1. Monitoring Network Security Events

To monitor network security events, we use **Graylog**, a SIEM tool that collects, analyzes, and visualizes logs from multiple sources. Our focus is on tracking unusual traffic patterns that might indicate a **DoS** attack targeting the **HTTP**service of a web server (`192.168.1.100`).

Graylog Setup:

1. **Data Sources**:
    - Syslog from web servers, routers, and firewalls.
    - Application logs from the web server (e.g., Apache/Nginx).
    - Firewall logs for dropped packets.
2. **Streams and Alerts**:
    - **Stream**: "Web Traffic"
    - **Search Query**: Identifying spikes in HTTP traffic (e.g., more than 1000 requests from a single IP within 10 seconds).
    - **Alert Condition**: If the number of HTTP requests from a source exceeds 1000 within 10 seconds, trigger an alert.

Log Data Simulation:

We simulate logs generated during the **DoS attack**. In this case, the attacker is sending a large number of HTTP requests to the web server, which could eventually lead to service disruption.

---

2. Simulated Security Incident: Denial of Service (DoS) Attack

Incident Details:

- **Attack Type**: **HTTP Flood DoS**.
- **Target**: Web server at IP `192.168.1.100`.
- **Attack Source**: A botnet with multiple IPs, originating from the external network.
- **Indicators**: Excessive HTTP requests from multiple source IPs.

Attack Simulation (Mock Logs):

In this scenario, we simulate that an attacker uses a botnet to send excessive **HTTP GET requests** to the web server, aiming to overwhelm the system and cause a denial of service.

Web Server Logs (Apache Access Logs): The server logs multiple requests for the same resource in a short time frame.
less
Copy

```
Jan 29 10:03:10 server apache2[12345]: 203.0.113.5 - -
[29/Jan/2025:10:03:10 +0000] "GET /index.html HTTP/1.1" 200 1326
Jan 29 10:03:12 server apache2[12346]: 203.0.113.5 - -
[29/Jan/2025:10:03:12 +0000] "GET /index.html HTTP/1.1" 200 1326
Jan 29 10:03:14 server apache2[12347]: 203.0.113.5 - -
[29/Jan/2025:10:03:14 +0000] "GET /index.html HTTP/1.1" 200 1326
Jan 29 10:03:15 server apache2[12348]: 203.0.113.5 - -
[29/Jan/2025:10:03:15 +0000] "GET /index.html HTTP/1.1" 200 1326
Jan 29 10:03:20 server apache2[12349]: 203.0.113.5 - -
[29/Jan/2025:10:03:20 +0000] "GET /index.html HTTP/1.1" 200 1326
Jan 29 10:03:25 server apache2[12350]: 203.0.113.5 - -
[29/Jan/2025:10:03:25 +0000] "GET /index.html HTTP/1.1" 200 1326
```

-
    - The **same IP address** (`203.0.113.5`) is making multiple **HTTP GET requests** to the `/index.html` page.
    - These requests are coming in **every 2–5 seconds**, indicating an automated attack.

Firewall Logs (UFW): The firewall detects and logs unusual traffic patterns, including an abnormal amount of requests to port 80 (HTTP).
less
Copy

```
Jan 29 10:04:00 server ufw[4567]: [DENIED] IN=eth0 OUT=
MAC=00:1a:2b:3c:4d:5e:6f:7g:8h:9i:10j: SRC=203.0.113.5
DST=192.168.1.100 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=54321 DF
PROTO=TCP SPT=45678 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
Jan 29 10:04:02 server ufw[4568]: [DENIED] IN=eth0 OUT=
MAC=00:1a:2b:3c:4d:5e:6f:7g:8h:9i:10j: SRC=203.0.113.5
DST=192.168.1.100 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=54322 DF
PROTO=TCP SPT=45679 DPT=80 WINDOW=29200 RES=0x00 SYN URGP=0
```

- These log entries show that the firewall is blocking excessive connections from `203.0.113.5` to port 80 on the web server (`192.168.1.100`).

---

3. Incident Response and Mitigation

Once the attack is detected, we follow a structured **incident response process**:

Step 1: Detection of the Attack

- The **Graylog alert** was triggered by the unusual HTTP request pattern. Specifically, we identified a **spike in requests** from IP `203.0.113.5`.
- The **web server logs** show **multiple requests to the same resource** (`/index.html`) from the same IP within a short time frame.
- **Firewall logs** confirm that the server is receiving a flood of requests, and the firewall is starting to block them.

Step 2: Containment and Mitigation

Block the Attacker's IP: Using **UFW (Uncomplicated Firewall)**, we block the IP `203.0.113.5` to prevent further requests from that source.

bash

Copy

```
sudo ufw deny from 203.0.113.5 to any port 80
```

-

Deploy Rate Limiting: On the web server, we apply **rate limiting** to prevent such attacks in the future. For example, using **mod_evasive** (Apache), we configure it to block IPs that exceed a set number of requests per second.

Example configuration for **mod_evasive**:

bash

Copy

```
# /etc/apache2/mods-enabled/evasive.conf
<IfModule mod_evasive20.c>
    DOSHashTableSize 3097
    DOSPageCount 10
    DOSSiteCount 150
    DOSPageInterval 1
    DOSSiteInterval 1
    DOSBlockingPeriod 10
</IfModule>
```

-

Step 3: Verification

- After blocking the malicious IP and applying rate limiting, we continue to monitor the **Graylog dashboards** to ensure that no further attacks are occurring.
- **System checks** are performed to ensure that no files were modified or deleted during the attack.

Step 4: Recovery

- The web server continues to serve traffic normally, and the attacker's IP is blocked at the firewall level.
- **Post-Incident Actions**:
    - **Review security configurations** and harden the web server further.
    - **Update the firewall rules** to block any new suspicious IP addresses in the future.
    - **Conduct a full security audit** to ensure no compromise occurred.

---

4. Logs and Screenshots for Evidence

Graylog Alert Screenshot:

In Graylog, we triggered an alert that shows a large number of HTTP requests from `203.0.113.5` within a short period.

- **Alert Screenshot**: This screenshot shows the Graylog alert details for the DoS attack detection. It includes the source IP, destination IP, and the number of requests within the specified threshold.

Firewall Log Screenshot:

Here's a screenshot of the **firewall log** showing that `203.0.113.5` is being blocked for excessive HTTP requests:

less
Copy

```
Jan 29 10:04:00 server ufw[4567]: [DENIED] IN=eth0 OUT=
MAC=00:1a:2b:3c:4d:5e:6f:7g
```