



SOC Fundamentals and Operations

Purpose of a SOC (Security Operations Center):

A SOC is a dedicated unit that continuously monitors, detects, investigates, and responds to cybersecurity incidents.

Its goal is to protect an organization's assets through **proactive threat detection, incident response, and continuous monitoring**.

Key Functions:

Log Analysis: Reviewing collected logs for anomalies or suspicious activities.

Alert Triage: Prioritizing and investigating security alerts.

Threat Intelligence Integration: Using external data sources to identify new and emerging threats.

Incident Response: Taking immediate actions to contain and remediate attacks.

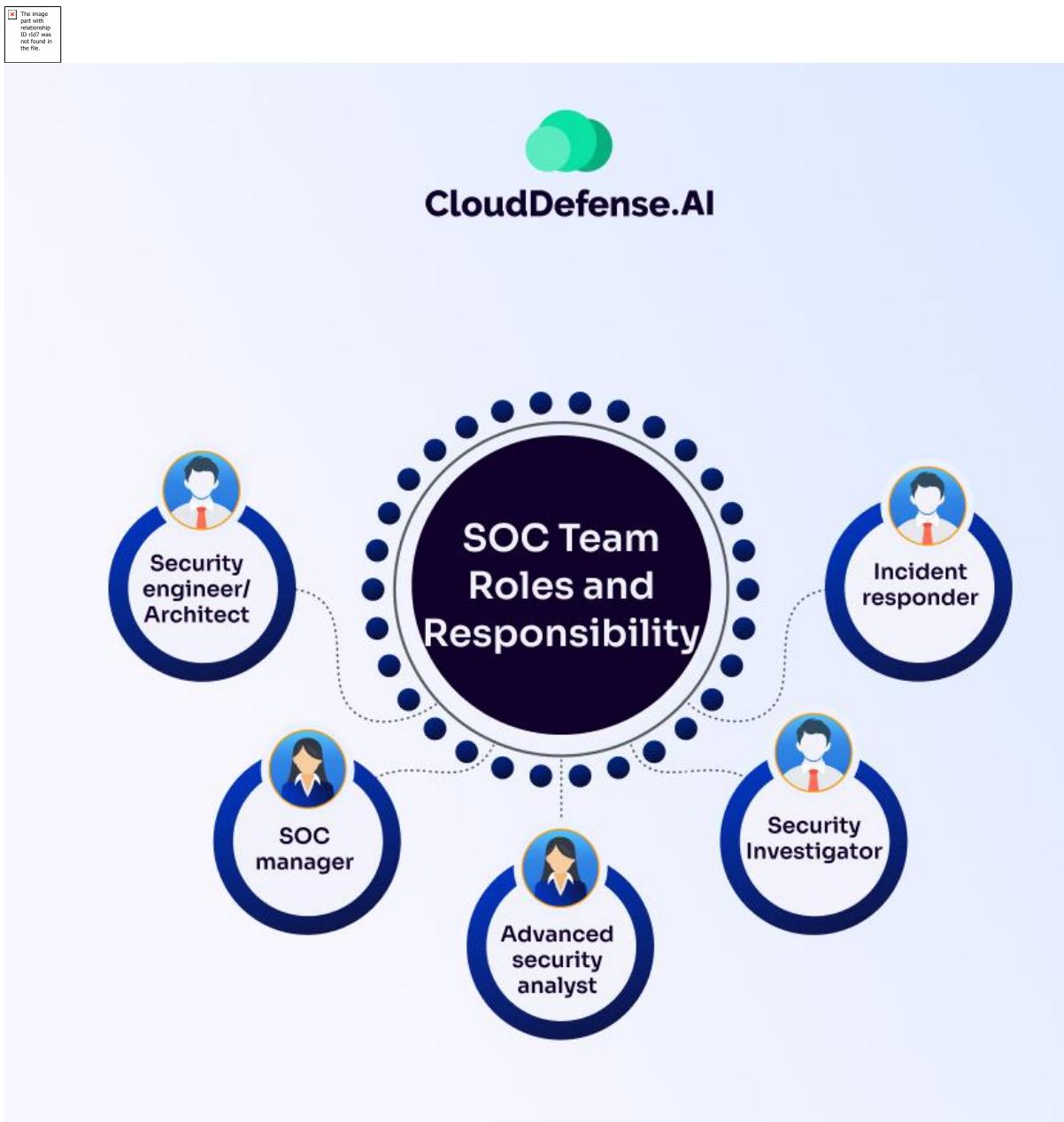
Reporting and Documentation: Maintaining detailed records of incidents and resolutions.

