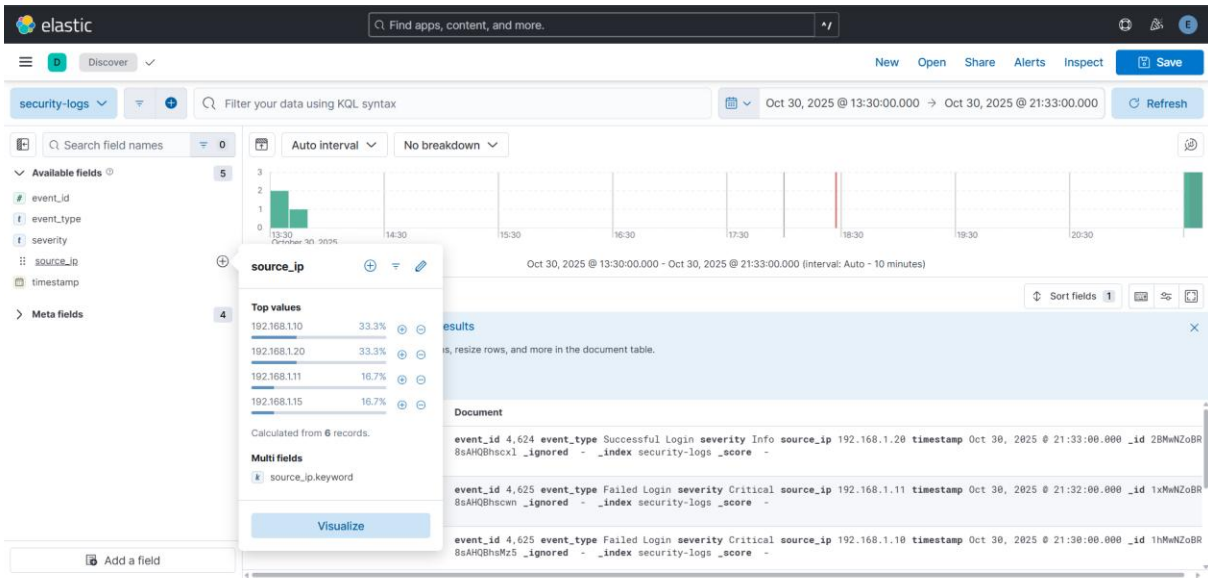
```
Incident report generated at: C:\Users\Public\MultipleFailedLoginReport.txt
PS C:\WINDOWS\system32> Get-Content "C:\Users\Public\MultipleFailedLoginReport.txt"
==============================
6.1.1 Multiple Failed Logins Mock Event
==============================

Multiple failed logins documentation – this is a practice of documenting the event of failed logins that contains the details from occurrence to mitigation and attacker details.

6.1.1.1 Incident Identification
-------------------------------
Incident Number : 2025-10-30-001
Reported By     : Tier 1 Analyst – Lab Environment
Date/Time Detected : 2025-10-30 16:36:27
Detection Method : Windows Event Viewer, Event ID 4625 (Failed Login)

6.1.1.2 Incident Description
----------------------------
Event Type : Brute-force login attempts
System Affected : Windows VM – User Account Source
IP/Host : 192.168.1.10
Summary : Multiple failed login attempts detected on the user account within 5 minutes, indicating a potential brute-force attack.

6.1.1.3 Incident Classification
-------------------------------
Category : Unauthorized Access Attempt
Severity : Medium
Impact : Minimal, but could lead to credential compromise in future.

6.1.1.4 Actions Taken
---------------------
Immediate Response : Account temporarily locked.
Investigation : Verified failed login events in Event Viewer.
Mitigation : Tested account lockout policy; reviewed login attempts.

6.1.1.5 Timeline of Events
--------------------------
Time           | Action / Observation                | Analyst / Tool
---------------|-------------------------------------|---------------
02:53:47       | First failed login attempt detected  | Event Viewer
02:54:15       | Second failed login attempt detected | Event Viewer
02:55:02       | Account locked automatically         | Windows Security Policy
03:00:10       | Analyst verified in Event Viewer     | Tier 1 Analyst

==============================
Prepared By: Rajeev (Tier 1 SOC Analyst)
Generated Using: PowerShell
==============================
PS C:\WINDOWS\system32>
```

7. Monitoring Dashboard

This setup demonstrates how to monitor and visualize security events using **Elasticsearch** and **Kibana** on **Ubuntu (via WSL2)**.
It helps SOC analysts track suspicious activities by creating dashboards that show:

- **Top 10 Source IPs generating alerts**
- **Frequency of Critical Event IDs**

By using **Kibana visualizations** and optional **Sigma detection rules**, you can analyze and detect security incidents in real time.



