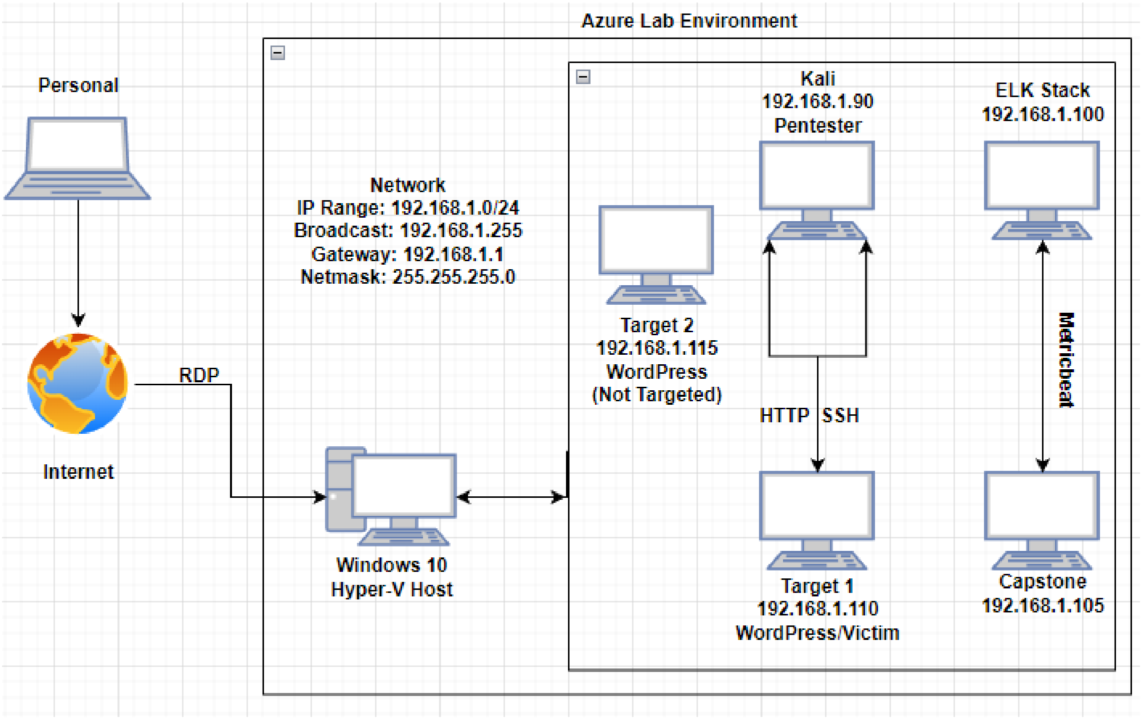
Blue Team: Summary of Operations

Table of Contents

Network Topology

The following machines were identified on the network:

- Kali
  - **Operating System**: Debian Kali 5.4.0
  - **Purpose**: Pentester (Attacker)
  - **IP Address**:192.168.1.90
- Target 1
  - **Operating System**: Windows 6.1 (Samba 4.2.14-Debian)
  - **Purpose**: MySQL Database Web Server
  - **IP Address**: 192.168.1.110
- ELK
  - **Operating System**: Ubuntu 18.04
  - **Purpose**: ELK Stack (Elasticsearch and Kibana)
  - **IP Address**: 192.168.1.100



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:

Alert 1: Excessive HTTP Errors

This alert is implemented as follows:

- **Metric**: http.response.body.bytes
- **Threshold**: Above 400 for the last 5 minutes.
- **Vulnerability Mitigated**: Identifying BruteForce Attacks.
- **Reliability**: High reliability, triggered multiple times during Brute Force attack.

Alert 2: HTTP Request Size Monitor

This alert is implemented as follows:

- **Metric**: http.request.bytes

- **Threshold**: Above 3500kb in the last minute.
- **Vulnerability Mitigated**: Identifying DOS Attacks.
- **Reliability**: Medium. Could create false positives for legitimate large http traffic.

Alert 3: CPU Usage Monitor

This alert is implemented as follows:

- **Metric**: system.process.cpu.total.pct
- **Threshold**: Above 0.5 for the last 5 minutes.
- **Vulnerability Mitigated**: Malicious software, program, malware, or virus taking up resources.
- **Reliability**: Medium. Relatively non-specific but a useful tool when used with other alerts. Can also help show where to improve on CPU with false positives.

Suggestions for Going Further

- 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: Excessive HTTP Errors

- **Patch:** SSHGuard: Monitors system logs to identify brute force attacks and blocks suspicious IPs

  E.g., install SSHGuard with apt-get install sshguard
- **Why it Works:** Identifies multiple failed attempts by monitoring logging activity and temporarily blocks suspicious IPs and puts them into iptables.

Vulnerability 2: Unreasonable HTTP Request Size

- **Patch**: Kernel Upgrade: Prevent DOS attacks that take advantage of Linux vulnerabilities by updating Linux Security patches and Kernel.

  E.g., install Upgrade kernel with apt-get upgrade linux-image-generic
- **Why It Works:** Keeps the OS up-to-date with latest security packages to prevent vulnerabilities from outdated software getting exploited.

Vulnerability 3: Exhausted CPU Resources.

- **Patch1:** Install Elastic Agent, Modern Endpoint Protection:

E.g., in kibana, add the Endpoint Security Agent.

- **Why It Works:** Blocks malware, ransomware, and advanced threats. Unifies prevention, detection, and response.

- **Patch2:** cpulimit: Set limits to processes to limit CPU usage.
  E.g., install cpulimit with apt-get install cpulimit

- **Why It Works:** When an alert triggers for CPU usage, Soc analyst is notified and in response uses the command *top* to detect which process is consuming excessive CPU and then set a limit on it by using *cpulimit.*