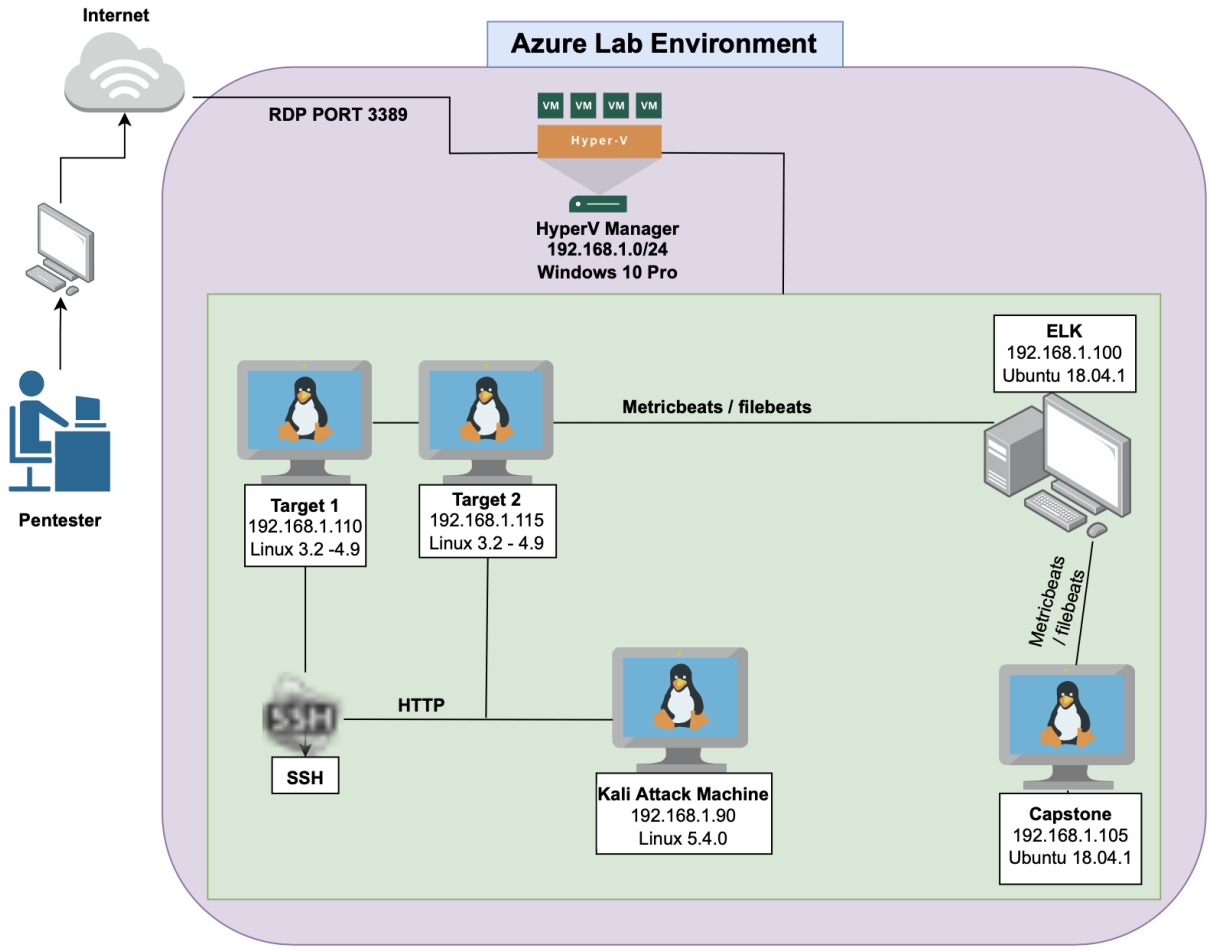
Blue Team: Summary of Operations

Table of Contents

Network Topology (Diagram)



Briana Norman

The following machines were identified on the network:

