Blue Team: Summary of Operations

Olivia Moore

Network Topology

The following machines were identified on the network:

[Windows Host Machine]

- Operating System: Windows 10 Pro
- Purpose: Nesting VMs
- IP Address: 192.168.1.1

[Kali Machine]

- Operating System: Kali Linux
- Purpose: Attacker Machine
- IP Address: 192.168.1.90

[Target 1]

- Operating System: Linux
- Purpose: Target Machine
- IP Address:192.168.1.110

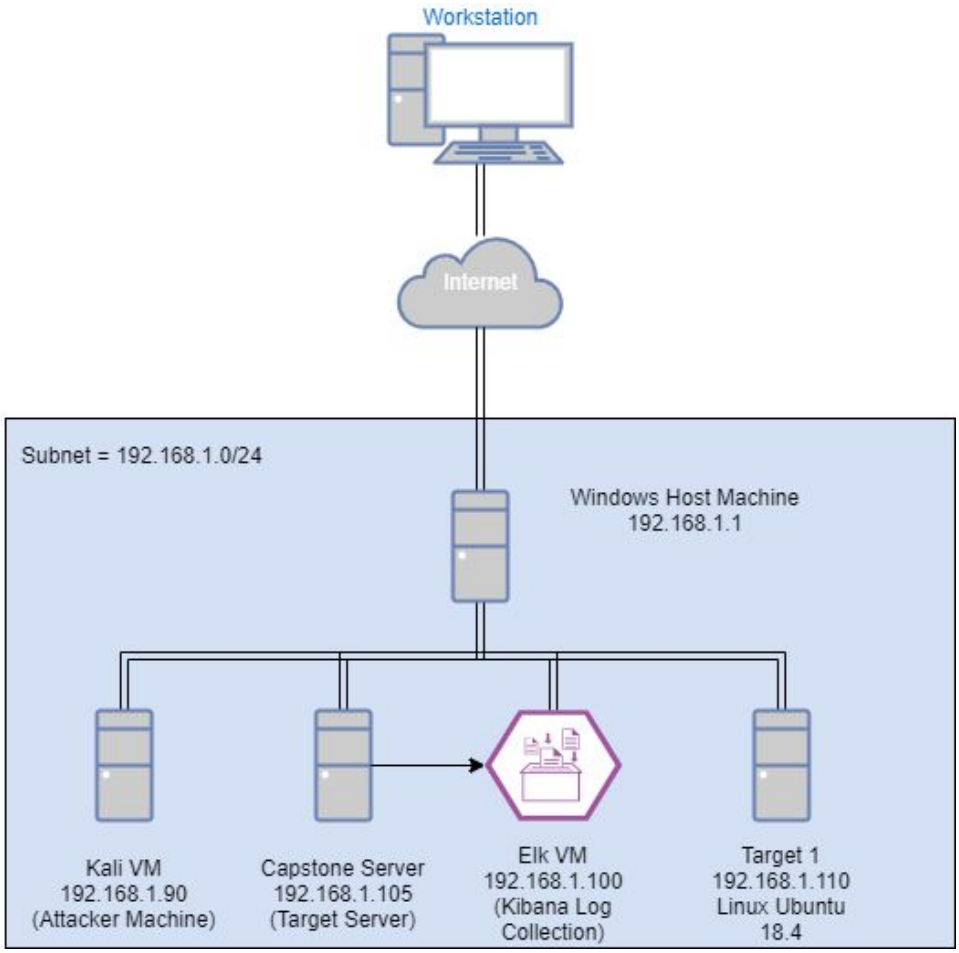
[ELK Machine]

- Operating System: Linux (Ubuntu 18.04)
- Purpose: Kibana Log Capture
- IP Address: 192.168.1.100

[Capstone Server]

- Operating System: Linux (Ubuntu 18.04)
- Purpose: Target Server
- IP Address: 192.168.1.105

Diagram of Topology



Description of Targets

Fill in the following:

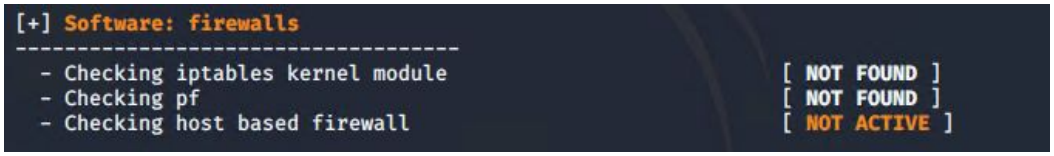
- The VM on the network that was vulnerable to attack: Target 1 192.168.1.110.

- Each VM functions as an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers.

  Target 1: 192.168.1.110

```
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
```

- Ports 445 and 139 can are also frequent points of intrusion that can be exploited for ransomware attacks (e.g. the 2017 WannaCry ransomware attack) as they manage file and printer sharing on the network. Without proper measures in place, such as a firewall, MAC address filtering, or appropriate encryption, these exposed ports can also leave a machine vulnerable to malicious actors.



