# **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: Kali
  + **Operating System**: Linux
  + **Purpose**: Used to attach other machines
  + **IP Address**: 192.168.1.90
* Name of VM 2: Capstone
  + **Operating System**: Linux
  + **Purpose**: vulnerable target VM for testing alerts
  + **IP Address**: 192.168.1.105
* Name of VM 1: ELK
  + **Operating System**: Linux
  + **Purpose**: holds the Kibana dashboards
  + **IP Address**: 192.168.1.100
* Name of VM 1: Target1
  + **Operating System**: Linux
  + **Purpose**: Exposes a vulnerable WordPress server, sends logs to ELK
  + **IP Address**: 192.168.1.110
* Name of VM 1: Target2
  + **Operating System**: Linux
  + **Purpose**: a more difficult WordPress target. Should be ignored unless all other portions of this project are completed. Sends logs to ELK.
  + **IP Address**: 192.168.1.115

### **Description of Targets**

The target of this attack was: Target 1, IP address of 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**

Alert 1 is implemented as follows:

* **Metric**: for packetbeat, when count() grouped over top 5 ‘http.response.status\_code’
* **Threshold**: above 400 for the last 5 minutes
* **Vulnerability Mitigated**: catches all HTTP error codes starting at 401 and above
* **Reliability**: High reliability at identifying brute force attacks, should not be too many false positives or negatives

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

Alert 2 is implemented as follows:

* **Metric**: using packetbeat, when the sum of http.request.bytes over all documents is...
* **Threshold**: above 3500 for the last 1 minute
* **Vulnerability Mitigated**: control amount of HTTP request sizes
* **Reliability**: medium, should not generate too many false positives/negatives

#### **CPU Usage Monitor**

Alert 3 is implemented as follows:

* **Metric**: using metricbeat, when the max of system.process.cpu.total.pct over all documents is...
* **Threshold**: above 0.5 for last 5 minutes
* **Vulnerability Mitigated**: memory dump
* **Reliability**: high reliability, simply reporting when CPU usage is over 50%

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

*TODO*:

* 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
  + **Patch**: TODO: E.g., *install special-security-package with apt-get*
  + **Why It Works**: TODO: E.g., *special-security-package scans the system for viruses every day*
* Vulnerability 2
  + **Patch**: TODO: E.g., *install special-security-package with apt-get*
  + **Why It Works**: TODO: E.g., *special-security-package scans the system for viruses every day*
* Vulnerability 3
  + **Patch**: TODO: E.g., *install special-security-package with apt-get*
  + **Why It Works**: TODO: E.g., *special-security-package scans the system for viruses every day*