Oct 30, 2025 @ 13:30:00.000 → Oct 30, 2025 @ 21:33:00.000

Oct 30, 2025 @ 13:30:00.000 - Oct 30, 2025 @ 21:33:00.000 (interval: Auto - 10 minutes)

Top values	
192.168.1.10	33.3%
192.168.1.20	33.3%
192.168.1.11	16.7%
192.168.1.15	16.7%

Calculated from 6 records.

esults

s, resize rows, and more in the document table.

Document

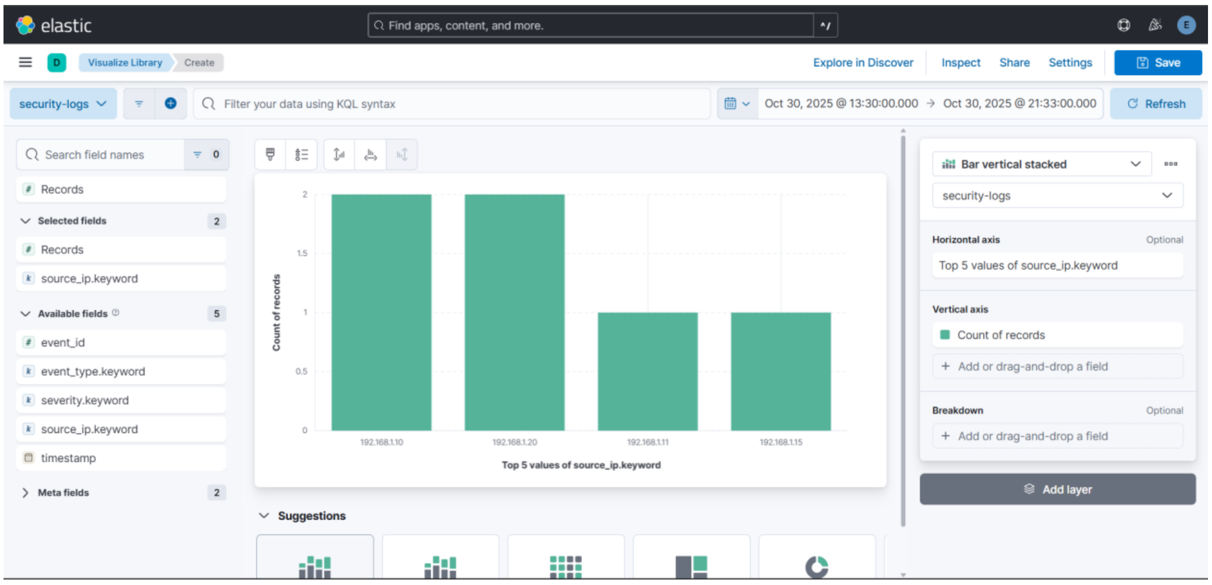
```
event_id 4,624 event_type Successful Login severity Info source_ip 192.168.1.20 timestamp Oct 30, 2025 @ 21:33:00.000 _id 2BMwNZoBR
8sAHQBhscxl _ignored  -  _index security-logs _score  -

event_id 4,625 event_type Failed Login severity Critical source_ip 192.168.1.11 timestamp Oct 30, 2025 @ 21:32:00.000 _id 1xMwNZoBR
8sAHQBhscwn _ignored  -  _index security-logs _score  -

event_id 4,625 event_type Failed Login severity Critical source_ip 192.168.1.10 timestamp Oct 30, 2025 @ 21:30:00.000 _id 1hMwNZoBR
8sAHQBhsMz5 _ignored  -  _index security-logs _score  -
```

7.1 Top Source IPs Generating Alerts

This visualization highlights the source IP addresses that are most frequently generating security alerts in the monitored environment. It helps analysts identify potential attack origins and recurring threat sources.

