Final Engagement

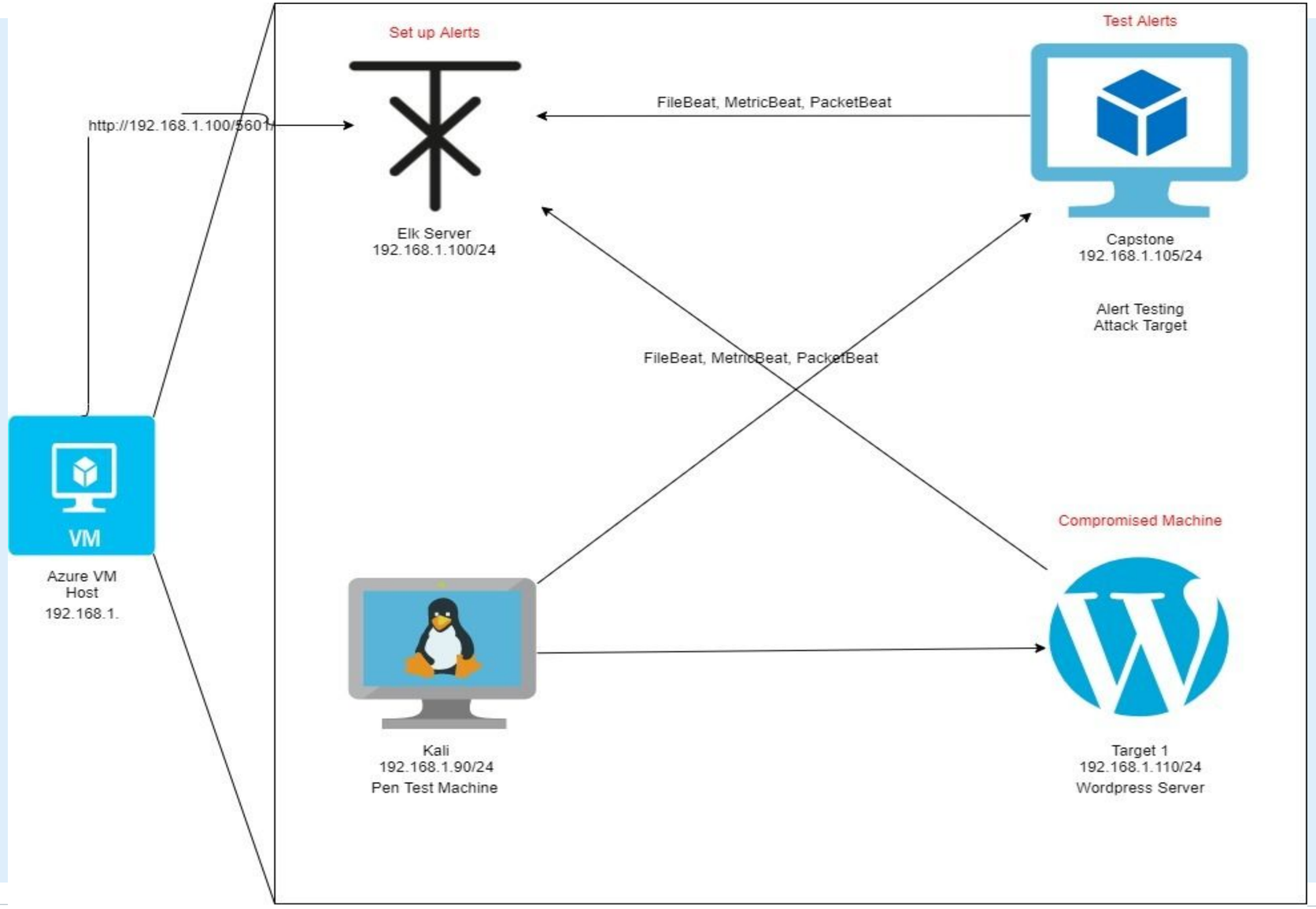
Attack, Defense & Analysis of a Vulnerable Network

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: 10.0.0.1

Machines
IPv4: 192.168.1.90
OS: Kali
Hostname: Kali (Attacker)

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.100
OS: ELK
Hostname: Elk Server

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Public access to port 22	Open and unsecured access	Servers with port 22 open are prone to brute-force attacks
MySQL is not secure, full access	Easily brute forced. Anyone can access data	database username, password, hostname, and the ability to create a database.
Weak name and passwords	Attacker can use numerous names and password combinations to gain access	Attackers gain access to data. Cause monetary, reputational, and operational risks.

