Splunk Log Analysis Workflow

I began by launching my Splunk instance and navigating to **Settings → Add Data** to ingest the practice dataset. I uploaded the file **bns_practice.csv**, selected the source type as **CSV**, and indexed the data under a new index named **bns_practice**.

Once the data was successfully ingested, I performed multiple searches and investigations using SPL to simulate common SOC analysis scenarios.

Searching and Filtering Activities

1. Investigating activity from a specific user

```
index=bns_practice user="Mbua"
```

---

2. Identifying failed login attempts by user and source IP

```
index=bns_practice event_type="login_attempt" login_status="failure"
| stats count by user, src_ip
```

---

3. Detecting known malicious or scanning tools

```
index=bns_practice user_agent IN ("sqlmap", "Nikito", "nmap", "dirbuster")
```

---

4. Listing events with malicious reputation

```
index=bns_practice reputation="Malicious"
```

---

5. Analyzing event volume by country

```
index=bns_practice | stats count by country
```

---

6. Monitoring access from high-risk or unusual countries

```
index=bns_practice country IN ("Russia", "Brazil", "India")
```

---

7. Correlating failed logins with high-risk regions

index=bns_practice event_type="login_attempt" login_status="failure"

country IN ("Russia", "Brazil", "India")

---

8. Dashboard Creation

To visualize user activity trends, I created a bar chart showing the most frequent users in the logs:

index=bns_practice | top user

I selected **Visualization → Bar Chart** to display the results clearly.

---

9. Visualizing automated attack behavior

To track suspicious tools used during access attempts, I ran:

index=bns_practice user_agent IN ("dirbuster", "sqlmap", "Nikito", "nmap")

| stats count by user_agent

This helped highlight automated scanning and attack patterns.

---

10. Alert Configuration

Finally, I created a real-time alert to detect successful logins involving suspicious tools:

index=bns_practice event_type="login_attempt" login_status="success"

user_agent IN ("sqlmap", "dirbuster", "python-requests", "nmap")

I clicked **Save As → Alert**, configured notification settings, and saved the alert to simulate SOC monitoring workflows.