- **Visualization Type**
  - **Type:** Bar Vertical Stacked Chart
  - **Tool Used:** Kibana (Elastic Stack)
- **Data Source**
  - **Index Pattern:** security-logs
  - **Selected Fields:**

1. source_ip.keyword ⟶ Represents the source IP address generating alerts
2. Count of records ⟶ Represents the frequency of alerts per IP



Bar vertical stacked
security-logs
Horizontal axis: Top 5 values of source_ip.keyword
Vertical axis: Count of records
Breakdown: Add or drag-and-drop a field

7.2 Frequency of Critical Event IDs

The purpose of this visualization is to analyze how **critical events** occur over a period of time. By plotting the frequency of event IDs against timestamps, this chart helps to identify **spikes or unusual patterns** in alert generation, which may indicate potential system compromise or active attack attempts.

➢ **Visualization Type**

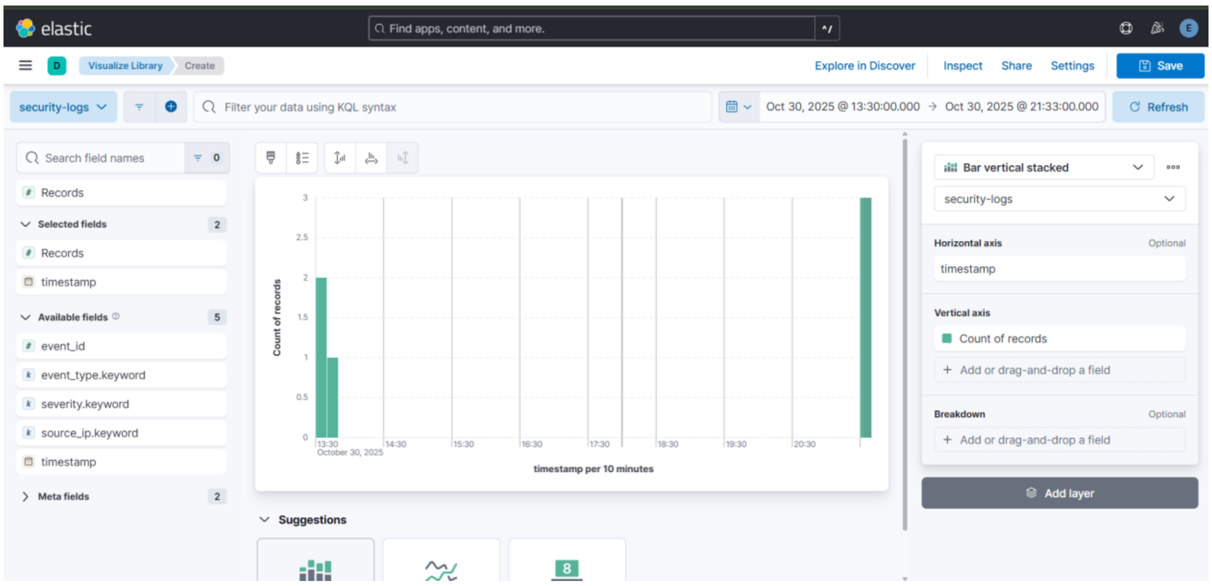
- **Type:** Bar Vertical Stacked Chart
- **Tool Used:** Kibana (Elastic Stack)

