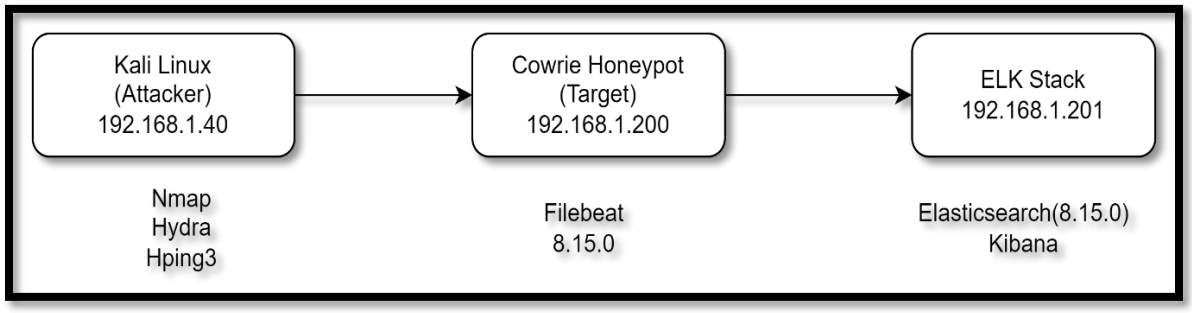
SOC ANALYST PROJECT:

Project Objectives:

Developing an automated system for SOC team to select from the attack types. The system helps to scan the targets for IP addresses and ports and let SOC team to choose a target server and specify the type of attack to perform.
The goal of this system is to help SOC teams verify that their SIEM can detect and alert these attacks.
All executions will be logged the attack type, execution time, and involved IP addresses, with logs saved in '/var/log' for further analysis.

Architecture:

Below shows the setup that I installed and configured on my local machine:



Description:

Set up **Kali Linux** to perform various types of cyber-attacks (Nmap, Hydra, hping3) against the Cowrie Honeypot to test the honeypot's effectiveness in capturing data.
Install and Configure Cowrie on an ubuntu machine to act as a honeypot, properly capturing detailed logs of all interactions. Cowrie is a medium- interaction SSH (port:2222) and Telnet (Port:2223) honeypot designed to log attacks performed by attackers. Filebeat, a lightweight log shipper is installed as an agent on this server to monitor log files in locations (/home/cowrie/cowrie/Var/log/cowrie and /var/log/kern.log) and collects log events and forwards them to Elasticsearch for indexing.
Install ELK Stack (Elasticsearch, Kibana) to collect, store and visualize the logs and use Kibana to visualize and analyze the data collected by cowrie and alerting the respective events.

Contents of script:

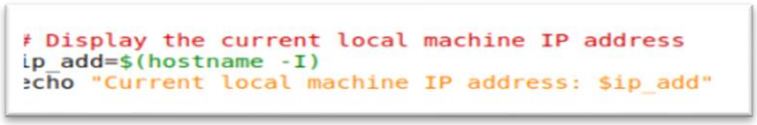
- Creating a Log File

- Displaying Attacker Server's IP Address
- Scanning Network for victim's IP Address
- Executing different types of Attacks
- Verifying logs in SIEM And Alerts
- Conclusion

Creating a Log File:

When running the script, the script first creates a file in the /var/log/ folder to log the script execution information's such as date, time, attack type and IP address of the attack.

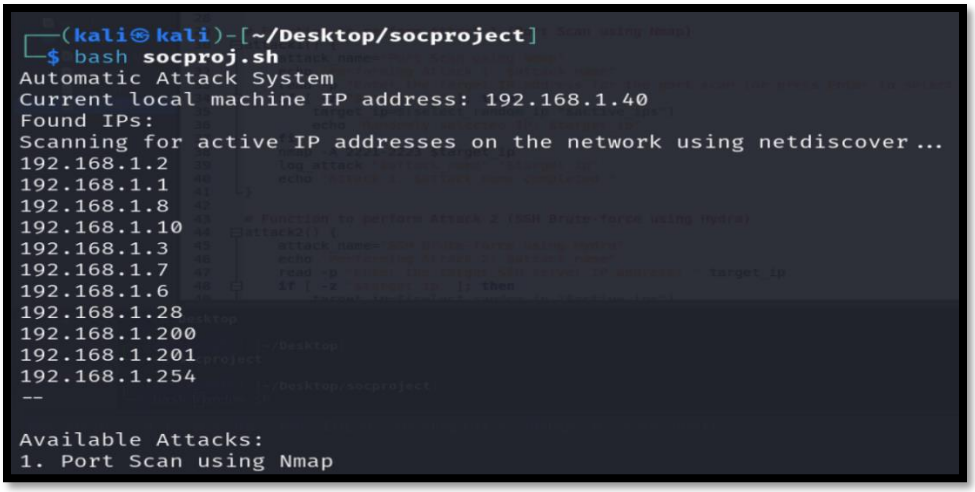
Displaying Attacker Server's IP Address:



When running the script, it will display the Attacker's server IP Address of the machine.

