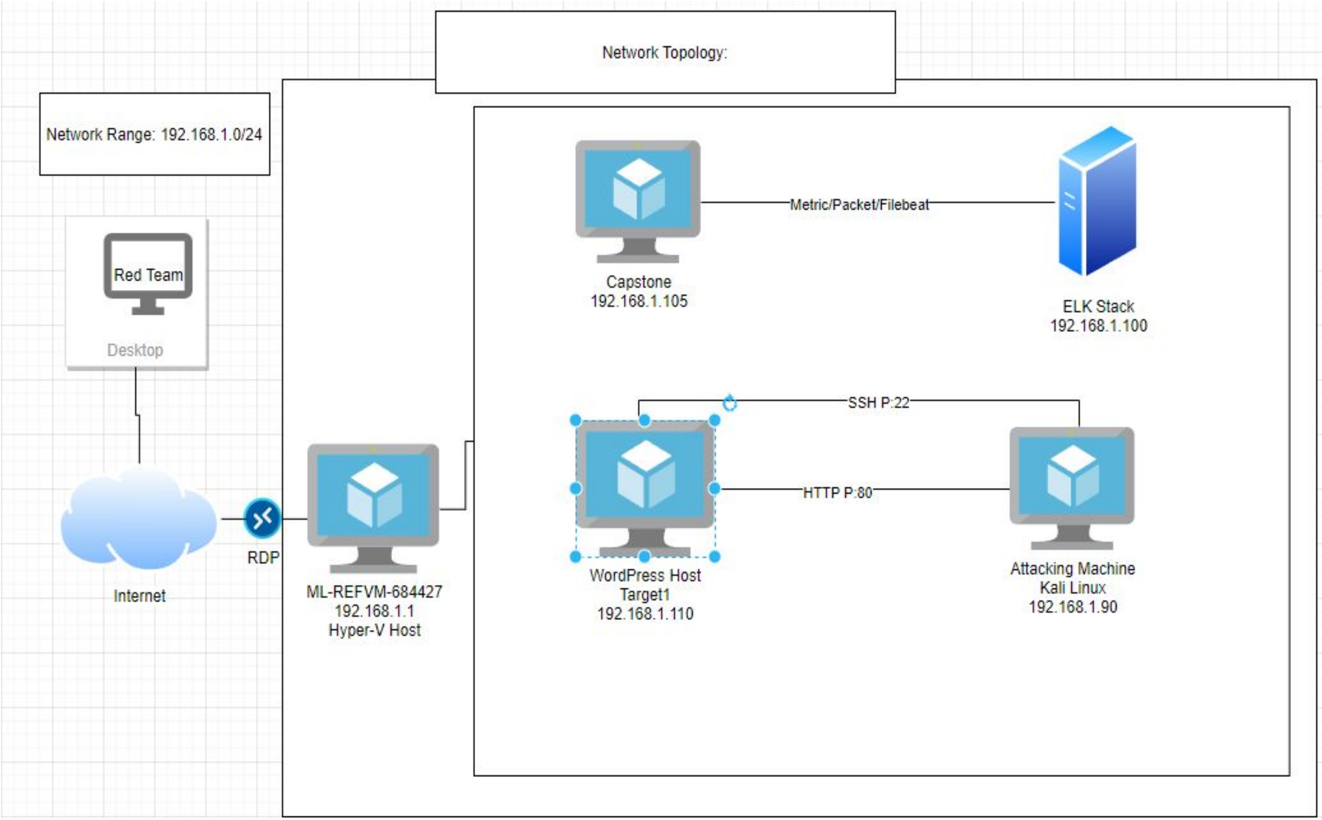
Blue Team

Summary of Operations

Table of Contents

1. Network Topology
2. Description of Targets
3. Monitoring the Targets
4. Patterns of Traffic & Behavior
5. Suggestions for Going Further

Network Topology:



Network:
Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines:
Kali:
IPv4: 192.168.1.90
OS: Debian Kali 5.4.0
Purpose: Attack Machine

Capstone:
IPv4: 192.168.1.105
OS: Linux
Purpose: Vulnerable Web Server

Target1:
IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Purpose: Wordpress Host

ELK:
IPv4: 192.168.1.100
OS: Ubuntu 18.04
Purpose: Elasticsearch and Kibana Stack

Description of Targets:

Target 1

VM: Target 1

IP address : 192.168.1.110

Alerts have been implemented as followed.

Functions as an Apache web server.

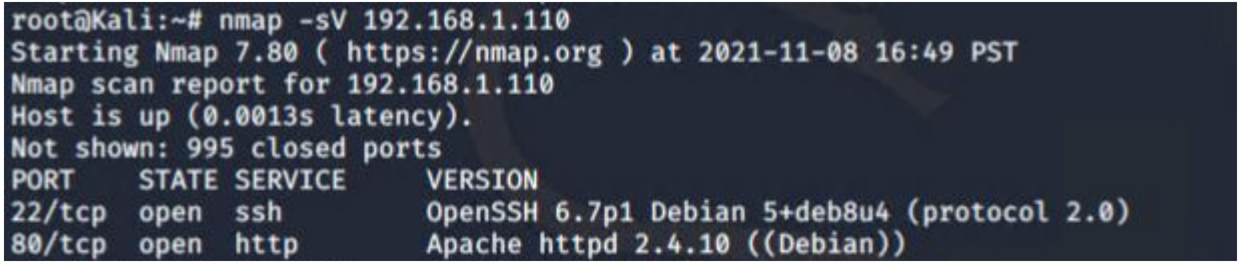
Monitoring the Targets

A service nmap scan identified the following services as entry points:

Target 1:

Port 22/TCP Open SSH 6.7p1 Debian 5+deb8u4

Port 80/TCP Open HTTP Apache httpd 2.4.10 (Debian)



```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-08 16:49 PST
Nmap scan report for 192.168.1.110
Host is up (0.0013s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
```

Patterns of Traffic & Behavior

Alert 1

Excessive HTTP Errors - Metric: WHEN count() GROUPED OVER top 5 ‘http.response.status_code’ - Threshold - IS ABOVE 400 - Vulnerability Mitigated: Enumeration & Brute Force - Reliability: High reliability, because measuring 400 codes and above will keep the normal/successful responses out of the alert. Client and server errors typically bring more concern, and this will track if they are happening at a high rate.

Alert 2

HTTP Request Size Monitor

Alert 3

CPU Usage Monitor

Suggestions for Going Further

Patch: Updating WordPress

Make sure to regularly update WP to ensure a higher level of security

Things like WordPress Core, PHP versions, and any plugins.

Maybe install a security plugin like Wordfence.

Disable unused WP features (WP XML-RPC, REST API)

Do not allow /wp-admin and /wp-login.php to be publicly accessible.

Updating regularly is the best way to get patches/fixes to vulnerabilities and exploits.

REST API is utilized by wpscan to enumerate users, etc. Disabling it would help mitigate.