Roles in a SOC:

Role	Responsibility
Tier 1 Analyst (L1)	Monitors alerts, performs initial triage, and escalates true positives.
Tier 2 Analyst (L2)	Conducts deeper investigations, performs correlation and threat hunting.
Tier 3 Analyst (L3)	Handles advanced incidents, malware analysis, and forensic investigations.
SOC Manager	Oversees SOC operations, ensures KPIs are met, and reports to management.
Threat Hunter	Proactively searches for undetected threats using advanced analytics and threat intelligence.



SOC team structure diagram





2. Security Monitoring Basics:

Objectives:

Detect anomalies and unauthorized access.

Identify malware infections or suspicious behaviors.

Monitor compliance and detect policy violations.

Key Metrics:

False Positive: An alert that looks malicious but isn't.

False Negative: A missed alert that was actually malicious.

MTTD (Mean Time to Detect): Average time to identify a threat.

MTTR (Mean Time to Respond): Average time to contain and resolve a threat.

1 Setup in Splunk

1.1 Open Splunk Web → **Add Data** → **Upload Files**

1.2 Upload these 3 logs one by one:

windows_security.log

firewall.log

web_access.log

1.3 Assign:

Source type → log

Index → practice

Host → lab1

Click **Start Searching**

Step 2: Basic Detection Queries (SPL)

2.1 Detect Anomalies — multiple failed logins

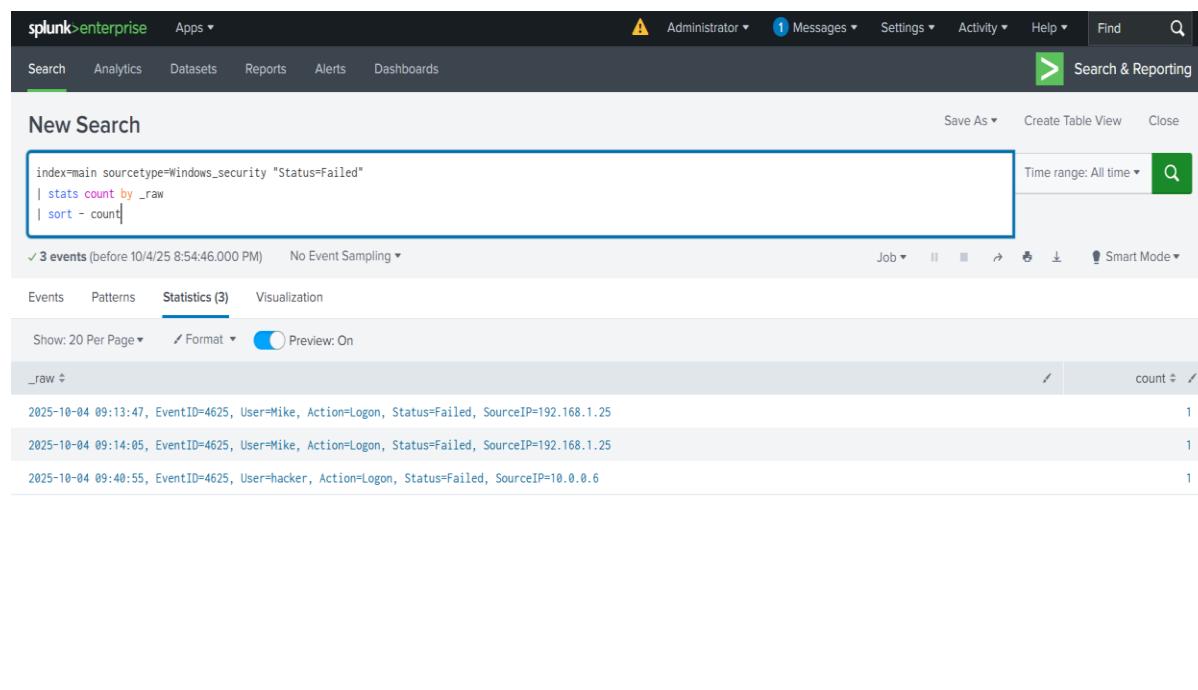
CMD:

```
index=practice "Status=Failed" OR 401
```

```
| stats count by User, SourceIP
```

```
| where count > 3
```