➢ **Data Source**

- **Index Pattern:** security-logs
- **Selected Fields:**

1. timestamp ⟶ Represents the event occurrence time

2. Count of records → Represents how many events occurred in each time interval



Oct 30, 2025 @ 13:30:00.000 → Oct 30, 2025 @ 21:33:00.000

security-logs

Bar vertical stacked

security-logs

Horizontal axis: timestamp

Vertical axis: Count of records

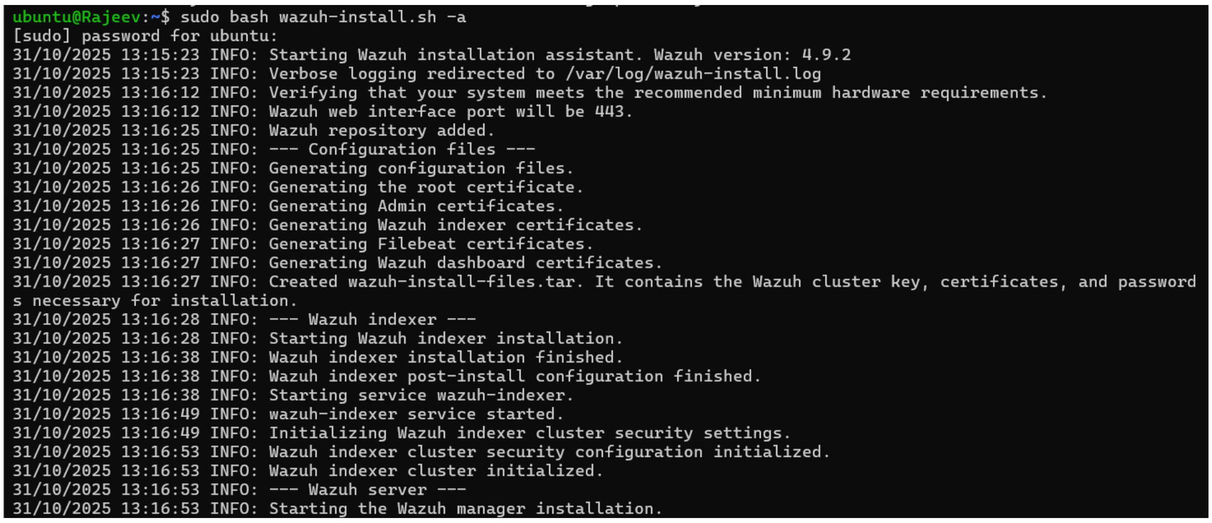
8. Monitoring Wazuh Dashboard

Wazuh is an **open-source Security Information and Event Management (SIEM)** and **XDR (Extended Detection and Response)** platform.

It helps monitor systems, detect threats, and ensure compliance across an IT environment. First

8.1 Wazuh Installation

The wazuh-install.sh -a command is run to perform an all-in-one installation of the Wazuh stack, which includes the Wazuh indexer, server, and dashboard.



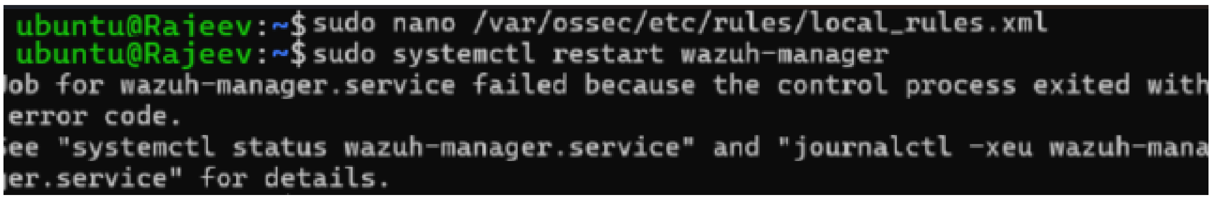
```
ubuntu@Rajeev:~$ sudo bash wazuh-install.sh -a
[sudo] password for ubuntu:
31/10/2025 13:15:23 INFO: Starting Wazuh installation assistant. Wazuh version: 4.9.2
31/10/2025 13:15:23 INFO: Verbose logging redirected to /var/log/wazuh-install.log
31/10/2025 13:16:12 INFO: Verifying that your system meets the recommended minimum hardware requirements.
31/10/2025 13:16:12 INFO: Wazuh web interface port will be 443.
31/10/2025 13:16:25 INFO: Wazuh repository added.
31/10/2025 13:16:25 INFO: --- Configuration files ---
31/10/2025 13:16:25 INFO: Generating configuration files.
31/10/2025 13:16:26 INFO: Generating the root certificate.
31/10/2025 13:16:26 INFO: Generating Admin certificates.
31/10/2025 13:16:26 INFO: Generating Wazuh indexer certificates.
31/10/2025 13:16:27 INFO: Generating Filebeat certificates.
31/10/2025 13:16:27 INFO: Generating Wazuh dashboard certificates.
31/10/2025 13:16:27 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and password
s necessary for installation.
31/10/2025 13:16:28 INFO: --- Wazuh indexer ---
31/10/2025 13:16:28 INFO: Starting Wazuh indexer installation.
31/10/2025 13:16:38 INFO: Wazuh indexer installation finished.
31/10/2025 13:16:38 INFO: Wazuh indexer post-install configuration finished.
31/10/2025 13:16:38 INFO: Starting service wazuh-indexer.
31/10/2025 13:16:49 INFO: wazuh-indexer service started.
31/10/2025 13:16:49 INFO: Initializing Wazuh indexer cluster security settings.
31/10/2025 13:16:53 INFO: Wazuh indexer cluster security configuration initialized.
31/10/2025 13:16:53 INFO: Wazuh indexer cluster initialized.
31/10/2025 13:16:53 INFO: --- Wazuh server ---
31/10/2025 13:16:53 INFO: Starting the Wazuh manager installation.
```