Exploits Used

Exploitation: [Open port 22]

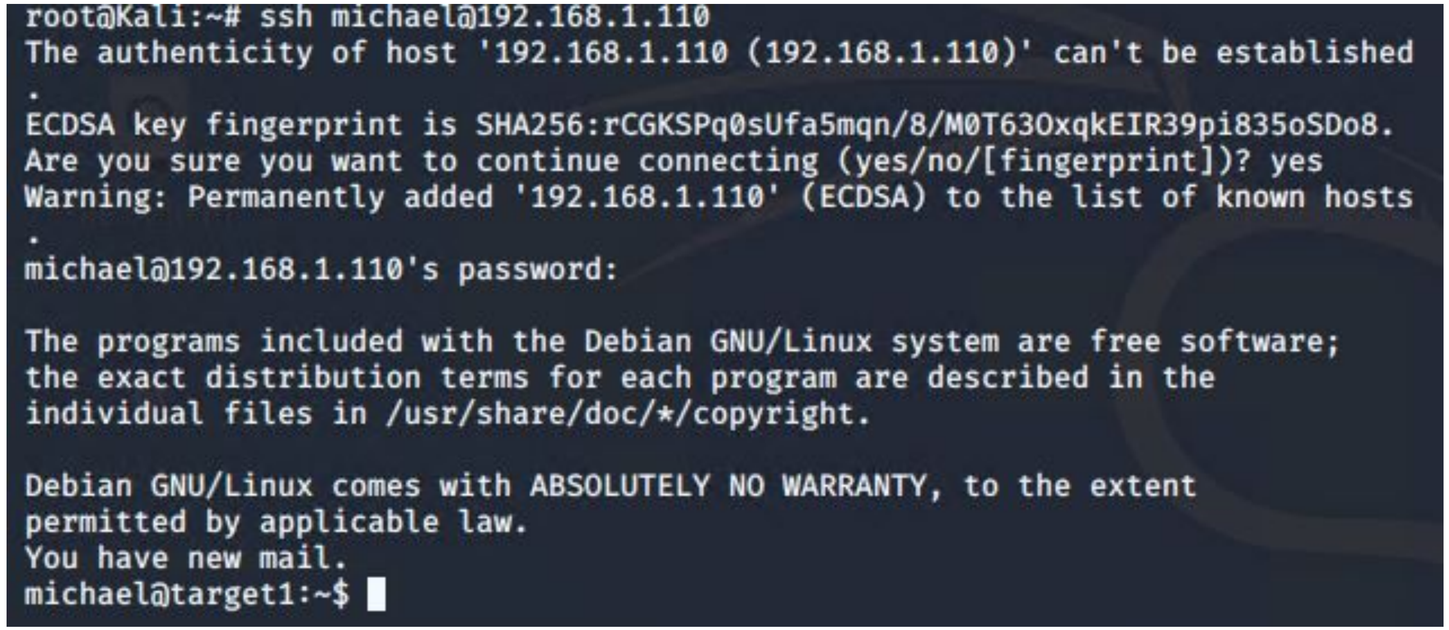
Summarize the following:

- How did you exploit the vulnerability?
  - nmap -sV 192.168.1.110
  - wpscan --url http://192.168.1.110/wordpress -eu
- What did the exploit achieve? Found two user accounts

Exploitation: [Weak passwords and usernames]

Summarize the following:

- How did you exploit the vulnerability?
  - ssh michael@192.168.1.110
  - Guessed Michael’s password
- What did the exploit achieve? Gained shell access



Exploitation: [MySQL Access]

Summarize the following:

- How did you exploit the vulnerability?
  - cat /var/www/html/wordpress/wp-config.php
- What did the exploit achieve? Found hashes for Steve and Michael

```
mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+-----
-------------+----------+---------------------+---------------------+-------
------+---------------+
| ID | user_login | user_pass                          | user_nicename | user
_email        | user_url | user_registered     | user_activation_key | user_s
tatus | display_name   |
+----+------------+------------------------------------+--------------+-----
-------------+----------+---------------------+---------------------+-------
------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | mich
ael@raven.org |          | 2018-08-12 22:49:12 |                     |
    0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | stev
en@raven.org  |          | 2018-08-12 23:31:16 |                     |
    0 | Steven Seagull |
+----+------------+------------------------------------+--------------+-----
-------------+----------+---------------------+---------------------+-------
------+---------------+
2 rows in set (0.00 sec)
```

Avoiding Detection

Stealth Exploitation of [Port 22 Access]

Monitoring Overview

- Which alerts detect this exploit? Port Scan alert, non-whitelisted IP alert, TCP scan
- Which metrics do they measure? Packets being sent, increase in network traffic
- Which thresholds do they fire at? 3500 or above in a 1 minute

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - nmap -f scan which does 100 scans
- Are there alternative exploits that may perform better?
  - --top -ports does less than 100
  - timing template min -rtt-timeout and max-rtt-timeout

Stealth Exploitation of [Weak Passwords and Usernames]

Monitoring Overview

- Which alerts detect this exploit? 401 alarm, increase in traffic, increase in http requests
- Which metrics do they measure? http errors and requests
- Which thresholds do they fire at? 10 failed login attempts within an hour

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - No
- Are there alternative exploits that may perform better?
  - phishing
  - Hydra

Stealth Exploitation of [Unsecured MySQL]

Monitoring Overview

- Which alerts detect this exploit? unauthorized access
- Which metrics do they measure? “GET” request
- Which thresholds do they fire at? 5 or more requests per hour

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
  - Whaling -obtain a superior’s information