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:

- Host
- Operating System: Windows
- Purpose: The host machine
- IP Address: 192.168.1.1

- ELK
- Operating System: Ubuntu
- Purpose: Elasticsearch and Kibana
- IP Address: 192.168.1.100

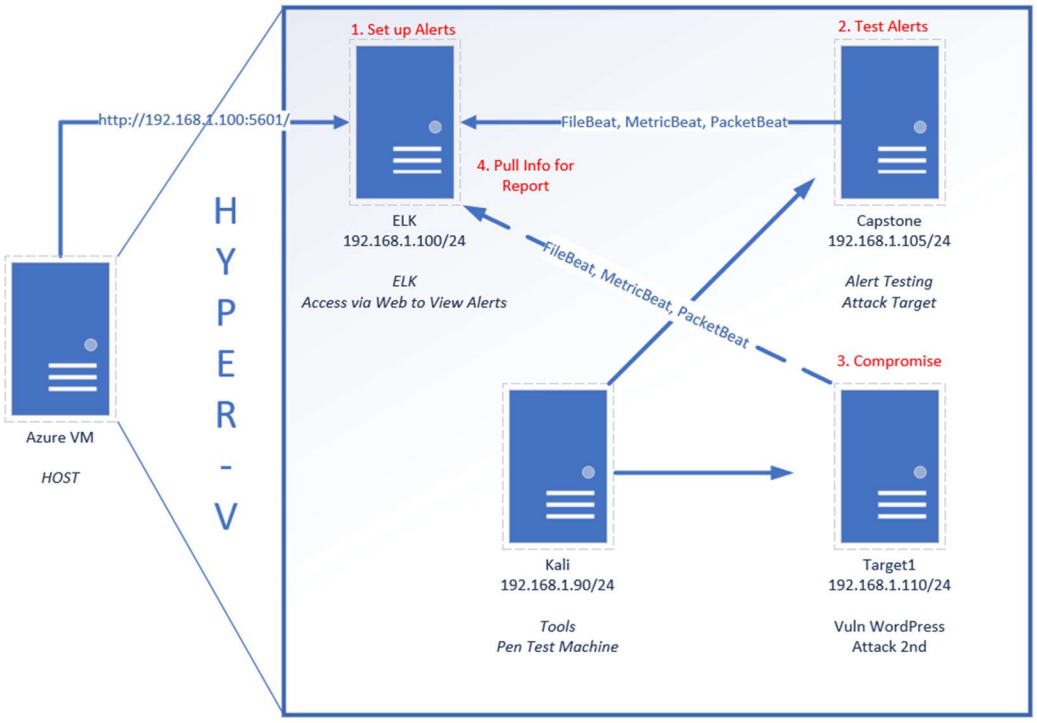
- Kali
- Operating System: Linux
- Purpose: Kali machine used for the penetration test
- IP Address: 192.168.1.90

- Capstone
- Operating System: Ubuntu
- Purpose: Vulnerable web server for Filebeat and Metricbeat
- IP Address: 192.168.1.105

- Target 1
- Operating System: Linux
- Purpose: WordPress server
- IP Address: 192.168.1.110

- Target 2 (optional)
- Operating System:
- Purpose:
- IP Address: 192.168.1.115

Network Diagram:



Description of Targets

The target of this attack was: Target 1 IP Address: 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:

CPU Usage Monitor

CPU Usage Monitor Alert is implemented as follows:

- **Metric**: WHEN max() OF system.process.cpu.total.pct OVER all documents
- **Threshold**: IS ABOVE 0.5 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Scanning for malicious software, programs running and taking up resources
- **Reliability**: High. Along with alerting to malicious attacks it can help improve CPU usage

Excessive HTTP Errors

Excessive HTTP Errors is implemented as follows:

- **Metric**: WHEN count() GROUPED OVER top 5 'http.response.status.code'
- **Threshold**: IS ABOVE 400 FOR THE LAST 5 minutes
- **Vulnerability Mitigated**: Enumeration/Brute Force
- **Reliability**: High. The error codes for 400+ are client and server error and filters out any normal or successful responses

HTTP Request Size Monitor

HTTP Request Size Monitor is implemented as follows:

- **Metric**: WHEN sum() OF http.request.bytes OVER all documents
- **Threshold**: IS ABOVE 3500 FOR THE LAST 1 minute
- **Vulnerability Mitigated**: Scans on the website for code injections in HTTP requests
- **Reliability**: Medium. The alert could create false positives.

Suggestions for Going Further (Optional)

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:

HTTP Request Size Monitor

- **Patch**: DDOS Hardening
    - Implement HTTP Request Limit on the web server
        - Limits include Maximum URL Length, maximum length of a contextual string, and maximum size of a request (to name a few)
    - Implement input validation on forms
    - Limit sysadmins sudo privileges on python
- **Why It Works**:
    - Error Alert 404 occurs when a HTTP request exceeds maximum URL length, maximum length contextual string and/or maximum size request
        - Assists in rejecting HTTP request that are too large
    - Forms with input validation protects against malicious data attempts to send to the web server
    - Limiting the number of users with sudo privileges will lessen the attack size as fewer users will have “root” privileges

Excessive HTTP Errors

- **Patch**: WordPress Hardening
    - Implement regular updates to ensure most current version of WordPress
        - Include PHP and Plugins
    - Disable unused WordPress features and settings
        - Include WordPress XML-RPC and REST API (both default settings)
    - Block requests to /?author=<number> by configuring the web server settings
    - Remove WordPress credentials from being accessed publicly
        - Include /wp-admin and /wp-login.php
    - Enforce an appropriate password policy
        - Include minimum and maximum length
        - Restrictions against password reuse
        - Restrictions against using common passwords
        - Restrictions against using contextual string in the password (e.g., user id, app name)

- **Why It Works**:
    - Regular updates to WordPress (PHP and plugins) is easily implemented and patches/fixes vulnerabilities
    - Disabling XML-RPC eliminates the option of an attacker using HTTP to transport data. Disabling REST API helps to mitigate by reducing the attack surface
    - Block requests to view all authors/users mitigates against user enumeration attacks
    - Removal of public access to WordPress login reduces the attack surface
    - Weak Passwords
    - Input validation will assist in protecting against malicious data attempts being sent over the web server or as a HTTP request

CPU Usage Monitor

- **Patch**: Malware Hardening
    - Implement or update antivirus software
    - Implement a Host Based Intrusion Detection System
- **Why It Works**:
    - Antivirus software specifically designed to prevent malicious attacks
        - Include removal, detection, and prevention on all computers
    - Host Based Intrusion Detection Systems monitor and provide analysis of internal systems
        - Include network packet monitoring and analysis