8.2 Custom Rule Configuration

In this step, the user opens the **local_rules.xml** file using the command:

sudo nano /var/ossec/etc/rules/local_rules.xml

This is done to **add or modify custom detection rules** in Wazuh. By editing this file, the user can define **specific alert conditions** or **tailor rule behavior** to meet organizational requirements.



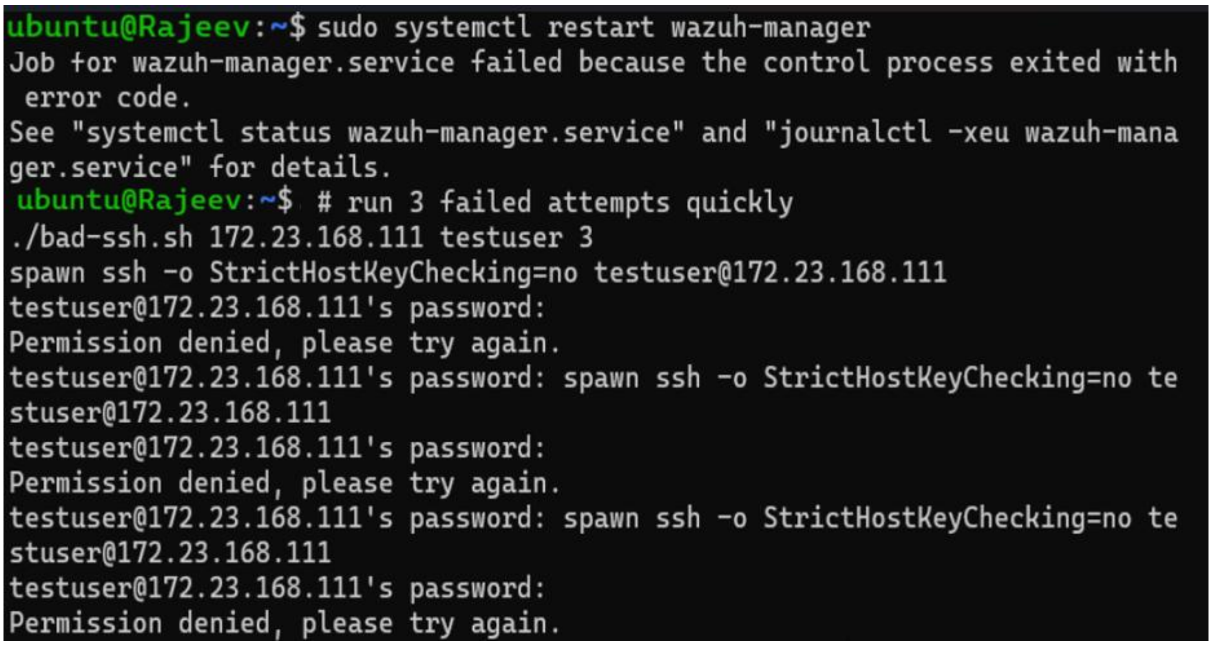
```
ubuntu@Rajeev:~$ sudo nano /var/ossec/etc/rules/local_rules.xml
ubuntu@Rajeev:~$ sudo systemctl restart wazuh-manager
ob for wazuh-manager.service failed because the control process exited with
error code.
ee "systemctl status wazuh-manager.service" and "journalctl -xeu wazuh-mana
er.service" for details.
```

8.3 Service Restart Failure

After saving the changes made to the **local_rules.xml** file, the user attempts to restart the Wazuh Manager service using the command:

sudo systemctl restart wazuh-manager

However, the restart **fails**, indicating that there is likely a **syntax or formatting error** in the recently edited local_rules.xml file. Such errors prevent Wazuh from reloading the configuration properly, requiring the user to review and correct the rule definitions before retrying the restart.



