Lab 4 – Análise de Pacotes utilizando o TCPDUMP

Objetivo:

i) Utilizar aplicativo de análise de pacotes TCPDUMP.

TCPDUMP:

O **tcpdump** é um programa cuja utilidade principal é visualizar e analisar o tráfego de uma rede local.

Verifique mais informações sobre o tcpdump nas páginas man do Linux.

Utilizando o tcpdump, devemos, primeiramente, identificar qual interface queremos observar, ou seja, de qual interface queremos capturar os pacotes processados. Para verificar as interfaces existentes, em qualquer Unix, fazemos uso do comando *ifconfig*. As interfaces padrão são eth0, eth1, etc.

Exemplo:

```
[root@jagger root]# ifconfig
eth0    Encapsulamento do Link: Ethernet  Endereço de HW 00:07:95:BB:37:41
        inet end.: 192.168.1.35  Bcast:192.168.1.255  Masc:255.255.255.0
        endereço inet6: fe80::207:95ff:febb:3741/64 Escopo:Link
        UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
        RX packets:7543 errors:0 dropped:0 overruns:0 frame:0
        TX packets:4052 errors:0 dropped:0 overruns:0 carrier:0
        colisões:0 txqueuelen:1000
        RX bytes:3006444 (2.8 Mb)  TX bytes:602747 (588.6 Kb)
        IRQ:11 Endereço de E/S:0xd400
```

Quando se executa o tcpdump sem passar nenhum parâmetro o Linux captura, por default, os pacotes presentes na eth0. Caso se deseje, por exemplo, capturar os pacotes da interface eth1, utilizamos o parâmetro **–i**. Da seguinte maneira:

```
[root@root /root]$tcpdump –i eth1
```

Quando estamos observando a rede pode ser interessante que o tcpdump não converta os endereços IP e números de portas para nomes, daí utiliza-se o parâmetro **–n**. Exemplo:

```
[root@root /root]$tcpdump –n –i eth1
```

Para estabelecer o tamanho dos dados que serão capturados usamos o parâmetro **–s** tam, onde o tam é o tamanho que nos interessa. Por default o tcpdump só captura os primeiros 68 bytes, o que é útil somente se desejamos capturar os cabeçalhos dos protocolos IP, TCP ou UDP. O restante dos dados do pacote serão truncados. É interessante utilizar como MTU (*Maximum Transfer Unit*) médio 1500 bytes. Usando a opção –s 1500 poderemos capturar quadros Ethernet completos. Exemplo:

```
[root@root /root]$tcpdump –n –i eth0 –s 1500
```

Podemos usar as opções **–v**, **-vv** ou **–vvv** em função da quantidade de detalhes que queremos que o tcpdump interprete, ou seja, cada "**v**" aumenta o grau de informações de saída do comando.

Se quisermos, imprimir o conteúdo dos pacotes (Hexadecimal) podemos usar a opção **–x**. Mas se queremos que o conteúdo dos pacotes sejam impressos em ASCII, podemos usar a opção **–X**. O tamanho dos dados que serão impressos é determinado pela opção –s, a ausência do parâmetro implica em 68 bytes. Exemplos:

```
[root@root /root]$tcpdump -n -x -i eth1
```
-> Imprime, somente, os primeiros 68 bytes do pacote.

```
[root@root /root]$tcpdump -n -x -i eth1 -s 1500
```
-> Imprime os 1500 bytes do pacote.

Se quisermos capturar um número limitado de pacotes, podemos utilizar a opção **–c**, indicando o número de pacotes que desejamos capturar. Exemplo:

```
[root@root /root]$tcpdump -c 10 -n -x -i eth1
```
-> Imprime, somente, os primeiros 68 bytes de cada pacote, parando a captura após 10 pacotes capturados.

Interpretando as saídas do tcpdump:

Primeiramente temos que entender que a saída depende do protocolo que estamos analisando. Para começar, devemos que saber que para todos os pacotes capturados, é apresentado, como primeiro campo, uma marca de tempo, que indica o tempo, com precisão em milésimos de segundo, da captura.