Kali

- Operating System:
  - Debian Kali 5.4.0
- Purpose:
  - The Penetration Tester / Attacking Machine
- IP Address:
  - 192.168.1.90

Hypervisor / HyperV Manager Host (ML-REFVM-684)

- Operating System:
  - Microsoft Windows / Microsoft 10 Pro
- Purpose:
  - Hypervisor/Gateway aka Host Machine with HyperV Manager
- IP Address:
  - 192.168.1.1

ELK

- Operating System:
  - Ubuntu 18.04 LTS
- Purpose:
  - The ELK (Elasticsearch, Logstash and Kibana) Stack
- IP Address:
  - 192.168.1.100

Target 1

- Operating System:
  - Debian GNU/Linux 8 ***(Also with Aggressive Scans they are Linux 3.x | 4.x and Linux 3.2 - 4.9))***
- Purpose:
  - Initial Victim Machine (also used as the WordPress Host)
- IP Address:
  - 192.168.1.110

Target 2

- Operating System:
  - Debian GNU/Linux 8 ***(Also with Aggressive Scans they are Linux 3.x | 4.x and Linux 3.2 - 4.9))***
- Purpose:
  - Secondary Victim Machine
- IP Address:
  - 192.168.1.115

Capstone

- Operating System:
  - Ubuntu 18.04 LTS
- Purpose:
  - Helps the ELK machine with filebeats and metricbeats

- IP Address:
  - 192.168.1.105

Screenshot Output:

```
Currently scanning: 172.16.140.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 210
-----------------------------------------------------------------------------
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.1     00:15:5d:00:04:0d      1      42  Microsoft Corporation
192.168.1.100   4c:eb:42:d2:d5:d7      1      42  Intel Corporate
192.168.1.105   00:15:5d:00:04:0f      1      42  Microsoft Corporation
192.168.1.110   00:15:5d:00:04:10      1      42  Microsoft Corporation
192.168.1.115   00:15:5d:00:04:11      1      42  Microsoft Corporation
```

```
root@Kali:~# nmap -sV 192.168.1.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-05 15:06 PDT
Nmap scan report for 192.168.1.1
Host is up (0.0014s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00072s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.
4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00045s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.110
Host is up (0.0015s 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))
111/tcp  open  rpcbind     2-4 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.115
Host is up (0.00068s 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))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.90
Host is up (0.0000080s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 28.90 seconds
root@Kali:~#
::1             ff02::2         ip6-allrouters  ip6-loopback    localhost
ff02::1         ip6-allnodes    ip6-localhost   Kali
root@Kali:~#
```

Description of Targets

- Target of attack on the network is: **Target 1 (192.168.1.110)**
- Target 1 has an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers.

Monitoring the Targets

This scan identifies the services below as potential entry points :