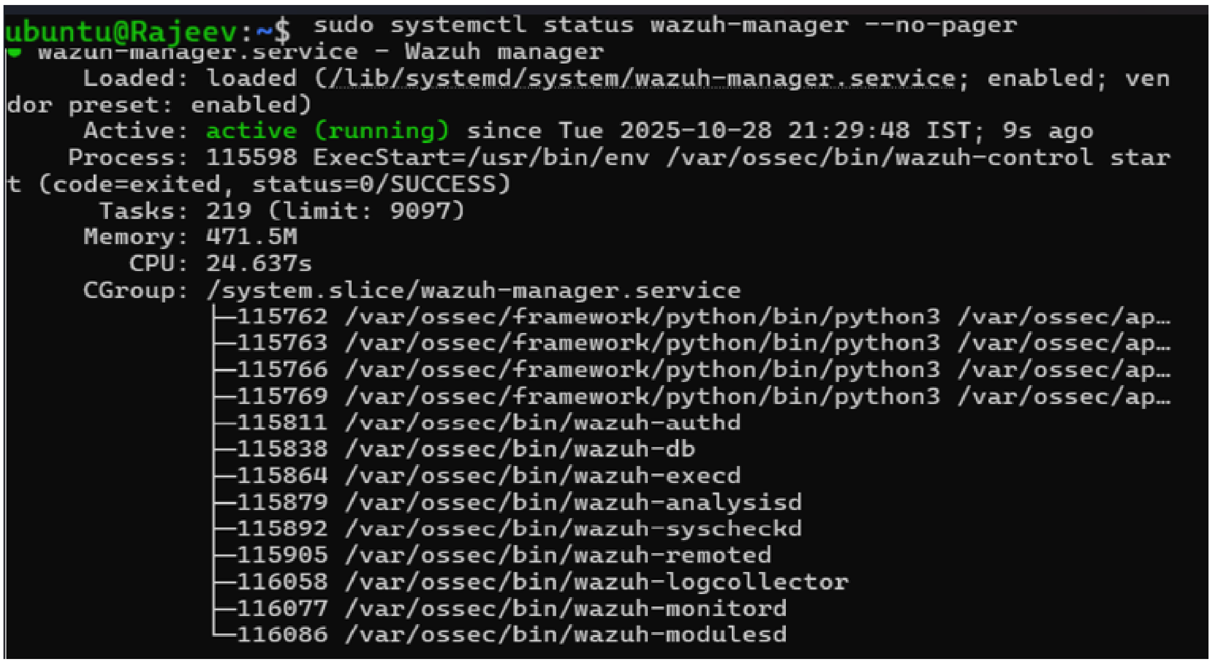
```
ubuntu@Rajeev:~$ sudo systemctl restart wazuh-manager
Job for wazuh-manager.service failed because the control process exited with
 error code.
See "systemctl status wazuh-manager.service" and "journalctl -xeu wazuh-mana
ger.service" for details.
 ubuntu@Rajeev:~$ # run 3 failed attempts quickly
./bad-ssh.sh 172.23.168.111 testuser 3
spawn ssh -o StrictHostKeyChecking=no testuser@172.23.168.111
testuser@172.23.168.111's password:
Permission denied, please try again.
testuser@172.23.168.111's password: spawn ssh -o StrictHostKeyChecking=no te
stuser@172.23.168.111
testuser@172.23.168.111's password:
Permission denied, please try again.
testuser@172.23.168.111's password: spawn ssh -o StrictHostKeyChecking=no te
stuser@172.23.168.111
testuser@172.23.168.111's password:
Permission denied, please try again.
```

