# **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**

*TODO: Fill out the information below.*

The following machines were identified on the network:

* Target 1
  + **Operating System**: Debian GNU/Linux8/v3.16.0-6
  + **Purpose**: WordPress Host
  + **IP Address**:192.168.1.110/24
* Capstone
  + **Operating System**: Ubuntu 18.04
  + **Purpose**: Vulnerable Server
  + **IP Address**: 192.168.1.105/24
* Target 2
  + **Operating System**: Debian GNU/Linux8/v3.16.0-6
  + **Purpose**: WordPress Host
  + **IP Address**: 192.168.1.115/24
* ELK
  + **Operating System:** Ubuntu 18.04
  + **Purpose:** Elasticsearch & Kibana Stack(filebeat, packetbeat, metricbeat)
  + **IP Address:** 192.168.1.100
* Kali
  + **Operating System:** Kali Release 2020.1 / Kernel: Linux 5.4.0
  + **Purpose:** Attacking Machine
  + **IP Address: 192.168.1.90**

### **Description of Targets**

The Targets of this Lab was the Target 1 VM (192.168.1.110) and the subsequent machine Target 2 VM (192.168.1.115).

These two VMs were pre configured WordPress management servers Written in PHP and run MySql databases for they’re operations. Both Servers are made to be Vulnerable with weak security configurations.

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 Rejects**

Alert 1 is implemented as follows:

* **Metric**: WHEN count() GROUPED OVER top 5 ‘http.response.status\_code’
* **Threshold**: Is above 400
* **Vulnerability Mitigated**: Enumeration attempts on WordPress
* **Reliability**: This Rule is high reliability in its ability to filter out specific error codes and display the count. This shows when somebody has failed the http request on a server. If the count is high then could be very significant.

**Attempts to Pull Files/Infiltration**

Alert 2 is implemented as follows:

* **Metric**: WHEN sum() of http.request.bytes OVER all documents
* **Threshold**: Is above 3500
* **Vulnerability Mitigated**: Excessive Traffic Size could Indicate Infiltration DOs or
* **Reliability**: This is a pretty reliable but can trigger false positives depending on the sizes of the documents being requested. However continuous triggers and high volume sizes remains significant of intrusion.

#### **CPU Usage Monitor**

Alert 3 is implemented as follows:

* **Metric**: WHEN max() OF system.process.cpu.total.pct OVER all documents
* **Threshold**: Is above 0.5
* **Vulnerability Mitigated**: Password Cracks and Malware Events
* **Reliability**: This is highly reliable and a significant measurement outside of attacks high cpu usage should be mitigated however also is indicative of malware.

**Suggestions for Going Further (Optional)**

* Each alert above pertains to a specific vulnerability/exploit.
* 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:
* HTTP Error Overload
  + **Patches:** Consistent updates on the Wordpress software including the language used whether sql or PHP and the various plugins.
  + Blocking logins from the public, admin accounts shouldn't be easily accessible.
  + IP blocks and firewalls on the server side to filter traffic
  + **Why It Works**: Updating the system regularly will naturally help mitigate problems and vulnerabilities that come with dated systems.
  + Blocks on logins helps mitigate any bruteforce attempts directly from the unauthorized, Whitelisting IPs could coincide with this whether allowing certain ranges or specific users.
  + Firewalls are always a good mitigation technique with unwanted traffic however an expensive fix but reducing the attack surface should always be at the forefront of security goals.
* Pulling Files and Infiltration
  + **Patch**: Request limits on the client side with server documents
  + Input validation in sensitive areas and Maintenance of account privileges.
  + **Why It Works**: A 404 will come up when the request size is too large data analysis can help find a proper limit for this rule. This will help block many malicious exfiltration attempts
  + Input validation limits code injections on the server through http interactions.
* **Cpu Usage and Malware Injection**
  + **Patch**: Hardening in Malware and Viruses
  + IPS and IDS implementations
  + **Why It Works**: Antivirus protections are a good start for detecting and cleaning malware with the help of IPS and IDS; they can be quarantined and removed from the server from network traffic.