# **Blue Team: Summary of Operations**

## **Table of Contents**

* Network Topology
* Description of Targets
* Monitoring the Targets
* Patterns of Traffic & Behavior
* Suggestions for Going Further

### **Network Topology**

The following machines were identified on the network:

* Name of VM 1 Final Lab
  + **Operating System**: Windows 10
  + **Purpose**: Hosts all of the below machines, and is used to access kibana
  + **IP Address**: 192.168.1.1
* Name of VM 2 Kali
  + **Operating System**: Kali linux
  + **Purpose**: Kali serves as the attacking machine to the 2 targets
  + **IP Address**: 192.168.1.90
* Name of VM 3 Capstone
  + **Operating System**: Linux ubuntu 18.04
  + **Purpose**: Hosts the company server
  + **IP Address**: 192.168.1.105
* Name of VM 4 Elk
  + **Operating System**: Linux ubuntu 18.04
  + **Purpose**: Reports metric and packet data to kibana which can be accessed on the Windows machine
  + **IP Address**: 192.168.1.100
* Name of VM 5 Target 1
  + **Operating System**: Linux 3.2 - 4.9
  + **Purpose**: Target Machine
  + **IP Address**: 192.168.1.110
* Name of VM 6 Target 2
  + **Operating System**: Linux 3.2 - 4.9
  + **Purpose**: Target Machine
  + **IP Address**: 192.168.1.115

### **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:

#### **Excessive HTTP Errors**

Excessive HTTP Error monitor is implemented as follows:

* **Metric**: http.response.status\_code
* **Threshold**: Above 400
* **Vulnerability Mitigated**: Monitor for a high amount of HTTP error codes which could be a sign of attack
* **Reliability**: This monitor is quite reliable and does not produce many false positives or negatives

#### **HTTP Request Size Monitor**

Http Request Size Monitor is implemented as follows:

* **Metric**: http.request.bytes
* **Threshold**: Above 3500
* **Vulnerability Mitigated**: HTTP requests above 3500 bytes would be deemed anomalous and can be investigated
* **Reliability**: This monitor recorded a few false positives but when we were penetration testing the monitor was firing

#### **CPU Usage Monitor**

CPU usage monitor is implemented as follows:

* **Metric**: system.process.cpu.total.pct
* **Threshold**: Above 0.5
* **Vulnerability Mitigated**: Monitor for packet flood attacks and overall server health
* **Reliability**: This monitor is quite unreliable and fires a lot. The threshold should be increased for future usage

### **Suggestions for Going Further (Optional)**

* 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, identified by the alerts above. 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:

* Vulnerability 1 Enumeration through wordpress
  + **Patch**: Wordpress enumeration can be patched with a plugin called wp hardening
  + **Why It Works**: WP hardening is a first party plugin from wordpress to patch up multiple vulnerabilities with the website
* Vulnerability 2 More complex passwords
  + **Patch**: Force users Michael and Steven to have more complex passwords
  + **Why It Works**: We were able to guess Michaels password easily (It was his name) and Stevens password was instantly found in a wordlist. Having more complex passwords would make it much harder to gain access to any account that has higher access.
* Vulnerability 3 Block off SSH from unknown and malicious ip addresses
  + **Patch**: Introduce a whitelist so only certain ip addresses are able to access each account
  + **Why It Works**: Blocking ssh from outside sources will only allow known ip addresses to ssh into each machine. Multiple ports were left open on each target machine so they should be prevented.