12:35:21.457350 200.135.233.1.1025 > 200.135.233.55.1345: udp 121 **[ttl1]**

A opção **–t** faz com que o tcpdump não imprima esta marca de tempo.

Segmentos TCP:

A linha geral de um segmento TCP é a seguinte:

src > dst : flags [seq ack windows urgent opções]

O protocolo TCP é o responsável pelo serviço de transferência garantida da Internet. O src, o dst e as flags estão sempre presentes. Os outros campos dependem do tipo de conexão TCP. O significado destes parâmetros são:

- **src**: maquina e porta de origem.
- **dst**: maquina e porta de destino.

Quando não se especifica o parâmetro **–n**, o tcpdump converte, via DNS, o IP ao nome correspondente, o qual é apresenta na saída do tcpdump.

- **flags**: São indicadas no cabeçalho TCP. Cujo significado depende exclusivamente da combinação em que aparecem:

A (ACK) _ Indica reconhecimento válido
R (RST)_ Usada para restabelecer uma conexão
S (SYN)_ Usada para abertura de conexão
F (FIN)_ Usada para encerramento de conexão
U (URG)_ Usado para indicar um dado urgente
P (PSH)_ Usada para indicar que o receptor deve passar imediatamente o dado para a camada aplicação.

- **seq**: Equivale ao numero de seqüência do primeiro byte de dados deste segmento tcp. O formato é o seguinte:

primeiro:último (sem incluir o último)

Significa que entre o primeiro e o último (sem incluir o último) existe um total de *n* bytes de dados, que aparecerá entre parênteses. Exceto quando existem segmentos com SYN, que também ocupam um numero de sequência.

- **ack**: É o número de reconhecimento. Indica o número seguinte de seqüência que deseja receber.

Em geral o tcpdump imprime os números de **seq** e **ack** de forma relativa. Isto é, mostra os números **seq** reais somente para os pacotes relativos à abertura de conexão. Para os demais pacotes são apresentados números relativos ao número de Bytes transmitidos (onde o primeiro segmento de dados recebe o número de seqüência 1).

Para fazer com que sejam impressos os números **seq** e **ack** absolutos, devemos utilizar a opção **–S**.

- **win**: Tamanho da janela de recepção.
- **urgent**: Indica a existência de dados urgentes.
- **opções**: Indica a existência de opções.

Exemplo: (Extraído da página de manual (man) do tcpdump:

1. Rtsg.1023 > csam.login: S 768512:768512(0) win 4096 <mss 1024>
2. Csam.login > rtsg.1023: S 947648:947648(0) ack 768513 win 4096 <mss 1024>
3. Rtsg.1023 > csam.login: . ack 1 win 4096
4. Rtsg.1023 > csam.login: P 1:2(1) ack 1 win 4096
5. Csam.login > rtsg.1023: . ack 2 win 4096
6. Rtsg.1023 > csam.login: P 2:21(19) ack 1 win 4096
7. Csam.login > rtsg.1023: P 1:2(1) ack 21 win 4077
8. Csam.login > rtsg.1023: P 2:3(1) ack 21 win 4077 urg 1
9. Csam.login > rtsg.1023: P 3:4(1) ack 21 win 4077 urg 1

Este exemplo simula uma conexão iniciada pela maquina rtsg com destino a maquina csam, executando o serviço rlogin. Note que não foi utilizada a opção –n, pois os nomes não foram traduzidos para números de IP/porta. O significado das linhas anteriores são:

a) Inicio da conexão, SYN de rtsg -> csam. Flags setadas:S; seq:768512; ack:0; win:janela de 4096B; urgent:0 e opções:mss1024.
b) SYN/ACK de csam -> rtsg. Flags setadas:S; seq:947648; ack:768513; win:janela de 4096B; urgent:0 sem opções.
c) ACK mandado por csam. Não aparecem flags; seq não é mostrada pois serão relativos ao número escolhido no início; ack:1 (relativo ao anterior); win:janela de 4096B; urgente:0 sem opções.
d) 1 byte de dados de rtsg -> csam. Flags setadas:P; seq:1-2 (os números de seqüência são relativos ao número inicial a menos que se especifique a opção –S, que imprime os números de seqüência de maneira absoluta; ack:1; win:janela de 4096B; urgent:0 sem opções.
e) ACK de dados por parte de csam.
f) 19 bytes de dados de rtsg a csam.
g) csam manda 1 byte de dados a rtsg e o reconhecimento (ack) dos 19 bytes recebidos anteriormente de rtsg. Flag P setada. Janela de recepção com 4077B.
h) csam envia a rtsg 1 byte de dados urgente a rtsg. Flag P setada. Janela de recepção com 4077B.
i) Idem a anterior.