8.4 Service Restart Success

After correcting the **syntax error** in the local_rules.xml file, the user verifies the Wazuh Manager service by running:

sudo systemctl status wazuh-manager

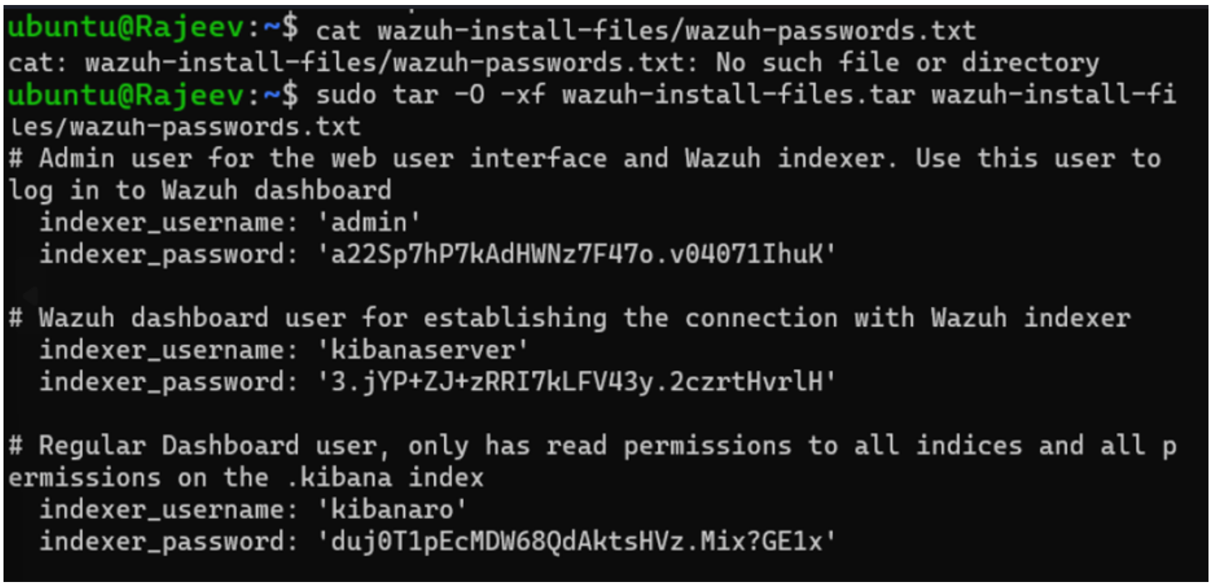
The output now shows the service as **“active (running)”**, confirming that Wazuh has successfully loaded the updated rule configurations and is operating without errors.



```
ubuntu@Rajeev:~$ sudo systemctl status wazuh-manager --no-pager
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; ven
dor preset: enabled)
     Active: active (running) since Tue 2025-10-28 21:29:48 IST; 9s ago
    Process: 115598 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control star
t (code=exited, status=0/SUCCESS)
      Tasks: 219 (limit: 9097)
     Memory: 471.5M
        CPU: 24.637s
     CGroup: /system.slice/wazuh-manager.service
             ├─115762 /var/ossec/framework/python/bin/python3 /var/ossec/ap…
             ├─115763 /var/ossec/framework/python/bin/python3 /var/ossec/ap…
             ├─115766 /var/ossec/framework/python/bin/python3 /var/ossec/ap…
             ├─115769 /var/ossec/framework/python/bin/python3 /var/ossec/ap…
             ├─115811 /var/ossec/bin/wazuh-authd
             ├─115838 /var/ossec/bin/wazuh-db
             ├─115864 /var/ossec/bin/wazuh-execd
             ├─115879 /var/ossec/bin/wazuh-analysisd
             ├─115892 /var/ossec/bin/wazuh-syscheckd
             ├─115905 /var/ossec/bin/wazuh-remoted
             ├─116058 /var/ossec/bin/wazuh-logcollector
             ├─116077 /var/ossec/bin/wazuh-monitord
             └─116086 /var/ossec/bin/wazuh-modulesd
```