Target 1

- Port 22/TCP Open SSH OpenSSH 6.7p1 Debian 5+deb8u4
- Port 80/TCP Open HTTP Apache httpd 2.4.10 (Debian)

```
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
Not shown: 65529 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))
111/tcp   open  rpcbind     2-4 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
60598/tcp open  status      1 (RPC #100024)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

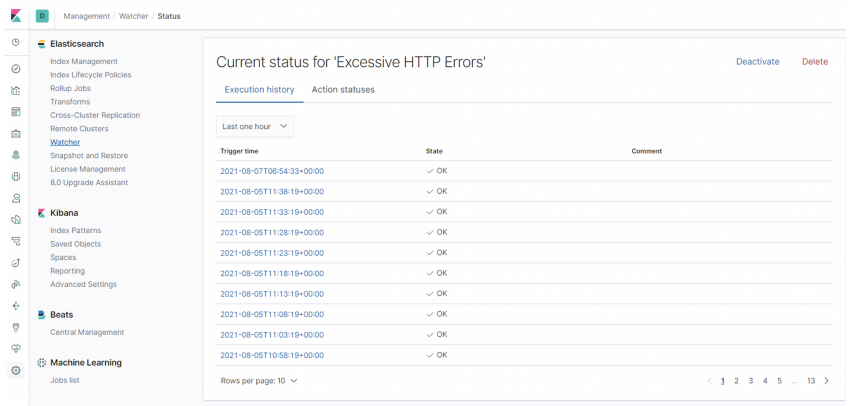
Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

Excessive HTTP Errors

Excessive HTTP Errors 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
- **Vulnerability Mitigated:** Enumeration/Brute Force
- **Reliability**: The alert is highly reliable. Measuring by abnormal HTTP error codes 400 and above will filter out any normal or successful responses. 400+ codes are client and server errors which are of more concern. Especially when taking into account these error codes going off at a high rate. Abnormal HTTP website error codes that high are typically indications of Brute Force or DDoS (Denial of Service) attacks.

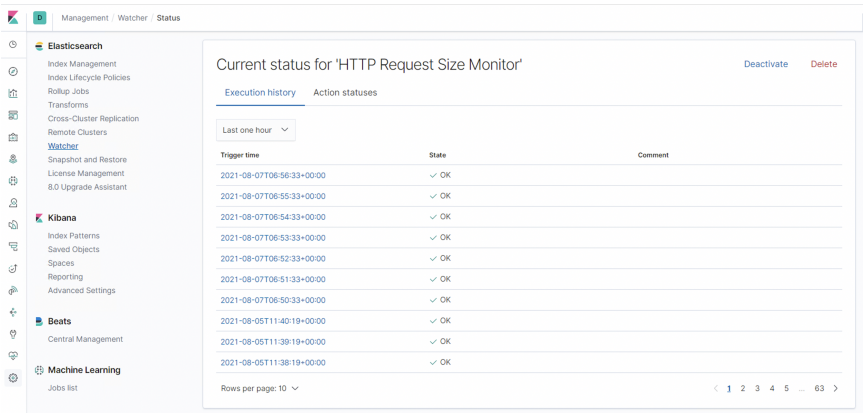


HTTP Request Size Monitor

HTTP Request Size Monitor 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 / *Total bytes monitored and sent for cumulative data within the last 1 minutes*
- **Threshold:** IS ABOVE 3500
- **Vulnerability Mitigated:** Code injection in HTTP requests (XSS and CRLF) or DDOS
- **Reliability:** *Medium to low:* Alert could create false positives. It comes in at medium to low reliability. There is a possibility for a large non malicious HTTP request or legitimate HTTP traffic.
    - *For my notes only: Reminder that typical usage of web pages can often contain more than 3500 bytes of data due to the amount of text, font and images used.*

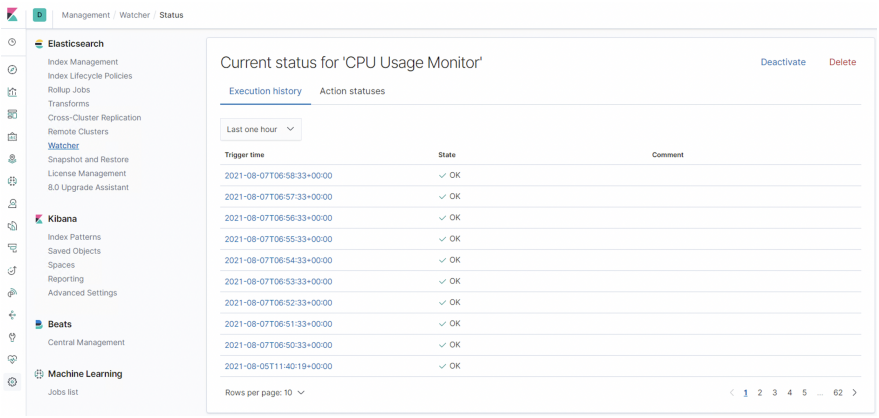


CPU Usage Monitor

CPU Usage Monitor 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:** When CPU IS ABOVE 0.5
- **Vulnerability Mitigated:** Malicious software, programs (malware or viruses) running taking up resources so it deters this CPU utilization processes
- **Reliability:** HIGH. The alert is highly reliable. Even if there isn't a malicious program running this can still help determine where to improve on CPU usage (anything over 0.5%).



Suggestions for Going Further

Each alert above pertains to a specific vulnerability/exploit. Recall that alerts only detect malicious behavior, but do not stop it. For each vulnerability/exploit identified by the alerts above, suggest a patch. E.g., implementing a blocklist is an effective tactic against brute-force attacks. It is not necessary to explain how to implement each patch.

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats. In addition to watching for occurrences of such threats, the network should be hardened against them. The Blue Team suggests that IT implement the fixes below to protect the network:

Excessive HTTP Errors

Patch: ***WordPress Hardening*** *- change to latest security patches*

Command: *sudo apt-update ; sudo apt-install WordPress*

*MY NOTES ONLY***

