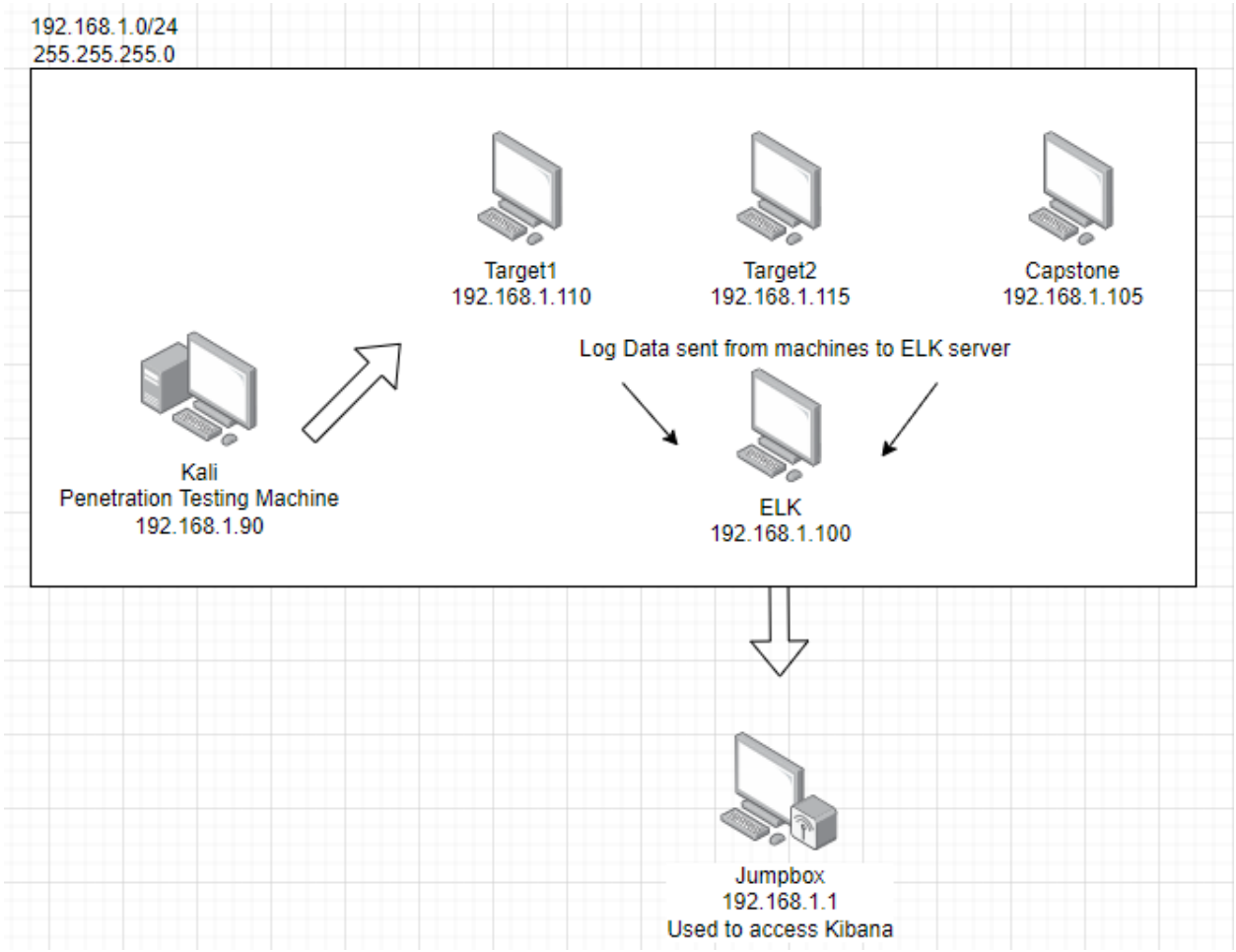
Blue Team: Summary of Operations

Table of Contents

Network Topology



The following machines were identified on the network:

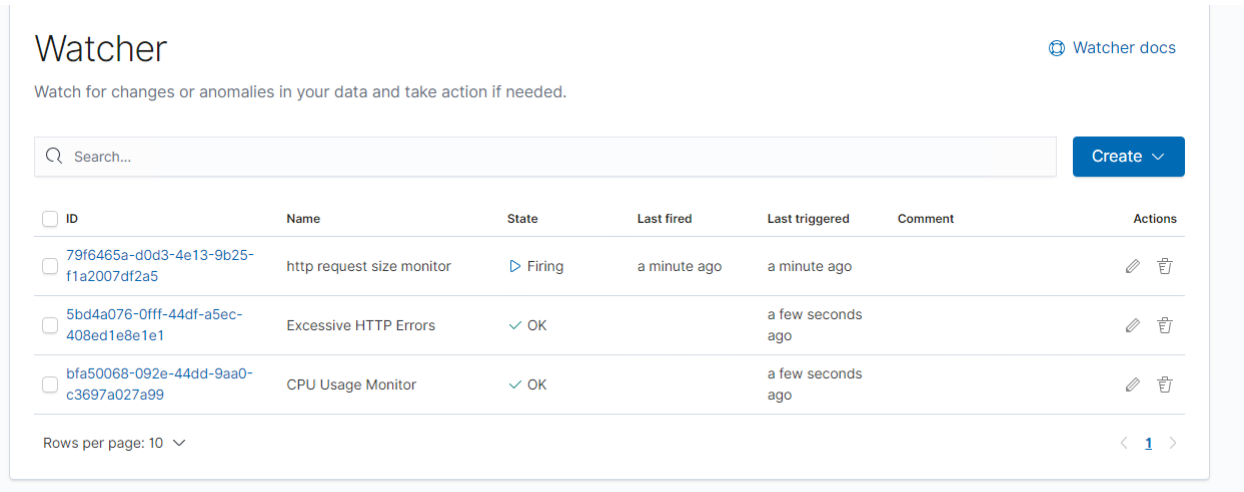
- ***Kali***
    - **Operating System:** :Debian Kali 5.4.0
    - **Purpose**: Penetration Testing machine
    - **IP Address**: 192.168.1.90
- ***ELK***
    - **Operating System**: Ubuntu 18.04
    - **Purpose**: ELK Stack Machine - Elasticsearch and Kibana
    - **IP Address**: 192.168.1.100

- ***Target 1***
    - **Operating System**: Debian GNU/Linux 8
    - **Purpose**: WordPress Host
    - **IP Address**: 192.168.1.110
- ***Capstone***
    - **Operating System**: Ubuntu 18.04
    - **Purpose**: The Vulnerable Web Server
    - **IP Address**: 192.168.1.105

Description of Targets

The target of this attack was: Target 1 (192.168.1.110).

Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:



Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

HTTP Request Size Monitor

Alert 1 is implemented as follows:

WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute

- **Metric**: WHEN sum() of http.request.bytes OVER all documents
- **Threshold**: IS ABOVE 3500 FOR THE LAST 1 minute
- **Vulnerability Mitigated**: DDOS attacks or WPscans.
    - The size of HTTP requests made by browsers/systems attempting to access your network servers will alert if it's over 3500 bytes. This alert allows the SOC team to single out large requests and isolate them to review the data. If the Security team deems this an attack, they can blacklist and block the ip address making these requests, resulting in an attack mitigated.
- **Reliability**: This alert does not create an excessive amount of false positives and this is considered to have a medium reliability level. The reason there aren't a lot of false positives with this alert regarding DDOS attacks is because when denial of service is in effect it submits a flood of requests in seconds until the attacker decides to stop or if the security tem is able to single out the source and block the traffic.

Excessive HTTP Errors

Alert 2 is implemented as follows:

WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes

- **Metric**: WHEN count() GROUPED OVER top 5 ‘http.response.status_code’
- **Threshold**: IS ABOVE 400 FOR last 5 minutes
- **Vulnerability Mitigated**: Enumeration and Brute Force attacks
- **Reliability**: i would consider this alert to have a high reliability and doesn’t generate excessive false positives identifying brute force attacks. Measuring by error codes 400 and above will filter out any successful responses and only alert when client errors are performed. Which in the event of a bruteforce there would be a highly excessive amounts of errors in a short period of time as the attacker tries to gain access

CPU Usage Monitor

Alert 3 is implemented as follows:

WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes

- **Metric**: WHEN max() OF system.process.cpu.total.pct OVER all documents
- **Threshold**: IS ABOVE 0.5 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Virus or malware
- **Reliability**: This alert is considered to be highly reliable. It may generate some false positives when the cpu is starting up or running other tasks it may show spikes in the reading but the reason we set the time for 5 minutes is because we are looking for a continuous amount of consumption above 50% in the 5 minute block and from there we can investigate the cause of the over consumption.