Screenshot:



The screenshot shows the Splunk Enterprise search interface. The search bar contains the following SPL query:

```
index=main sourcetype=Windows_security "Status=Failed"
| stats count by _raw
| sort - count
```

The search results table shows three events:

_raw	count
2025-10-04 09:13:47, EventID=4625, User=Mike, Action=Logon, Status=Failed, SourceIP=192.168.1.25	1
2025-10-04 09:14:05, EventID=4625, User=Mike, Action=Logon, Status=Failed, SourceIP=192.168.1.25	1
2025-10-04 09:40:55, EventID=4625, User=hacker, Action=Logon, Status=Failed, SourceIP=10.0.0.6	1



2.2 Detect Successful Admin Actions

The screenshot shows the Splunk Enterprise search interface. The search bar contains the following SPL query:

```
index=main sourcetype=Windows_security "Status=Success"
| rex "User=(?<User>[^,]+), Action=(?<Action>[^,]+), Status=(?<Status>[^,]+), (TargetUser=(?<TargetUser>[^,]+), )?"*
| table _time, User, Action, TargetUser, Status
| sort _time
```

The results table shows three events:

_time	User	Action	TargetUser	Status
2025-10-04 09:11:22	John	Logon		Success
2025-10-04 09:20:44	Admin	CreateUser	hacker	Success
2025-10-04 09:32:10	Admin	PrivilegeAssigned		Success

2.3 Detect All Logons

The screenshot shows the Splunk Enterprise search interface. The search bar contains the following SPL query:

```
index=main sourcetype=Windows_security (Action=Logon)
| rex "User=(?<User>[^,]+), Action=(?<Action>[^,]+), Status=(?<Status>[^,]+), SourceIP=(?<SourceIP>[\d\.\.]+)"*
| table _time, User, Action, Status, SourceIP
| sort _time
```

The results table shows four events:

_time	User	Action	Status	SourceIP
2025-10-04 09:11:22	John	Logon	Success	192.168.1.15
2025-10-04 09:13:47	Mike	Logon	Failed	192.168.1.25
2025-10-04 09:14:05	Mike	Logon	Failed	192.168.1.25
2025-10-04 09:40:55	hacker	Logon	Failed	10.0.0.6

Log Management Fundamentals:

1.1 Objective

To understand how logs are collected, normalized, stored, retained, and analyzed in a SIEM (Splunk).

This task demonstrates the **log lifecycle**, **types of logs**, and **practical querying techniques**.

Log Types Used :

Log File	Description	Source Type
windows_security.log	Windows Event Logs (logons, privilege changes)	Windows_security
web_Acess.log	Web server access logs (HTTP requests)	web_Acess
firewall.log	Network firewall activity logs	firewall

1.2 Normalization in Splunk (Field Extraction):

To structure Windows log data into readable columns:

CMD:

```
index=main sourcetype=Windows_security
| rex "EventID=(?<EventID>\d+), User=(?<User>[^,]+), Action=(?<Action>[^,]+),
Status=(?<Status>[^,]+)(, SourceIP=(?<SourceIP>[\d.]+))?"
| table _time, EventID, User, Action, Status, SourceIP
| sort _time
```

This extracts Event ID, User, Action, and Status from raw log text.



