**SSH Log Analysis Tool with Python on Kali Linux**

**Parsing and Visualizing Failed SSH Login Attempts**

Name: Parth Nagpal

Date:21/9/2025

**Introduction:**

**Purpose:**  
This project demonstrates how to analyze SSH login attempts on a Kali Linux VM using Python. The main goal is to **parse failed SSH logins** from the system log file (/var/log/auth.log), organize the data in a structured format, and **visualize the patterns** of failed login attempts by IP address.

**Tools and Technologies Used:**

* **Python 3** – main programming language for parsing and analysis
* **pandas** – to create structured DataFrames for analysis
* **matplotlib** – for visualizing failed login attempts with bar charts
* **Regular Expressions (regex)** – to extract relevant data from log lines
* **Kali Linux VM** – for generating and testing SSH login attempts

**Key Learning Outcomes:**

* How to read and parse system log files.
* Extract relevant security information using **regex**.
* Analyze and summarize data with **pandas**.
* Create clear visualizations of cybersecurity events using **matplotlib**.
* Understand how failed SSH attempts can be monitored for security purposes.

**Project Folder Structure**

**Description:**  
The project folder contains all the files necessary to run the SSH log analysis tool. Organizing files properly ensures clarity and easy execution.

**Folder:** ~/ssh\_log\_analysis

**Files included:**

1. **analyze\_ssh\_logs.py** – The main Python script that parses the log file and visualizes failed SSH login attempts.
2. **ssh\_auth.log** – A copy of the system SSH authentication log (/var/log/auth.log) containing failed login attempts for analysis.

**“Project folder structure showing analyze\_ssh\_logs.py and ssh\_auth.log in ~/ssh\_log\_analysis.”:**

A screenshot of a computer

AI-generated content may be incorrect.

**Python Script:**

**Description:**  
This script parses the SSH authentication log (ssh\_auth.log) to identify failed login attempts, analyzes the data using pandas, and visualizes the number of failed attempts per IP address with a bar chart.

**Python Code (analyze\_ssh\_logs.py):**

import re

import pandas as pd

import matplotlib.pyplot as plt

# Open the SSH auth log file

log\_file = 'ssh\_auth.log'

# Regex to find failed login attempts and IP addresses

failed\_login\_regex = r'Failed password for .\* from (\d+\.\d+\.\d+\.\d+)'

# Parse the log file

with open(log\_file, 'r') as file:

logs = file.readlines()

# Extract IP addresses from failed attempts

failed\_ips = [re.search(failed\_login\_regex, line).group(1)

for line in logs if re.search(failed\_login\_regex, line)]

# Create a DataFrame with counts per IP

df = pd.DataFrame(failed\_ips, columns=['IP'])

ip\_counts = df['IP'].value\_counts().reset\_index()

ip\_counts.columns = ['IP', 'Failed\_Attempts']

print(ip\_counts)

# Plot bar chart of failed login attempts by IP

plt.figure(figsize=(10,6))

plt.bar(ip\_counts['IP'], ip\_counts['Failed\_Attempts'], color='red')

plt.xlabel('IP Address')

plt.ylabel('Number of Failed Attempts')

plt.title('Failed SSH Login Attempts by IP')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

**Simulating Failed SSH Login Attempts**

**Description:**  
To analyze failed SSH logins, we need some test data. This step shows how to generate multiple failed login attempts on the Kali Linux VM.

**Command (run in Terminal):**

for i in {1..15}; do

ssh -o StrictHostKeyChecking=no -o PasswordAuthentication=no kali@localhost exit;done

* This loop attempts 15 SSH logins with wrong authentication.
* It generates failed login entries in /var/log/auth.log that can later be parsed by the Python script.

A screen shot of a computer

AI-generated content may be incorrect.

**Copying /var/log/auth.log to Project Folder**

**Description:**  
After generating failed SSH login attempts, we need to copy the authentication log to the project folder so the Python script can parse it. Correct permissions ensure the script can read the file.

**Commands (run in Terminal):**

cp /var/log/auth.log ~/ssh\_log\_analysis/ssh\_auth.log

chown kali:kali ~/ssh\_log\_analysis/ssh\_auth.log

* cp copies the log file to your project folder (~/ssh\_log\_analysis/).
* chown sets ownership so your user can read/write the copied log.
* This step ensures the Python script can access the file without permission errors.

A screenshot of a computer

AI-generated content may be incorrect.

**Running the Python Script**

**Description:**  
This step demonstrates executing the analyze\_ssh\_logs.py script to parse the copied SSH authentication log and display the DataFrame of failed login attempts.

**Command (run in Terminal):**

python3 ~/ssh\_log\_analysis/analyze\_ssh\_logs.py

A screenshot of a computer

AI-generated content may be incorrect.

**Visualizing Failed SSH Attempts by IP**

**Description:**  
This step shows the graphical representation of failed SSH login attempts per IP address using a bar chart. The chart helps quickly identify IPs with the most failed login attempts.

Python Code (already included in analyze\_ssh\_logs.py):

plt.figure(figsize=(10,6))

plt.bar(ip\_counts['IP'], ip\_counts['Failed\_Attempts'], color='red')

plt.xlabel('IP Address')

plt.ylabel('Number of Failed Attempts')

plt.title('Failed SSH Login Attempts by IP')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

A screenshot of a computer

AI-generated content may be incorrect.

**Conclusion / Learnings**

**Description:**  
This project demonstrates how to parse SSH authentication logs, analyze failed login attempts, and visualize patterns using Python. It highlights practical skills in log analysis and basic cybersecurity monitoring.

**Key Learnings:**

* Parsed and analyzed SSH log files (/var/log/auth.log) for failed login attempts.
* Learned to use **regex** to extract relevant information from logs.
* Created **pandas DataFrames** to organize and count failed login attempts by IP.
* Visualized data using **matplotlib** to identify suspicious activity easily.
* Gained hands-on experience in **cybersecurity monitoring** and **log analysis**.