# Blue Team: Summary of Operations

## Table of Contents

### Network Topology

The following machines were identified on the network:

- **Target 1**
  - Operating System: Windows 6.1 (Samba 4.2.14 Debian)
  - Purpose: vulnerable wordpress server
  - IP Address: 192.168.1.110
- **Target 2**
  - Operating System: Linux 2.6.32
  - Purpose: vulnerable wordpress server
  - IP Address: 192.168.1.115
- **Kali**
  - Operating System: Debian(Linux)
  - Purpose: Attacking machine
  - IP Address: 192.168.1.90

-**ELK**
  - Operating System: Ubuntu(Linux 2.0)
  - Purpose: Report machine used for log analysis
  - IP Address: 192.168.1.100

-**Capstone**
  - Operating System: Ubuntu(Linux 2.0)
  - Purpose: Vulnerable VM target for alerts testing
  - 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: Excessive HTTP Errors, HTTP Request Size Monitor, CPU Usage Monitor.

### Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

Excessive HTTP Errors is implemented as follows:

- Metric: packetbeat-*
- Threshold: WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- Vulnerability Mitigated: Brute Force Attack
- Reliability: We believe it has high reliability because It does not generate a lot of false positives and it only monitors errors, which indicate failed login attempts during a brute force attack. This amount of errors in this timeframe is highly unusual

HTTP Request Size Monitor is implemented as follows:

- Metric: packetbeat-*
- Threshold: WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Vulnerability Mitigated: Denial of Service
- Reliability: highly reliable due to the fact it will not generate a high amount of false negatives or positives. The amount of http request bytes is unusual and a false positive would only happen in the highly unlikely event a large file is transferred

CPU Usage Monitor is implemented as follows:

- Metric: metricbeat-*
- Threshold: WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Vulnerability Mitigated: Remote Code execution
- Reliability: This has a high reliability rate given that it won’t generate false positives and negatives. RCE runs processes that otherwise wouldn’t on a machine which results in high CPU usage. A 50% threshold is unlikely to generate false positives given it’s so high.

Suggestions for Going Further

Despite the fact that implementing an alert for Excessive HTTP Errors warns against brute forcing attacks, this alert doesn’t prevent them from occurring. However, there are several steps the organization can take to prevent brute force attacks from occurring including the implementation of:

- Two factor authentication
- A limit to the number of failed login attempts that can be made
- An Ip address blacklist
- The enforcement of password complexity standards

The alert set for HTTP Request Size Monitor successfully generates warnings of possible denial of service attempts. Actions an organization can take to further address and prevent denial of service attempts include the implementation of the following:

- A limit on the amount of requests that can be implemented in a given timeframe
- The creation of an IP whitelist for trusted addresses
- The denial of requests made outside usual service requests

The alert set for CPU Usage Monitor successfully warns about attempts at remote code execution. However, an organization should take further steps to prevent this type of attack such as:

- Setting a rule in place to prevent PHP files from being uploaded to the system
- Implementing input validation for all relevant fields

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:

Identified vulnerabilities and their mitigations

- **Denial of Service(smb-vuln-regsvc-dos CVE-2002-0724)**
  - Mitigation(s):

    install-microsoft-patch-8fac75d5adfe9cbd42b6443f5f745e3e
    install-microsoft-patch-b5cc325c2c4d5e544069f40ccfb503fa
    install-microsoft-patch-ea0114e104fd70b075f22ea5f0f31c82
    install-microsoft-patch-af149537a845cfdca1e6eb13bf520ac3

  - Why It Works: The patches listed above are updates from Microsoft that eliminate the buffer overflow vulnerability within Server Message Block(SMB) protocol that allows an attacker to execute a denial of service attack
- **Cross Site Request Forgery**
  - Mitigation(s):
    - Reauthentication
    - One-time tokens
    - Captcha implementation
  - Why It Works: The above methods deter malicious actors by taking extra steps to verify the authenticity of users