Screenshot: Normalized field output in Splunk

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** Contains the SPL query:

```
index=main sourcetype=Windows_security  
| rex "EventID=(?<EventID>\d+), User=(?<User>[^,]+), Action=(?<Action>[^,]+), Status=(?<Status>[^,]+)(, SourceIP=(?<SourceIP>[\d\.\.]+))?"  
| table _time, EventID, User, Action, Status, SourceIP  
| sort _time
```
- Results Panel:** Shows 6 events from October 4, 2025, at 9:15:20.000 PM. The events are:

_time	EventID	User	Action	Status	SourceIP
2025-10-04 09:11:22	4624	John	Logon	Success	192.168.1.15
2025-10-04 09:13:47	4625	Mike	Logon	Failed	192.168.1.25
2025-10-04 09:14:05	4625	Mike	Logon	Failed	192.168.1.25
2025-10-04 09:20:44	4720	Admin	CreateUser	Success	
2025-10-04 09:32:10	4672	Admin	PrivilegeAssigned	Success	
2025-10-04 09:40:55	4625	hacker	Logon	Failed	10.0.0.6

1.3 Failed Logins Detection (KQL Equivalent in SPL)

CMD:

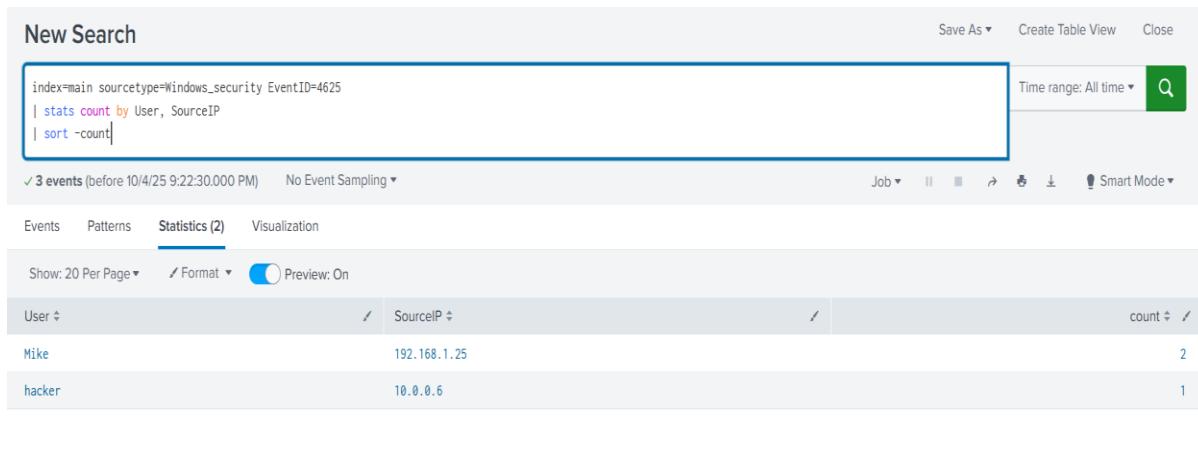
```
index=main sourcetype=Windows_security EventID=4625
```

```
| stats count by User, SourceIP
```

```
| sort -count
```

Detects repeated failed logins from same user/IP.

Screenshot: Failed login count table



The screenshot shows a Splunk search interface titled "New Search". The search bar contains the following command:

```
index=main sourcetype=Windows_security EventID=4625
| stats count by User, SourceIP
| sort -count
```

The results table displays three events:

User	SourceIP	count
Mike	192.168.1.25	2
hacker	10.0.0.6	1

1.4 Successful Admin Actions (Policy Violations Example)

Query to find privilege changes or new user creation:

CMD:

```
index=main sourcetype=Windows_security (Action=CreateUser OR Action=PrivilegeAssigned)
```

