Manipulando Pacotes de Rede

+ Vamos montar o pacote IP, que vamos chamar de pIP (Escolhemos um nome aleatório)

```
>>> pIP = IP(dst="192.168.0.1")
```

→ com isso declaramos o ip de destino. Poderíamos também declarar o ip de origem, bastava fazer

```
>>> pIP = IP(dst="192.168.0.1",src="192.168.1.148")
```

→ Como não digitamos nada anteriormente, ele já admite por padrão o nosso endereço como sendo o de origem.

+ Para mostrar mais informações, temos as seguintes opções:

```
>>> pIP.show()
```

```
>>> pIP.summary()
```

+ Montaremos agora o nosso pacote TCP:

```
>>> pTCP = TCP(dport=80, flags="S")
```

```
>>> pTCP
```
→ Para ver informações básicas do pacote

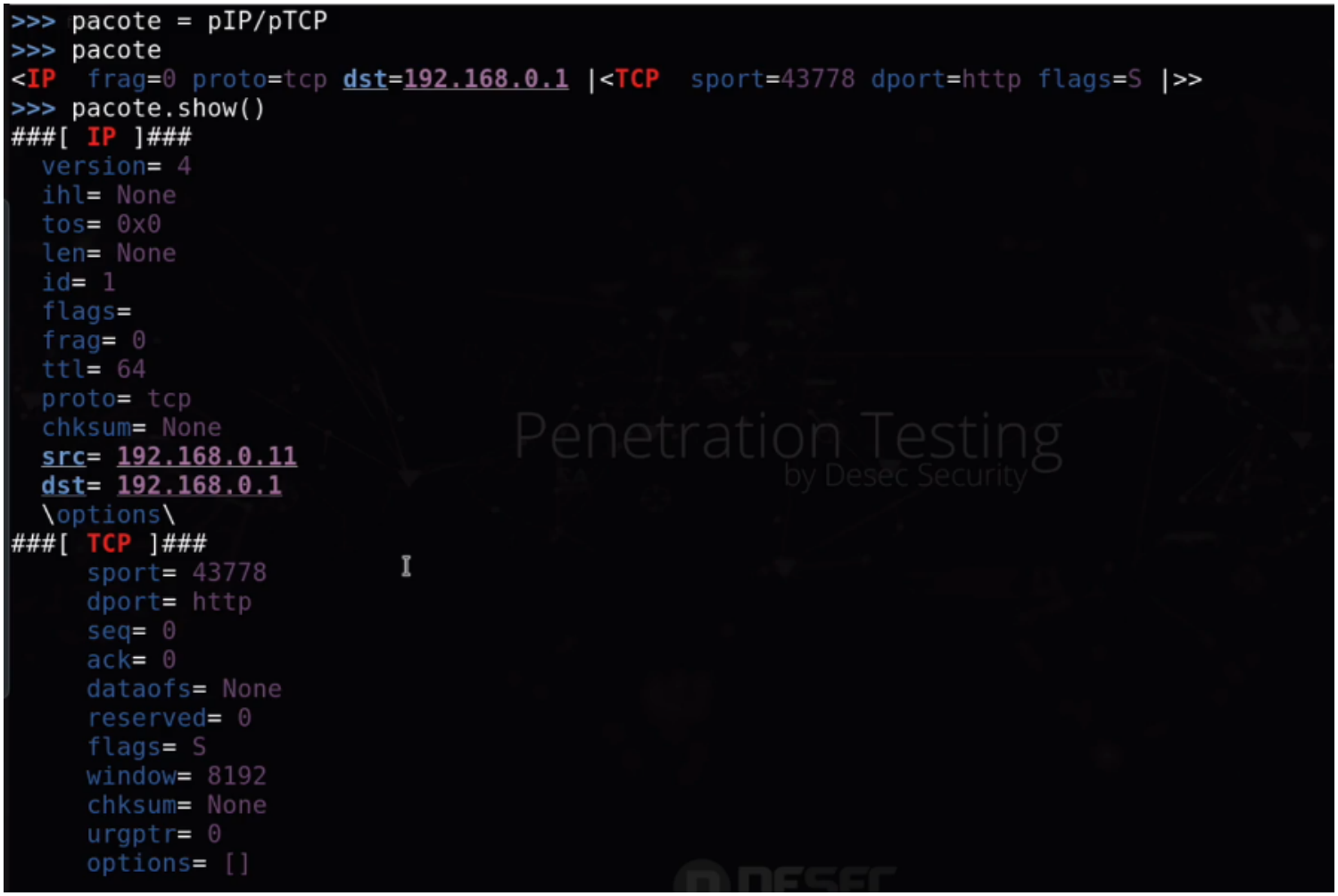
```
>>> pTCP.show()
###[ TCP ]###
  sport= ftp_data
  dport= http
  seq= 0
  ack= 0
  dataofs= None
  reserved= 0
  flags= S
  window= 8192
  chksum= None
  urgptr= 0
  options= []
```

+ Podemos determinar a porta de origem:

```
>>> pTCP.sport=43778
```

+ Para encapsular o pacote TCP dentro do IP, basta:

```
>>> pacote = pIP/pTCP
```



→ isso é semelhante ao wireshark

+ Para enviar o pacote:

```
>>> sr1(pacote)
```

+ Para enviar a saída do envio para uma determinada variável chamada "resposta"