Scanning Network for Victim's IP Address:

The system scans for the active IP addresses on the network using netdiscover where the user can choose which IP address to perform the attack(victim).

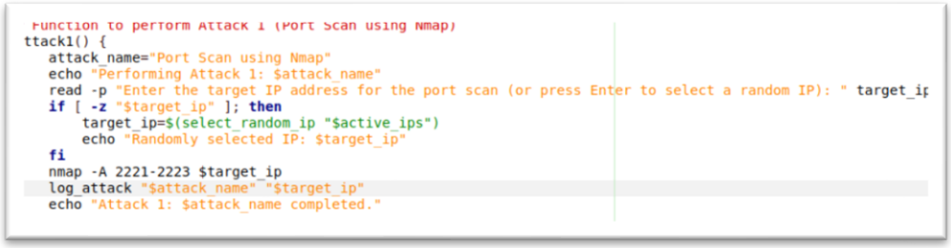


Executing Different Types of Attack :

The script prompt to choose three options. User can choose options 1 for port scan, option 2 for Brute Force and option 3 for Dos attack or 'r' for randomly choose an option.

- **Attack 1: Port Scan using Nmap**

Below shows the script for option 1 port scanning:



The Nmap command performs an aggressive scan on ports 2222 and 2223 of the target Ip address. -A is the flag enables aggressive scanning which gives detailed checking on the target. It checks for open ports, detects the services running on the ports, and determine the versions of the services. It also provides the information about SSH host keys (eg: ECDSA, ED25519, RSA) which are cryptographic keys used to identify the server, and for authentication purposes. Port 2222 is likely to part of the Cowrie honeypot setup, simulating an SSH service to capture attacker behaviour.

```
Choose an attack (1-3 or 'r' for random) or 'q' to quit: 1
Performing Attack 1: Port Scan using Nmap
Enter the target IP address for the port scan (or press Enter to select a ra
ndom IP): 192.168.1.200
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-17 03:23 EDT
Failed to resolve "2221-2223".
Nmap scan report for 192.168.1.200
Host is up (0.00085s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; proto
col 2.0)
| ssh-hostkey:
|   256 68:bf:f0:4f:5d:fc:3c:dc:d2:92:7b:d5:24:e3:be:84 (ECDSA)
|_  256 db:53:5d:47:e8:87:5e:b8:8c:ef:b7:30:0b:de:c0:25 (ED25519)
2222/tcp open  ssh     OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
| ssh-hostkey:
|   2048 b6:c3:26:34:12:f3:57:a5:18:84:9b:31:51:9a:b6:b3 (RSA)
|   256 5b:39:79:8f:ee:7d:4d:46:6f:6f:dc:44:5b:76:ac:10 (ECDSA)
|_  256 60:15:7f:88:c4:2a:2c:cf:6f:9e:2c:d1:df:b3:4e:1f (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Below shows the logs saved in the /var/log/attacks.log

```
Attack: Port Scan using Nmap
Target IP: 192.168.1.200
Time of execution: Tue Sep 17 03:23:11 AM EDT 2024
```

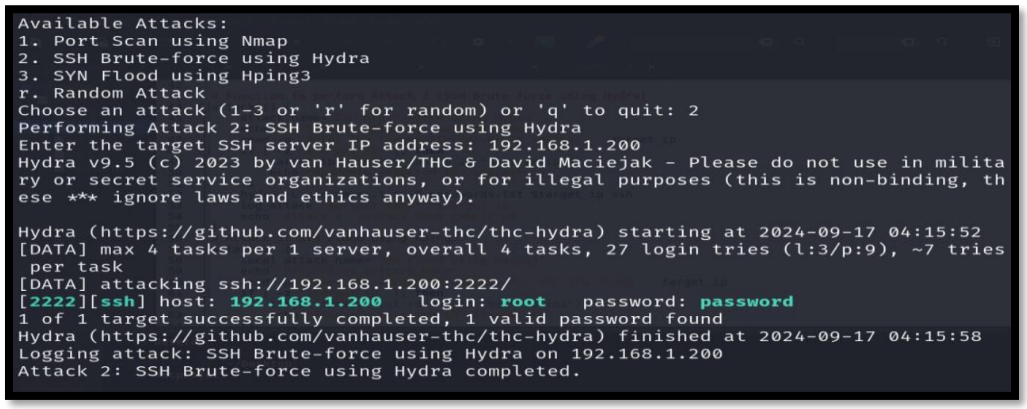
Below shows the logs visible in ELK:

- **Attack 2: Brute force Attack**

Below shows the script for option 2 Brute Force :