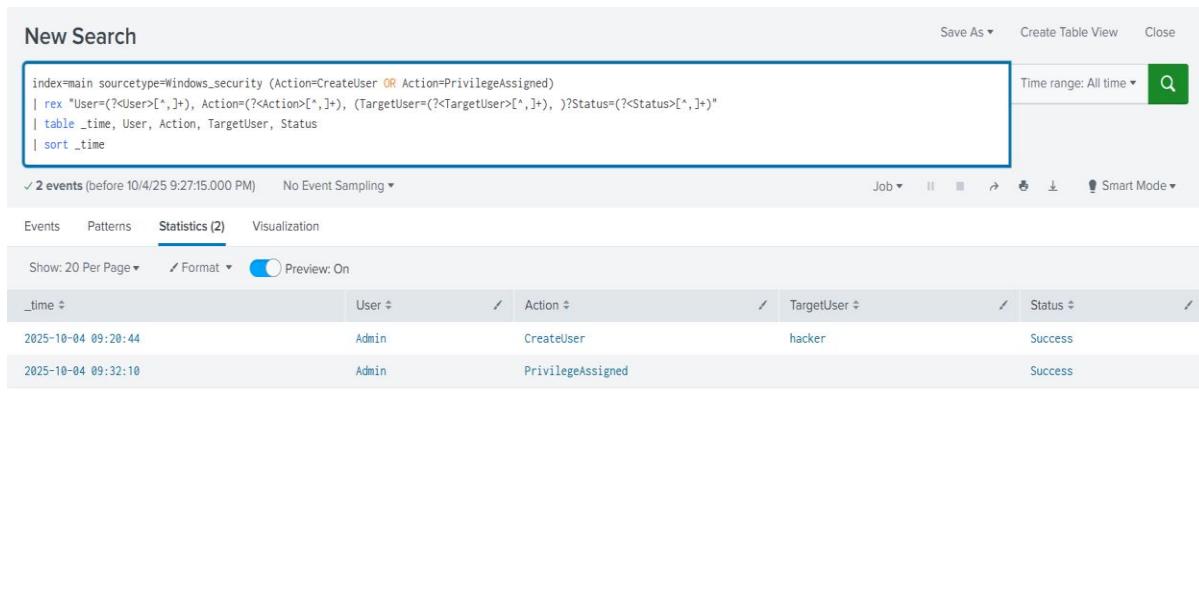
```
| rex "User=(?<User>[^,]+), Action=(?<Action>[^,]+), (TargetUser=(?<TargetUser>[^,]+), )?Status=(?<Status>[^,]+)"
```

```
| table _time, User, Action, TargetUser, Status
```

```
| sort _time
```

Shows when an Admin creates or modifies users (potential policy violation).

Screenshot: Query results for privilege changes



New Search

```
index=main sourcetype=Windows_security (Action/CreateUser OR Action=PrivilegeAssigned)
| rex "User=(?<User>[^,]+), Action=(?<Action>[^,]+), TargetUser=(?<TargetUser>[^,]+), Status=(?<Status>[^,]+)"
| table _time, User, Action, TargetUser, Status
| sort _time
```

2 events (before 10/4/25 9:27:15.000 PM) No Event Sampling ▾

Events Patterns Statistics (2) Visualization

Show: 20 Per Page ▾ Format ▾ Preview: On

_time	User	Action	TargetUser	Status
2025-10-04 09:20:44	Admin	CreateUser	hacker	Success
2025-10-04 09:32:10	Admin	PrivilegeAssigned		Success

1.5 Web Access Log Analysis

Extract client IP and HTTP method:

CMD:

```
index=main sourcetype=web_Acess
```

```
| rex "(?<ClientIP>\d+\.\d+\.\d+\.\d+).*\] \"(?<Method>[A-Z]+) (?<URL>[\^s]+)"

| table _time, ClientIP, Method, URL

| sort _time
```

Displays which IPs accessed which URLs on your web server.

