Blue Team: Summary of Operations

By: Eren Yargici

Table of Contents

Network Topology

The following machines were identified on the network:

- Hypervisor
  - **Operating System**: Microsoft Windows
  - **Purpose**: Hypervisor / Gateway
  - **IP Address**: 192.168.1.1
- ELK
  - **Operating System**: Linux Ubuntu
  - **Purpose**: Elasticsearch, Logstash, Kibana Server
  - **IP Address**: 192.168.1.100
- Capstone
  - **Operating System**: Linux
  - **Purpose**: HTTP Server
  - **IP Address**: 192.168.1.105
- Target 1
  - **Operating System**: Linux Debian 3.16.57
  - **Purpose**: HTTP Server / WordPress
  - **IP Address**: 192.168.1.110
- Target 2
  - **Operating System**: Linux Debian 5.14.13
  - **Purpose**: HTTP Server

- ○ **IP Address**: 192.168.1.115

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.response.status_code > 400
- **Threshold**: 5 hits in 5 minutes
- **Vulnerability Mitigated**: This alert allows the security team to identify potential attacks and block the IP, change passwords, and closely filter sensitive ports such as 22.
- **Reliability**: This alert is not expected to generate false positives. This alert is reliable in identifying brute force attacks.

HTTP Request Size Monitor

Alert 2 is implemented as follows:

- **Metric**: http.request.bytes
- **Threshold**: 3500 hits in 1 minute
- **Vulnerability Mitigated**: Controlling the number of HTTP request size through a filter in order to protect against distributed denial of service attacks.
- **Reliability**: This alert does not generate false positives.

CPU Usage Monitoring

Alert 3 is implemented as follows:

- **Metric**: system.process.cpu.total.pct
- **Threshold**: 0.5 usage in 5 minutes

- **Vulnerability Mitigated**: Controlling the CPU usage can be useful to spot malware or unoptimized malicious code executed.
- **Reliability**: This alert can generate false positives because of CPU spikes unrelated to an attack. (Example: applications can cause CPU usage to be more than 50%)

Suggestions for Further Vulnerability Mitigation

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 - Excessive HTTP Errors
    - **Patch**: Require a strong password policy for each user on the network. Configuring Windows Group Policies to have a strong password policy for accounts.
    - **Why It Works**: Having a strong password will make it harder to guess or brute force.

- Vulnerability 2 - HTTP Request Size Monitor
    - **Patch**: Use of modern intrusion prevention and threat management systems that include firewalls, VPN's, content filtering, and load balancing. These operate together to provide continual and consistent network security, preventing denial of service attacks. This encompasses everything from spotting any network traffic irregularities to stopping the attack with utmost accuracy.
    - **Why It Works**: Monitoring request sizes of HTTP packets can minimize denial of service threats.

- Vulnerability 3 - CPU Usage Monitor
    - **Patch**: Using Host Intrusion Prevention System to identify potential denial of service attacks.
    - **Why It Works**: This preventive measure can alert and stop malware by monitoring processing behavior.