## **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:

* Machines
* IPv4: 192.168.1.110
* OS: Linux
* Hostname: Raven 1
* IPv4: 192.168.1.115
* OS: Linux
* Hostname: Raven 2
* IPv4: 192.168.1.100
* OS: Ubuntu
* Hostname: ELK Stack
* IPv4: 192.108.1.105
* OS: Ubuntu
* Hostname: Capstone
* IPv4: 192.168.1.90
* OS: LInux
* Hostname: Kali

### **Description of Targets**

*TODO: Answer the questions below.*

Two VMs on the network were vulnerable to attack: Target 1 192.168.1.110 and Target 2 192.168.1.115.

Each VM functions as 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:

#### **Monitoring wpscans against the network**

*TODO: Replace Alert 1 with the name of the alert.*

Alert 1 is implemented as follows:

* **Metric**: monitor user.agent.original
* **Threshold**: 1
* **Vulnerability Mitigated**: Mitigate the damage of a wpscan by seeing who is doing the scan
* **Reliability**: High reliability because no-one on your network should be running a wp scan against your own network

#### **Detecting of traffic to wordpress directory**

Alert 2 is implemented as follows:

* **Metric**: Alert set up to detect any processes working inside the wordpress directory
* **Threshold**: Any Processes inside that are !=1
* **Vulnerability Mitigated**: Any access to the wp-config.php file
* **Reliability**: Its medium reliability as you can get a lot of false positives if someone is simply working in the directory. However it makes sure that all the information about it is being logged.

#### **Traffic through SSH over port 22**

Alert 3 is implemented as follows:

* **Metric**: Lookings at traffic from unrecognized IPS attempting access over port 22
* **Threshold**: Requests going more 5 within 10 minutes
* **Vulnerability Mitigated**: Mitigates the amount of brute force attempts that can be made against the server
* **Reliability**: Medium reliability as sometimes people can just have bad times remembering passwords or attackers can have the information before hand and be let in with no issues

*TODO Note: Explain at least 3 alerts. Add more if time allows.*

### **Suggestions for Going Further**

*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**: Update WordPress to the latest and secure it behind a server
  + **Why It Works**: They won't be able to scan or access the network because it's hidden behind another server.
* Vulnerability 2
  + **Patch**: Force complicated passwords.
  + **Why It Works**: We were able to guess Michael's password but wouldn't be able to if he had it something much more complex.
* Vulnerability 3
  + **Patch**: Block connections from IPs outside of the server range.
  + **Why It Works**: By making it so that connections cannot come from sources outside the network so that hackers cannot enter the server from outside IPS. This would only allow people to work from the office but it’s the safest bet.