**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***.*

A screenshot of a computer

Description automatically generated with medium confidence

***Diagram above as used in our group presentation***

The following machines were identified on the network:

* Capstone
  + **Operating System**: Ubuntu 18.04
  + **Purpose**: This is a vulnerable web server
  + **IP Address**: 192.168.1.105
* ELK
  + **Operating System**: Ubuntu 18.04
  + **Purpose**: The ELK Stack holds the Kibana and Elasticsearch. It is where all the logging are stored
  + **IP Address**: 192.168.1.100
* KALI
  + **Operating System**: Kali Linux (Debian kali 5.0)
  + **Purpose**: It is the attacking machine where all the penetration testing occurred.
  + **IP Address**: 192.168.1.90
* TARGET 1
  + **Operating System**: Debian GNU /Linux 8
  + **Purpose**: This is the vulnerable machine that was attacked and compromised
  + **IP Address**: 192.168.1.110
* TARGET 2
  + **Operating System**: Debian GNU /Linux 8
  + **Purpose**: Another vulnerable machine in the network
  + **IP Address**: 192.168.1.115

**Description of Targets**

There are two vulnerable vms in this network namely Target 1 and Target 2

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:

**ALERST:**

**Excessive HTTP Errors**

This is implemented as follows:

*WHEN count() GROUPED OVER top 5 ‘http.response.status\_code IS ABOVE 400 FOR THE LAST 5 minutes*

* **Metric**: WHEN count() GROUPED OVER top 5 ‘http.response.status\_code
* **Threshold**: IS ABOVE 400
* **Vulnerability Mitigated**: Brute Force /Enumeration
* **Reliability**: Does this alert generate lots of false positives/false negatives? Rate as low, medium, or high reliability.
  + This is highly reliable alert. Error codes above greater or equal to 400 points to the server side and will definitely reflect when it occurs

Graphical user interface, text, email

Description automatically generated

**HTTP Request Size Monitor**

Implemented as follows:

*WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute*

* **Metric**: WHEN sum() of http.request.bytes OVER all documents
* **Threshold**: IS ABOVE 3500
* **Vulnerability Mitigated**: SQL code injection, Distributed Denial Of Service, and Cross-site scripting attacks
* **Reliability**: Does this alert generate lots of false positives/false negatives? Rate as low, medium, or high reliability.
  + It is possible to have false positives because there could be authentic or legitimate http requests that could be above the size of the threshold but this should not be at high sides. I would say medium rate of reliability.
* A screenshot of a computer

  Description automatically generated

**CPU Request Size Monitor**

Implemented as follows:

*WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes*

* **Metric**: *WHEN max() OF system.process.cpu.total.pct OVER all documents*
* **Threshold**: *IS ABOVE 0.5*
* **Vulnerability Mitigated**: viruses, malwares and any program that’s runs and takes up more CPU resources
* **Reliability**: TODO: Does this alert generate lots of false positives/false negatives? Rate as low, medium, or high reliability.
* Graphical user interface, text, table

  Description automatically generated with medium confidence

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

**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 : HTTP ERRORS
  + **Patch**:
    - **Hardening WordPress**
      * Regular updates should be made with apt-get
      * Security plugins like wordfence could be installed
      * Unused features should be disabled to minimize open doors for attack
      * Wordpress admin logins should be removed from public access
  + **Why It Works**:
    - Regular updates will patch some of the exploits/vulnerabilities
    - Security plugs can scan for malicious codes.
      * It can also provide firewall for tto block harmful traffic
    - Removal of admin logins from public access reduces attack surface
    - Disabling unused features reduces attack surface.
* Vulnerability 2: REQUEST SIZE MONITOR
  + **Patch**:
    - Distributed denial of service (DDOS), code injection and cross site scripting should all be hardened
      * HTTP request limit on the web server should be set with regards to the length of querying string, and request size.
      * Input validated should be implemented.
  + **Why It Works**:
    - When the limit set is reached, errors will occur thereby creating rejection of the request
    - Input validation prevents malicious attacks from non-human agents.

* Vulnerability 3: CPU USAGE MONITOR
  + **Patch**: Malware /virus hardening
    - Strong antivirus programs should be installed and updated
    - Host based Intrusion Detection System (HIDS) can be installed
  + **Why It Works**:
    - Strong antiviruses scan and remove all malicious codes that are usurping the system resources
    - HIDS monitors and alerts of any malicious traffic into the system