Monitoring the Targets

This scan identifies the services below as potential points of entry:

- **Target 1**
  - Network Services and Traffic Management
    - Host-Based Firewall [Inactive]
    - Absence of Malware Scanner
    - Vulnerable Ports Open
    - Nil Brute force or DDoS measures in place
  - User Accounts
    - Nil session timeouts in place
    - Vulnerable Packages Installed
    - Access to Private/Sensitive Information for External Actors
  - Password Protection
    - Insufficient Encryption
    - Expired SSL Certificate
    - Insufficient Password Protection
    - Weak Password Parameters and Aging

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

Excessive HTTP Errors

Excessive HTTP Errors is implemented as follows:

- Metric: WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- Threshold: 0
- Vulnerability Mitigated: Stack traces, database dumps, and error messages displayed to external actors (CWE-200: Exposure of Sensitive Information to an Unauthorized Actor (https://cwe.mitre.org/data/definitions/200.html)
- Reliability: Reliability is high, as it would be unlikely that a large number of HTTP responses would fail within the 5 minute period and be something benign.

HTTP Request Size Monitor

HTTP Request Size Monitor is implemented as follows:

- Metric: WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Threshold: 0
- Vulnerability Mitigated: HTTP Request smuggling, resource-consuming DDoS attacks,
- Reliability: TODO: In the event that a large enough file was being sent from an authorized user over the network, it would be easily identifiable if it were to ping this particular alert. The reliability is high.

CPU Usage Monitor

CPU Usage Monitor is implemented as follows:

- Metric: WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Threshold: 0
- Vulnerability Mitigated: Packet dropping and loss of availability
- Reliability: TODO: Alert has the potential to generate false positives, but not frequently enough that an admin would not be able to investigate any time a log message was received that the CPU was above the nominated percentage. This has a high level of reliability.

Suggestions for Going Further

The logs and alerts generated during the assessment suggest that this network is susceptible to several active threats. 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:

General Overview

It is absolutely paramount that all vendor security and integrity patches are installed and maintained at all times. Without these vital up to date fixes and tweaks to each product, we cannot seek to mask the holes they present by working over the top of them. By referring to the current CIS Benchmarks (https://www.cisecurity.org/cis-benchmarks/), we are able to determine how secure our internet-facing system is, and what we can implement to better protect it from future attacks. Attack response is important, but mitigation is always preferred.

Network Services and Traffic Management

[CWE-552: Files or Directories Accessible to External Parties, CWE-359: Exposure of Private Personal Information to an Unauthorized Actor, and CWE-200: Exposure of Sensitive Information to an Unauthorized Actor]

- Patch: Retrofit MySQL with {sudo mysql-secure-installation}
- Why It Works: For hackers, a database is a goldmine. Though MySQL is preferred for its speed and functionality, the stock standard version of MySQL comes at the cost of security. By using the secure installation command we can mitigate a portion of the risk with a cost-effective measure, as we are not adding another product.

[Weak SSH - CWE-553: Command Shell in Externally Accessible Directory]

- Patch: Amend /etc/ssh/sshd_config for more secure parameters
- Why It Works: From the sshd_config file, we can cover many bases when it comes to SSH security. By adjusting the parameters within, we can harden the restrictions for maximum failed login attempts, empty passwords, X11 forwarding, and idle timeout intervals. These will generally allow for a more secure SSH shell, preventing external actors from tunneling GUIs into applications via SSH, brute-force attacks, and other unwanted breaches.

[Weak Apache Server]

- Patch: Enable a Firewall and arm it with Fail2Ban
- Why It Works: Firewalls are a must-have for most networks, but for that extra level of ongoing security, Fail2Ban is an excellent edition. As it blacklists IP addresses automatically after a predetermined number of failures from a host, it works as a good preventative measure for unwanted access but is particularly strong against scripted attacks and botnets.

[Intrusion Detection]

- Patch: Implement an Intrusion Detection System
- Why It Works: Adding software like Snort or SolarWinds Security Even Manager will allow 24/7 network monitoring for malicious activity and policy violation. While it is less powerful as a standalone solution, it is a worthwhile one when used in conjunction with other security methods and software.

Password Protection and User Accounts

[CWE-521:Weak Password Requirements]

- Patch: Adjust password parameters in PAM
- Why It Works: Making passwords more complex, and raising the minimum character count required is a cost-effective and surprisingly simple way to drastically reduce the chance of an external actor brute-forcing their way into user accounts. The recommended length for passwords should be 8 characters, while complexity should be varied to include a minimum of once uppercase character, one lowercase character, and a number or symbol.

[CWE-262: Not Using Password Aging]

- Patch: Set password to expire by modifying /etc/login.defs
- Why It Works: Another simple security measure that will make a difference against actors attempting to brute-force their way into a user account. Setting passwords to expire every 30-day period so that they must be changed ensures that inside actors who previously had access to a file or directory that is no longer authorized will be kept at bay, while also ensuring external actors will have a harder time getting in at all.

[CWE-307: Improper Restriction of Excessive Authentication Attempts]

- Patch: Reduce the number of failed login attempts and set an Advanced Alert
- Why It Works: By modifying the /etc/login.defs file, the amount of failed attempts to log into a user account can be limited as a means to protect the account from anyone trying to guess the password. In addition to this, setting an advanced alert through Watcher using a JSON file is a great way to keep track of any attempts at brute-force by way of guessing as an alert can be sent as soon as the number of attempts remaining drops to zero.

[CWE-284: Improper Access Control]

- Patch: Use a Role-Based Authentication System or deny access by default
- Why It Works: Allowing only admin and root users to access certain files, databases, and functions can prevent an outside actor from jumping between accounts, changing permissions, and altering files. The use of RBAC to manage user privilege within a single system is widely accepted as a best practice.