8.5 Retrieving Passwords

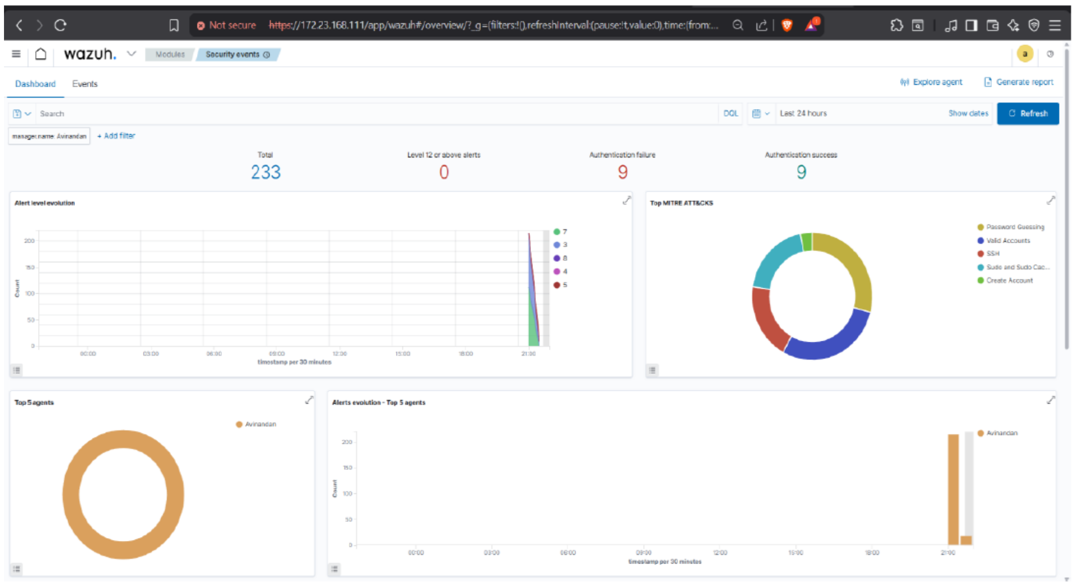
After the installation, the user extracts the **wazuh-install-files.tar** archive that was created during setup. This archive contains the **default generated passwords** for the **admin account** and other **internal Wazuh components**. Accessing these files allows the user to retrieve important credentials needed for logging into the Wazuh web interface and managing system configurations.



```
ubuntu@Rajeev:~$ cat wazuh-install-files/wazuh-passwords.txt
cat: wazuh-install-files/wazuh-passwords.txt: No such file or directory
ubuntu@Rajeev:~$ sudo tar -O -xf wazuh-install-files.tar wazuh-install-fi
les/wazuh-passwords.txt
# Admin user for the web user interface and Wazuh indexer. Use this user to
log in to Wazuh dashboard
  indexer_username: 'admin'
  indexer_password: 'a22Sp7hP7kAdHWNz7F47o.v04071IhuK'

# Wazuh dashboard user for establishing the connection with Wazuh indexer
  indexer_username: 'kibanaserver'
  indexer_password: '3.jYP+ZJ+zRRI7kLFV43y.2czrtHvrlH'

# Regular Dashboard user, only has read permissions to all indices and all p
ermissions on the .kibana index
  indexer_username: 'kibanaro'
  indexer_password: 'duj0T1pEcMDW68QdAktsHVz.Mix?GE1x'
```

8.8 Dashboard Overview

After successfully accessing the **Wazuh web dashboard**, the user navigates to the main overview page. The dashboard provides a comprehensive summary of the system's security status, displaying a total of **233 alerts** and **9 authentication failures**. Additionally, the **"TOP MITRE ATTACKS"** visualization highlights detected attack patterns, accurately identifying **"Password Guessing"** and **"SSH"**-related activities. This overview helps the user quickly assess the security posture and identify potential threats in the monitored environment.



9. Key Learnings

Through this activity, several important skills and insights were developed:

1. **Log Forensics** – Learned how to perform detailed log analysis after a security incident to identify the nature of attacks, trace their origin, and reconstruct the event timeline.
2. **Incident Documentation** – Understood the significance of proper incident recording and gained experience in creating clear, structured documentation for security events.
3. **SIEM Implementation** – Gained hands-on experience with implementing a SIEM system, including log ingestion, visualization creation, and alert configuration to enhance real-time threat monitoring.

10. Challenges

During the implementation, challenges included interpreting complex log data, correcting syntax errors in custom rule configurations, managing a high volume of alerts with false positives, and understanding the various dashboard visualizations. Additionally, maintaining clear and detailed incident documentation required extra effort and attention to accuracy.