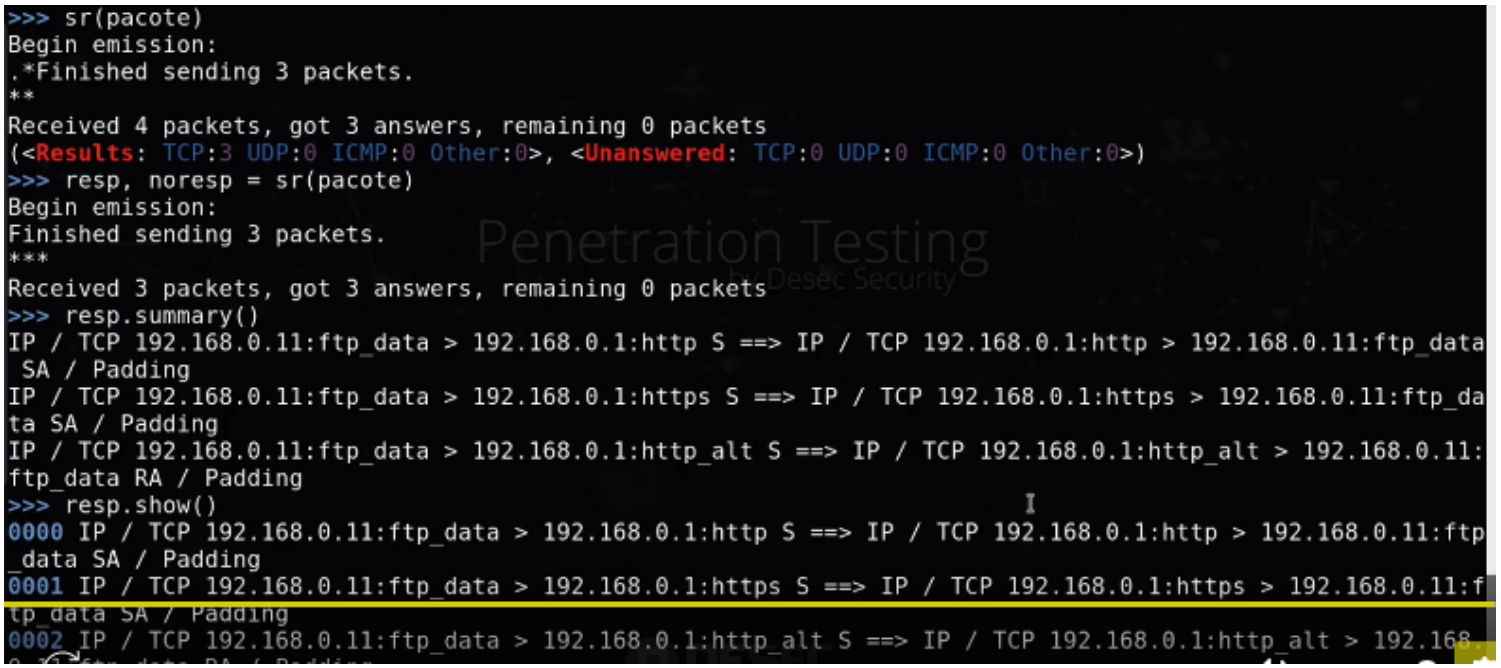
```
>>> resposta = sr1(pacote)
```

+ Podemos setar as informações específicas que queremos também
→ Se quisermos, por exemplo, o ip de destino

```
>>> resposta[IP].dst
```

+ Se quisermos fazer mais pacotes (tipo, iguais que ataquem portas diferentes), basta adicionarmos o [] que indica o range que vamos determinar.
→ No exemplo a seguir, vamos fazer o TCP mandar pacotes para as portas 80,443,8080
→ `>>> pTCP = TCP(dport=[80,443,8080])`
→ + Depois, encapsulamos novamente o pacote para enviá-lo
→ `>>> pacote = pIP/pTCP`

+ Para pegar a resposta do envio do pacote, gravar os resultados em resp e os que não deram resposta em noresp, basta seguir os passos abaixo


>>> sr(pacote)
Begin emission:
.*Finished sending 3 packets.
**
Received 4 packets, got 3 answers, remaining 0 packets
(<Results: TCP:3 UDP:0 ICMP:0 Other:0>, <Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>)
>>> resp, noresp = sr(pacote)
Begin emission:
Finished sending 3 packets.
***
Received 3 packets, got 3 answers, remaining 0 packets
>>> resp.summary()
IP / TCP 192.168.0.11:ftp_data > 192.168.0.1:http S ==> IP / TCP 192.168.0.1:http > 192.168.0.11:ftp_data SA / Padding
IP / TCP 192.168.0.11:ftp_data > 192.168.0.1:https S ==> IP / TCP 192.168.0.1:https > 192.168.0.11:ftp_data SA / Padding
IP / TCP 192.168.0.11:ftp_data > 192.168.0.1:http_alt S ==> IP / TCP 192.168.0.1:http_alt > 192.168.0.11:ftp_data RA / Padding
>>> resp.show()
0000 IP / TCP 192.168.0.11:ftp_data > 192.168.0.1:http S ==> IP / TCP 192.168.0.1:http > 192.168.0.11:ftp_data SA / Padding
0001 IP / TCP 192.168.0.11:ftp_data > 192.168.0.1:https S ==> IP / TCP 192.168.0.1:https > 192.168.0.11:ftp_data SA / Padding
0002 IP / TCP 192.168.0.11:ftp_data > 192.168.0.1:http_alt S ==> IP / TCP 192.168.0.1:http_alt > 192.168