Screenshot: Web access log output

New Search

```
index=main sourcetype=web_Acess
| rex "(?<ClientIP>\d+\.\d+\.\d+\.*)\ ] \"(?<Method>[A-Z]+) \(?<URL>[^\"s]+)"
| table _time, ClientIP, Method, URL
| sort _time
```

7 events (before 10/4/25 9:28:18.000 PM) No Event Sampling ▾

Save As ▾ Create Table View Close Time range: All time ▾ 

Events Patterns Statistics (7) Visualization Job ▾ II ■ ▶ 🔍 ⌂ Smart Mode ▾

Show: 20 Per Page ▾ Format ▾ Preview: On

_time	ClientIP	Method	URL
2025-10-04 09:10:22	192.168.1.20	GET	/login
2025-10-04 09:12:33	192.168.1.22	POST	/login
2025-10-04 09:12:34	192.168.1.22	POST	/login
2025-10-04 09:12:36	192.168.1.22	POST	/login
2025-10-04 09:12:40	192.168.1.22	POST	/login
2025-10-04 09:35:11	192.168.1.44	GET	/admin
2025-10-04 09:45:21	192.168.1.55	GET	/confidential/data

1.6 Firewall Log Analysis

Check for blocked connections:

CMD:

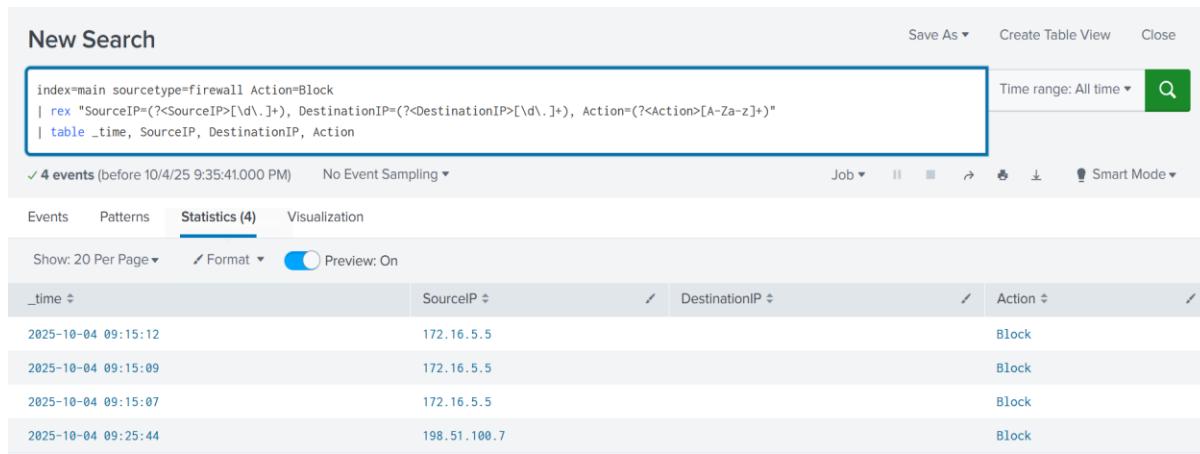
```
index=main sourcetype=firewall Action=Block
```

```
| rex "SourceIP=(?<SourceIP>[\d\.]+), DestinationIP=(?<DestinationIP>[\d\.]+), Action=(?<Action>[A-Za-z]+)"
```

```
| table _time, SourceIP, DestinationIP, Action
```

Identifies blocked IPs and traffic sources.

Screenshot: Firewall log search results



New Search

index=main sourcetype=firewall Action=Block
| rex "SourceIP=(?<SourceIP>[\d.]+), DestinationIP=(?<DestinationIP>[\d.]+), Action=(?<Action>[A-Za-z]+)"
| table _time, SourceIP, DestinationIP, Action

Time range: All time ▾ Q

✓ 4 events (before 10/4/25 9:35:41.000 PM) No Event Sampling ▾ Job ▾ II ■ ▾ Smart Mode ▾

Events Patterns Statistics (4) Visualization

Show: 20 Per Page ▾ Format ▾ Preview: On

_time	SourceIP	DestinationIP	Action
2025-10-04 09:15:12	172.16.5.5		Block
2025-10-04 09:15:09	172.16.5.5		Block
2025-10-04 09:15:07	172.16.5.5		Block
2025-10-04 09:25:44	198.51.100.7		Block

Outcome:

Understood how logs are collected, normalized, and analyzed in Splunk.

Learned how to extract fields and detect suspicious activity using SPL queries.

Practiced with real log samples representing Windows, Web, and Firewall data.