- *The command to find a software package (I know you didn't ask this, but it's important) is: → apt-cache search <keyword> - while will return a list of packages and their descriptions. Then, you can use: → sudo apt-get install <packages> to install desired packages. To find and apply needed updates (you called it "update all the software packages"), First update the package version database with: → sudo apt-get update . Only after this finishes will the packaging system know which installed packages have more recent versions available. Then, to apply the package upgrades: → sudo apt-get upgrade . Once the upgrade finishes look for files called → /var/run/reboot*. If these file exist, they'll contain hints about why your system needs to be rebooted. For example, I remember seeing the package name linux-base in → /var/run/reboot*, when I'd upgraded linux-base and needed to reboot. If there is NO /var/run/reboot* file, no reboot is needed.*
    - Implement regular updates to WordPress
        - WordPress Core, PHP version, Plugins
    - Install security plugin(s) = (Wordfence which adds security functionality)
    - Disable unused WordPress features and settings like; WordPress XML-RPC (on by default) and WordPress REST API (on by default)
    - Block requests to /?author= by configuring web server settings
    - Remove WordPress logins from being publicly accessible specifically: /wp-admin and /wp-login.php

→ ***Why It Works*:**

- Regular updates to WordPress, the PHP version and plugins is an easy way to implement patches or fixes to exploits/vulnerabilities.
- Depending on the WordPress security plugin it can provide things like: Malware scans; Firewall; IP options (to monitor/block suspicious traffic)
- REST API is used by WPScan to enumerate users; Disabling it will help mitigate WPScan or enumeration in general
- XML-RPC uses HTTP as it's method of data transport
- WordPress links (permalinks) can include authors (users); Blocking request to view the all authors (users) helps mitigate against user enumeration attacks
- Removal of public access to WordPress login helps reduce the attack surface

HTTP Request Size Monitor

- **Patch:** ***Code Injection/DDoS Hardening.*** Implementing on the web server the HTTP Request Limits and Implementation of input validation on forms; *(Limits can include a number of things: the maximum URL Length; the maximum length of a query string; the maximum size of a request)*

→ ***Why It Works*:** *If an HTTP request URL length, query string and over size limit of the request (which is up to the specific amount chosen), then a 404 range of errors will occur. → (This will help reject these requests that are too large). —- Input validation can also help protect against malicious data anyone attempts to send to the server via the website or application in/across a HTTP request.*

CPU Usage Monitor

- **Patch:** ***Install Virus or Malware Hardening / IDS or HIDS Systems/ Add or update to a good antivirus/ Implement and configure Host Based Intrusion Detection System (HIDS) → (Ex. SNORT/HIDS)***

→ ***Why It Works*:** ***Antiviruses specialize in removal, detection and overall prevention of malicious threats against computers. → (Any modern antivirus usually covers more than viruses and is a more robust solution to protecting a computer in general). Another is, IDS and HIDS. Where it monitors and analyzes internals of computing systems. → (They also monitor and analyze network packets).***