**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**

**Diagram

Description automatically generated**

The following machines were identified on the network:

* Kali
  + **Operating System**: Kali GNU/Linux Rolling 8
  + **Purpose**: Penetration
  + **IP Address**: 192.168.1.90
* Target 1
  + **Operating System**: Debian GNU/Linux
  + **Purpose**: Wordpress Server
  + **IP Address**: 192.168.1.110
* ELK
  + Operating System: Ubuntu 18.04.4
  + Purpose: Kibana Dashboards
  + IP Address: 192.168.1.100
* Capstone
  + Operating System: Ubuntu 18.04.1
  + Purpose: Vulnerable Server
  + IP Address: 192.168.1.105

**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 Errors alerts are implemented as follows:

* **Metric**:
  + WHEN count() GROUPED OVER top 5 ‘http.response.status\_code’ IS ABOVE 400 FOR THE LIST 5 minutes
* **Threshold**:
  + ABOVE 400
* **Vulnerability Mitigated**:
  + Brute force, Enumeration
* **Reliability**:
  + The alert is highly reliable. Setting alerts for error codes at 400 leave enough room for normal traffic to proceed without tripping false alarms.

Chart, histogram

Description automatically generated

**HTTP Request Size Monitor:**

HTTP Request Size Monitor alerts are implemented as follows:

* **Metric**:
  + WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
* **Threshold**:
  + ABOVE 35**00**
* **Vulnerability Mitigated**:
  + Code injection into HTTP requests
* **Reliability**:
  + The alert was prone to false positive alerts. The reliability was marginal at best.

Graphical user interface, text

Description automatically generated with medium confidenceChart, line chart

Description automatically generated

**CPU Usage Monitor**

CPU Usage Monitor alert is implemented as follows:

* **Metric**:
  + WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
* **Threshold**:
  + ABOVE 0.5
* **Vulnerability Mitigated**:
  + Malicious programs/software using excessive resources
* **Reliability**:
  + The alert was reliable and performed as desired. It provided accurate and gave a clear indication when a large amount of the CPU was being used.

Graphical user interface, table

Description automatically generated with medium confidenceGraphical user interface, chart

Description automatically generated