```
 Function to perform Attack 2 (SSH Brute-force using Hydra)
ttack2() {
    attack_name="SSH Brute-force using Hydra"
    echo "Performing Attack 2: $attack_name"
    read -p "Enter the target SSH server IP address: " target_ip
    if [ -z "$target_ip" ]; then
        target_ip=$(select_random_ip "$active_ips")
        echo "Randomly selected IP: $target_ip"
    fi
    hydra -L usernames.lst -P passwords.lst ssh://$target_ip:2222 -t 4
    log_attack "$attack_name" "$target_ip"
    echo "Attack 2: $attack_name completed."
```

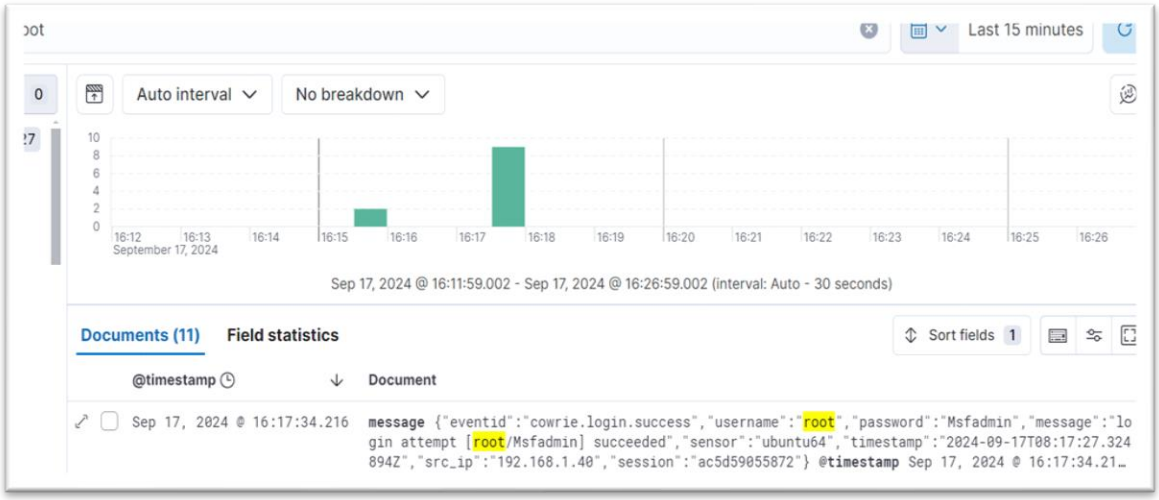
The second attack is brute-forcing using the tools Hydra to simulate the unauthorized access attempts on the cowrie honeypot using ssh or telnet server. Cowrie typically runs on a port (2222 or 2223). To initiate the attack, username and passwords list can provided to hydra Which systematically attempts to guess the correct login credentials.



Below shows the logs saved in the /var/log/attacks.log:

Below shows the logs visible in ELK:

The logs which are used to capture the successful login attempts.



Below shows the logs visible for failed login in ELK:

In brute-force attempt, it's important not only to track successful logins but also to monitor failed login attempts. Failed attempts provide helps to identify persistent

threats, and reveal where attacks are coming from. Monitoring these attempts can trigger alerts to prevent further attacks, while logging failed attempts offer valuable data for analyzing attacker behavior.

- **Attack 3: Denial of Service**

Below shows the script for option 3 Dos attack :

