

Ferramentas de Rede

João Marcelo Uchôa de Alencar

Universidade Federal do Ceará - Quixadá

31 de Outubro de 2019

Introdução

- ▶ **wget**: baixar conteúdo da *web*
- ▶ **nc**: brincando com portas TCP/IP
- ▶ **netpipes**: parecido com o *nc*, porém com algumas facilidades.

O comando *wget*

```
$ wget http://www.quixada.ufc.br/docentes
```

Se for uma página irá baixar o .html, se for um arquivo baixará o mesmo.

O comando *nc* (*netcat*)

Devemos deixar uma máquina escutando em uma conexão enquanto outras da rede se conectam e trocam informações.

Em uma máquina que irá escutar por conexões:

```
$ nc -l -p 8080
```

Na máquina cliente:

```
$ nc scripts.joao.marcelo.nom.br 8080
```

O que você digitar na máquina cliente, irá aparecer na tela do servidor.

Trocando arquivos:

```
$ nc -l -p 8080 > arquivo.destino
```

```
$ nc scripts.joao.marcelo.nom.br 8080 < arquivo.origem
```

Na verdade, o *nc* pode ser colocado dentro de comando com *pipes*, dessa forma a saída pode ser encaminhada para um *grep*, *awk*, etc...

O pacote *netpipes*

São um conjunto de comandos

- ▶ **faucet**: servidor
- ▶ **hose**: cliente
- ▶ **encapsulate**: faz multiplexação
- ▶ **timelimit**: limita o tempo de um processo criado para lidar com uma conexão
- ▶ **sockdown**: desliga uma conexão
- ▶ **getpeername**: retorna informações sobre a conexão

\$ sudo apt-get install netpipes

O comando *faucet*

No servidor:

```
faucet 8080 --out echo Olá Mundo
```

Em um cliente, você pode fazer:

```
$ telnet scripts.joao.marcelo.nom.br 8080
```

```
$ nc scripts.joao.marcelo.nom.br 8080
```

Em ambos os casos, o cliente exibirá a mensagem, mas o servidor não será finalizado. Novas conexões irão exibir a mensagem novamente.

```
$ faucet 8080 --out sh -c "echo Olá Cliente; w;"
```

O comando *hose*

É a versão cliente do *faucet*.

No servidor:

```
$ faucet 8080 --in sh -c "cat"
```

No cliente:

```
$ hose scripts.joao.marcelo.nom.br 8080 \  
--out sh -c "echo Olá Mundo do Cliente"
```

Resumo

No servidor:

```
$ faucet 8080 --out echo hello world
```

No cliente:

```
$ hose scripts.joao.marcelo.nom.br 8080 \  
--netslave
```


Resumo

No servidor:

```
$ faucet 8080 --in echo Fui acessado, `date`
```

No cliente:

```
$ hose scripts.joao.marcelo.nom.br 8080 \  
--netslave
```

Resumo

No servidor:

```
$ faucet 8080 --in sh -c "grep firefox > temp.txt"
```

No cliente:

```
$ hose scripts.joao.marcelo.nom.br 8080 \  
--in --out sh -c "ps aux"
```

Resumo

No servidor:

```
$ faucet 8080 --out sh -c "ps aux"
```

No cliente:

```
$ hose scripts.joao.marcelo.nom.br 8080 \  
--netslave
```

Exercícios em Sala

Faça um sistema de monitoramento simples.

- ▶ O *script cliente.sh* deve receber dois parâmetros: IP e Porta. Em seguida, deve enviar ao servidor o seu *hostname* seguido da saída do comando *uptime*
- ▶ O *script servidor.sh* deve escutar no IP e Porta fornecidos ao cliente. A cada conexão, deve guardar as informações recebidas do cliente no arquivo *monitoramento.log*

Não deixe o servidor escutando em um IP válido!!!

Atividades

Coloque os arquivos na pasta *atividades/atividade12*. Vamos fazer um pequeno servidor de arquivos.

- ▶ O *script cliente.sh* deve receber três parâmetros: IP do Servidor, Porta do Servidor e o nome de um arquivo. Ele deve se conectar ao servidor enviando além do nome do arquivo desejado, o IP e Porta na qual espera o recebimento do arquivo.
- ▶ O *script servidor.sh* deve escutar no IP e Porta fornecidos ao cliente. A cada conexão, deve procurar o arquivo requisitado no diretório no qual está executando e enviá-lo através do IP e Porta fornecidos pelo cliente.

Para funcionar, tanto cliente quanto servidor devem estar na mesma rede.