# **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: **Target 1** 
  + **Operating System**: Debian GNU/Linux 8 (Jessie)
  + **Purpose**: Raven Security Web server
  + **IP Address**: 192.168.1.110
* Name of VM 2: Target 2
  + **Operating System**: Debian GNU/Linux 8 (Jessie)
  + **Purpose**: Web Server
  + **IP Address**: 192.168.1.115
* Name of VM 3: Elk Stack
  + **Operating System**: Ubuntu 18.04.4 LTS
  + **Purpose**: Kibana log server
  + **IP Address**: 192.168.1.100
* Name of VM 4: Kali
  + **Operating System**: Kali 2020.1
  + **Purpose**: Attacker machine
  + **IP Address**: 192.168.1.90
* Name of VM 5: Capstone
  + **Operating System**: Ubuntu 18.04.4 LTS
  + **Purpose**: Machine for alert testing
  + **IP Address**: 192.168.1.105
* Name of VM 6: Hyper-V Host
  + **Operating System**: Windows 10 Version 1909
  + **Purpose**: Hyper-V Host/Default Gateway
  + **IP Address**: 192.168.1.1

### 

### **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**

Alert 1 is implemented as follows:

* **Metric**: HTTP Errors / Packetbeat
* **Threshold**: Response code above 400 (bad requests/page not found) for the past five minutes
* **Vulnerability Mitigated**: Brute force attack
* **Reliability**: High reliability. Will certainly indicate the presence of a brute force attack in progress.

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

Alert 2 is implemented as follows:

* **Metric**: HTTP Request Bytes / Packetbeat
* **Threshold**: Total bytes over 3,500 in the past minute
* **Vulnerability Mitigated**: Brute force, buffer overflow, DDOS attack, XSS - indicates presence of an attack
* **Reliability**: Medium/High reliability. Depending on the normal flow of traffic to this site, with legitimate user requests, this could alert occasional false positives. Establishing a baseline of traffic flow would increase the reliability of this alert.

#### **CPU Usage (%) Monitor**

Alert 3 is implemented as follows:

* **Metric**: CPU Usage (%) Monitor / Metricbeat
* **Threshold**: Process CPU total when above 0.5 (50%) in the last five minutes
* **Vulnerability Mitigated**: Denial of Service Attack (DDOS)
* **Reliability**:High - Given the services dedicated to this server, typically a high-reliability alert metric.

### 

### **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:

* **Weak Passwords** 
  + **Patch**: Implement organizational training and policy requiring complex passwords and multi-factor authentication
  + **Why It Works**: Makes brute-forcing more difficult, and failed login attempts easier to detect
* **Unsalted Hash** 
  + **Patch**: Add prepended or appended salt to each password before hashing
  + **Why It Works**: Adding a salt makes the same password hash into a completely different string every time. The salt does not need to be secret. Just by randomizing the hashes, lookup tables, reverse lookup tables, and rainbow tables become ineffective. (*https://crackstation.net/hashing-security.htm)*
* **Privilege Escalation**
  + **Patch**: Remove python script running privileges for User:Steven
  + **Why It Works**: This was the only discoverable privilege escalation vulnerability in User:Steven(‘s) account. Removing the python ability should prevent Steven, or any hacker in his account, from escaping to root.