```
# Attack 3: SYN Flood using Hping3
attack3() {
    local attack_name="SYN Flood using Hping3"
    echo "Performing $attack_name"
    read -p "Enter the target IP address for the SYN flood: " target_ip
    if [ -z "$target_ip" ]; then
        target_ip=$(select_random_ip "$active_ips")
        echo "Randomly selected IP: $target_ip"
    fi
    sudo hping3 -S --flood -p 2222 $target_ip
    log_attack "$attack_name" "$target_ip"
    echo "$attack_name completed."
}
```

The third attack is Denial of service (Dos) in which makes a system, service or network resource unavailable by overwhelming it with a flood of illegitimate traffic or requests. In a TCP SYN flood, sends a large number of TCP SYN packets to a target system, exhausting its resources and preventing legitimate connections. Dos attacks, such as SYN flooding, can be carried out using a tool called Hping3.In Hping3, the -S option sends SYN packets (for a TCP SYN flood attack), while --flood sends packets as fast as possible without waiting for a reply, creating a flood. The -p option specifies the target port (e.g., 2222) on the actual IP address (192.168.1.200) of the target machine.

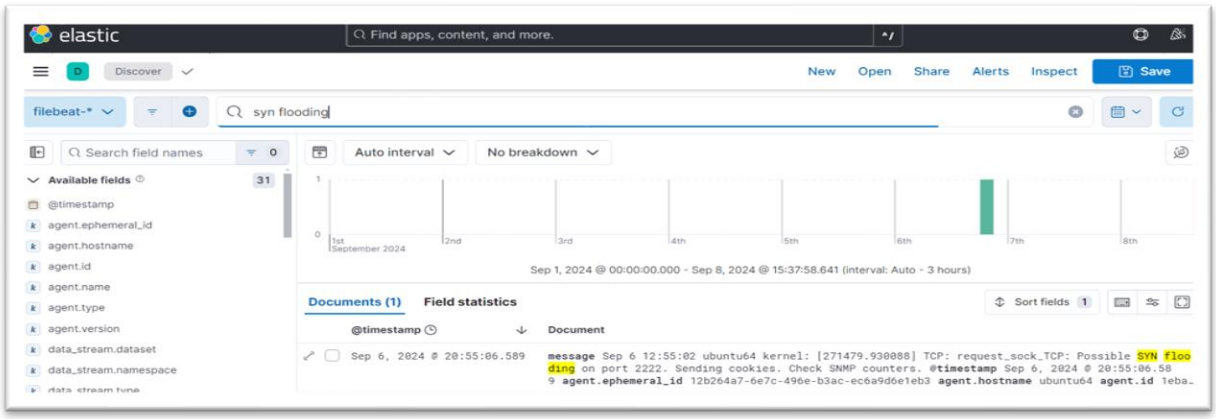
```
Choose an attack (1-3 or 'r' for random) or 'q' to quit: 3
Performing SYN Flood using Hping3
Enter the target IP address for the SYN flood: 192.168.1.200
HPING 192.168.1.200 (eth0 192.168.1.200): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.200 hping statistic ---
135227 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
Logging attack: SYN Flood using Hping3 on 192.168.1.200
SYN Flood using Hping3 completed.
```

Below shows the logs saved in the /var/log/attacks.log:

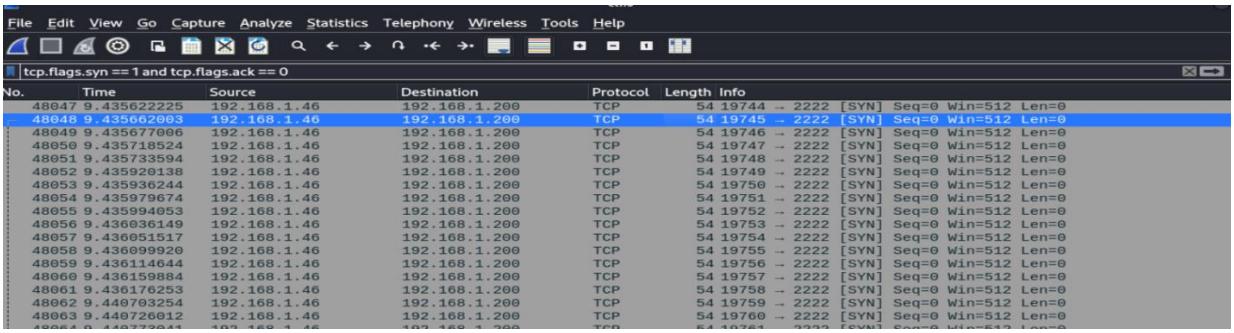
```
Attack: SYN Flood using Hping3
Target IP: 192.168.1.200
Time of execution: Fri Sep  6 09:05:29 AM EDT 2024
```

Below shows the logs visible in ELK:

The logs capturing the SYN flooding for the DoS attack are displayed in ELK.



Below shows the packets capture in Wireshark:



This captures the sending of packets in Wireshark, as the system will begin flooding the target server (192.168.1.200) with SYN packets. The sending of packets can only be manually stopped by the user.

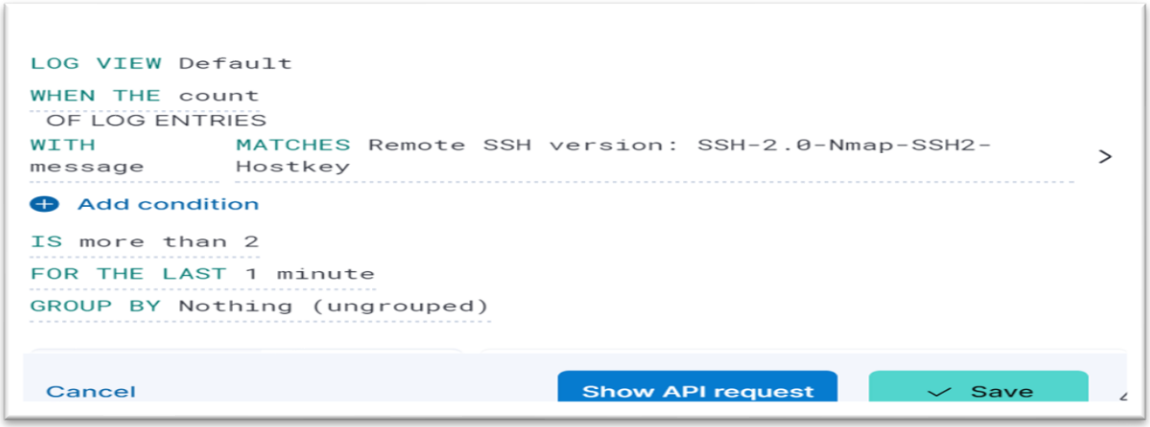
Verifying logs in SIEM AND Alerts:

In a Security Information and Event Management (SIEM) system using the ELK Stack (Elasticsearch, Logstash, and Kibana), collects and analyzes logs from various sources like Cowrie honeypot, Server logs that helps to detect and respond to attacks, such as port scans, SSH brute-force attempts and Dos attacks. By continuously monitoring these logs, the ELK-based SIEM can generate alerts for suspicious activities and helps track attacker's actions for further analysis.

In a SIEM system built using the ELK Stack (Elasticsearch, Logstash, and Kibana), alerts play a crucial role in identifying potential security incidents. When predefined rules or thresholds are met (e.g., abnormal login attempts, unusual network traffic), ELK generates alerts to notify the attack.

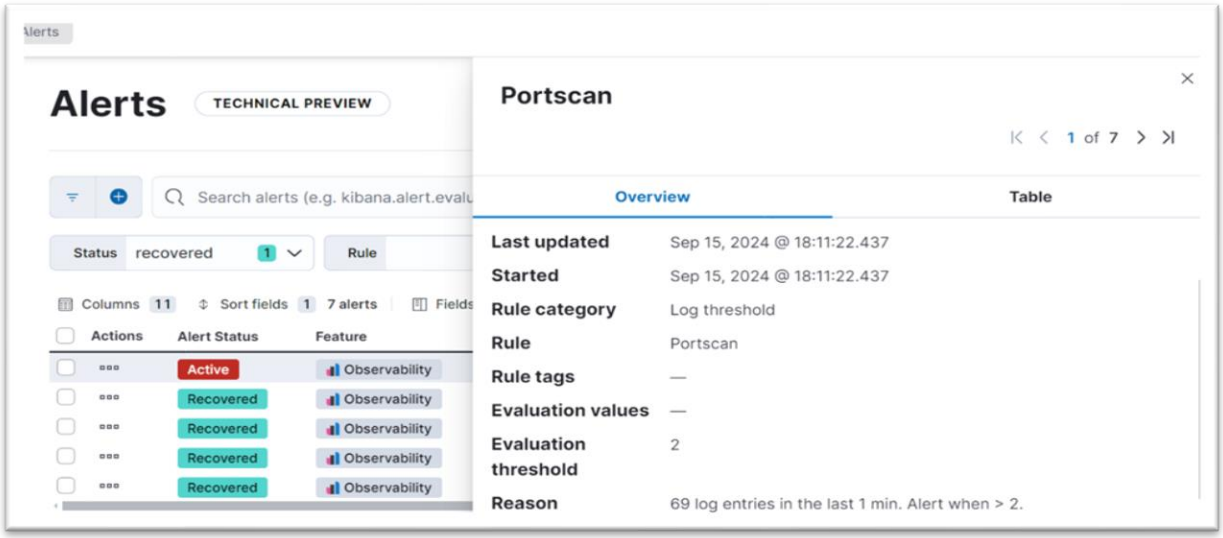
Port Scan using Nmap:

Below shows an Alert rule for the Port scan:



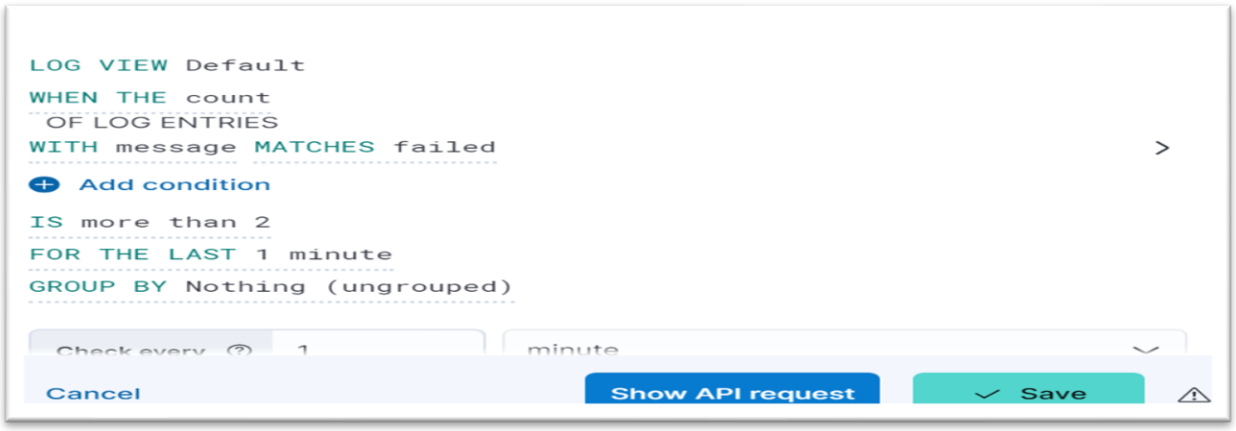
This rule tracks port scanning activities by monitoring for log entries where the message matches 'SSH-2.0-Nmap-SSH2', indicating that Nmap is being used to scan SSH ports. The rule is triggered when this message appears more than 2 times within the last 1 minute. Once this condition is met, an alert is activated, notifying that someone is attempting to perform a port scan."

Below shows an Alert is Activated:



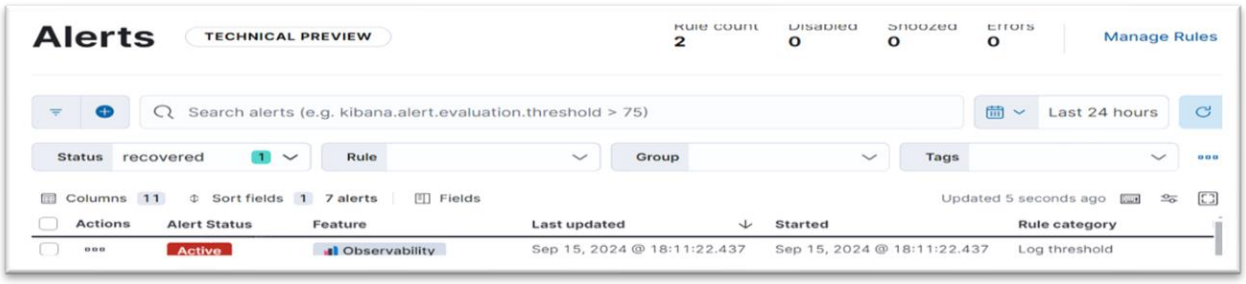
Brute force Attack:

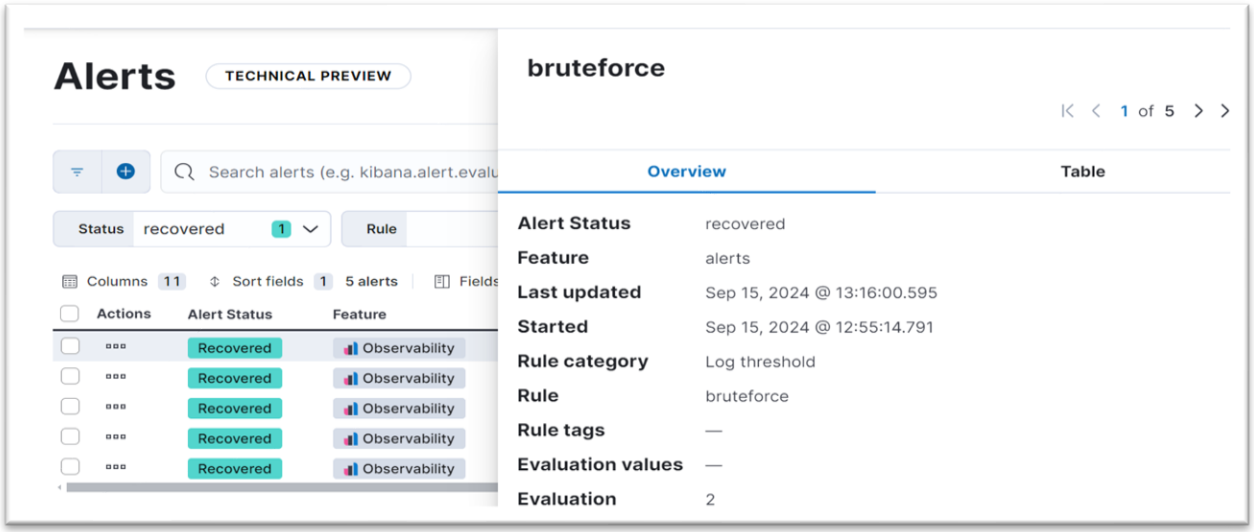
Below shows an Alert rule for the brute force attack:



This rule detects failed actions, such as failed login attempts, by monitoring the logs for occurrences of the word "failed." If more than 2 failures are logged within 1 minute, the rule triggers an alert, helping to identify possible brute-force attacks or other suspicious activities.

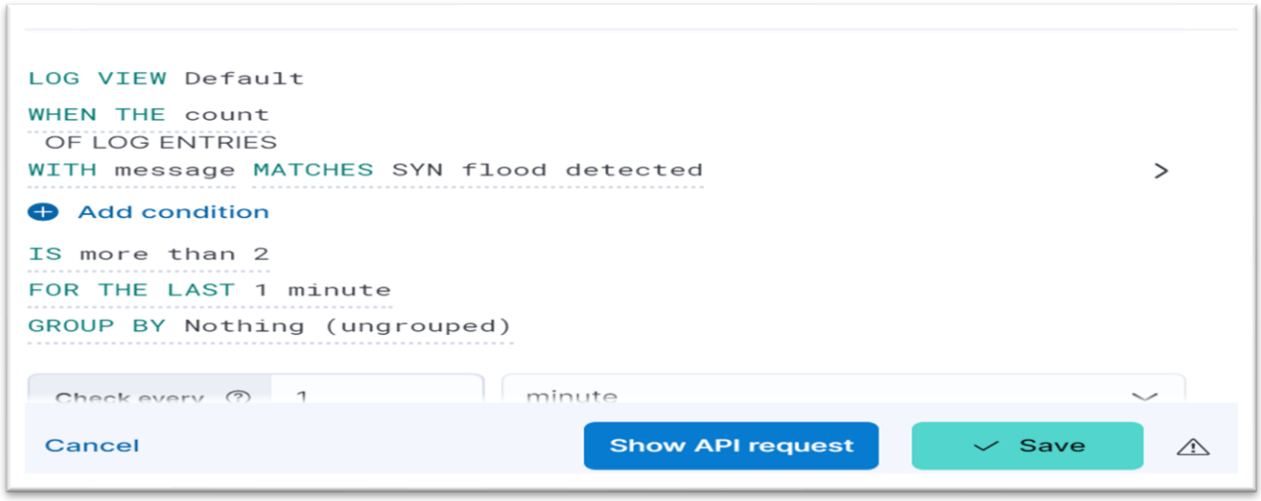
Below shows an Alert is Activated:





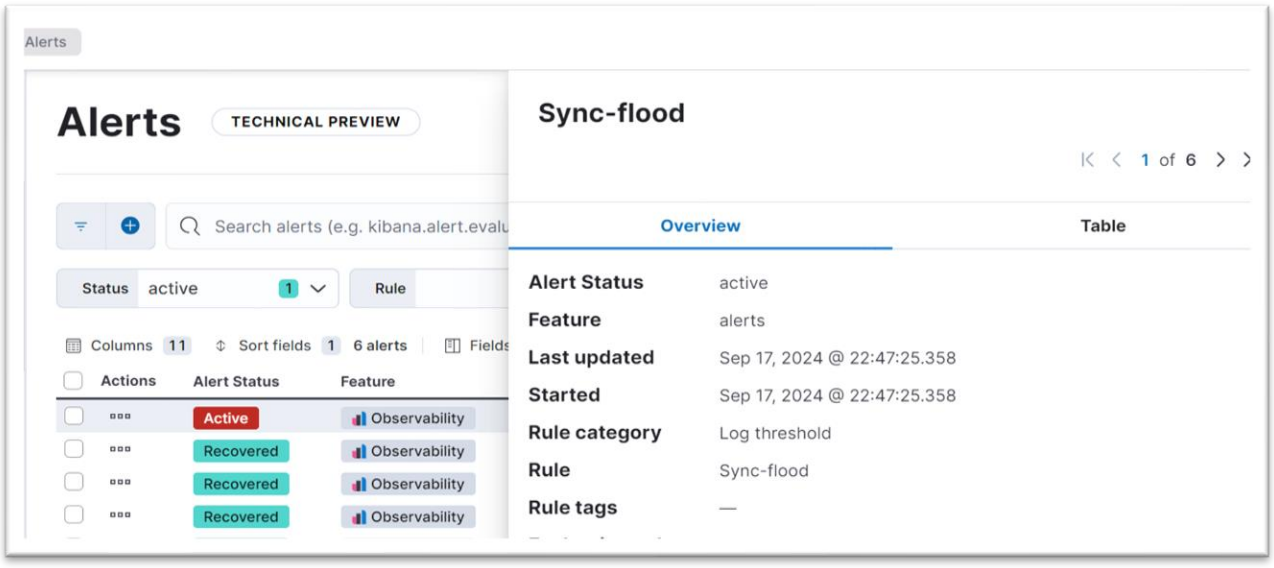
DOS ATTACK:

Below shows an Alert rule for Dos Attack:

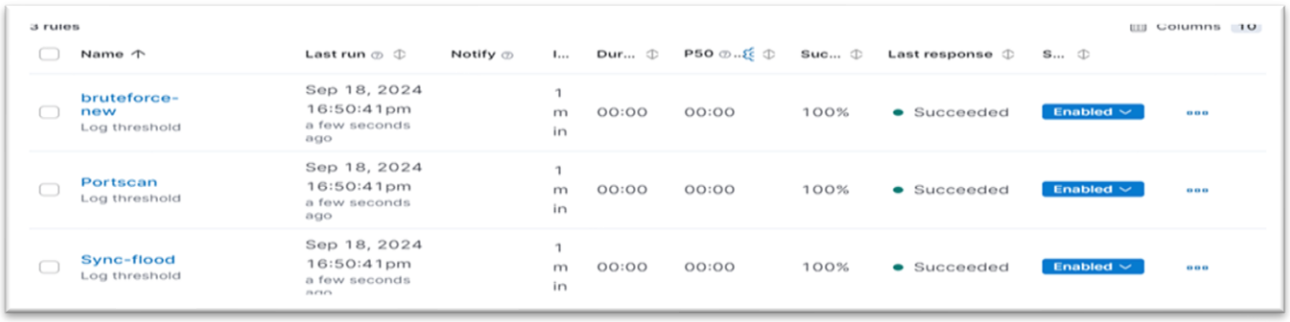


This rule is created to detect a SYN flood attack, a type of DoS attack where multiple SYN requests are sent to overwhelm a target. The rule will trigger an alert if more than 2 instances of log entries where the message matches 'SYN flood detected' are found within a 1-minute window. This helps ensure to notified quickly about SYN flood attacks, in order to take immediate action.

Below shows an Alert is Activated:



Below screenshot shows a rule has been "Enabled" for each type of attacks on the ELK:



Each rule runs every 1 minute and checks logs based on set conditions. The "Succeeded" status indicates that the rule successfully Executed and checked the logs, while the "Enabled" status means the rule is active and will continue monitoring. The 100% success shows that all conditions were met and the rule is functioning as expected.

CONCLUSION:

The project showcases a practical approach to network security using a virtual environment with Cowrie, the ELK Stack, and Kali Linux. I created scripts that allow me to select and launch different types of cyber-attacks, helping me better understand how these attacks work. By testing these attacks, I learned how a system responds, how logs are collected, and the alerts play in detecting the attacks. This hands-on experience strengthened my understanding of network vulnerabilities and reinforced the importance of security measures in protecting digital systems.