UDP:

Um pacote UDP aparece da seguinte maneira:

Origem.portaorigem > destino.portadestino: udp long

- **origem:** Nome ou IP da maquina origem;
- **portaorigem**: porta de origem do pacote;
- **destino:** Nome ou IP da maquina de destino;
- **portadestino:** porta a qual o pacote esta destinado;
- **long:** Indica a longitude dos dados do usuário.

Exemplo:

12:35:21.457350 200.135.233.1.1025 > 200.135.233.55.1345: udp 121 **[ttl1]**

Filtros:

Das atividades realizadas com o tcpdump, as mais importantes são executadas com o uso de filtros. Um filtro é uma expressão que sucede as opções e que nos permite selecionar os pacotes que desejamos capturar. Na ausência de filtros o tcpdump se capturara todo o tráfego do adaptador de rede selecionado.

As expressões usadas para definir os filtros seguem uma série de primitivas, com três possíveis "campos".
Os três possíveis campos são:

• *tipo*. Pode ser **host, net** e **port**. Indicam respectivamente, uma maquina, uma rede completa ou uma porta concreta.

Exemplos

Tcpdump host 200.135.233.55

Tcpdump net 200.135.233/**24** -> mascara de rede associada

Tcpdump port 80

Por default, o tcpdump assume o tipo **host**.

• *Direção*. Especifica de onde e para onde vão os pacotes que queremos capturar. Temos **src** e **dst**, podem ser usados isoladamente ou combinados com **or** ou **and**. Caso estejamos trabalhando com protocolos do tipo ponto-a-ponto podemos substitui-los por inbound ou outbound. Por exemplo, se queremos capturar pacotes com origem na 200.135.233.55 e destinados a 200.135.233.1, basta aplicar um dos filtros abaixo:

tcpdump dst 200.135.233.1 and src 200.135.233.55

tcpdump dst 200.135.233.1 or src 200.135.233.55

Se não especificamos a direção o tcpdump se põe a capturar pacotes **src or dst**.

Podemos combinar estas opções de filtros com os vistos anteriormente.

• *Protocolo*. Permite selecionar o protocolo dos pacotes que desejamos capturar. Pode ser: **tcp, udp, ip ether, arp, rarp** e **fddi** (para redes FDDI, sendo seu encapsulamento exatamente igual ao ether).

- Captura o tráfego cujo IP origem é igual a 200.135.233.55
  Tcpdump src host 200.135.233.55

- Captura todo o tráfego com destino ou origem na maquina 200.135.233.55
  Tcpdump host 200.135.233.55

- Captura o tráfego com destino no adaptador de rede, cujo MAC é 0:2:a5:e:ec:10
  Tcpdump ether dst 0:2:a5:e:ec:10

- Captura os pacotes (destinados ou oriundos) da maquina cujo MAC seja igual a 0:2:a5:e:ec:10
  Tcpdump ether host 0:2:a5:e:ec:10

- Captura todo o tráfego destinado a rede 200.135.233.0
  Tcpdump dst net 200.135.233.0

- Captura todo o tráfego com origem na rede 200.135.233.0/24
  Tcpdump src net 200.135.233.0/24
  Tcpdump src net 200.135.233.0 mask 255.255.255.0

- Captura os pacotes destinados a porta 23 do host em questão
  Tcpdump dst port 23

- Captura todo o tráfego com origem ou destino na porta 80 deste host
  Tcpdump port 80

- Captura todo o tráfego arp
  Tcpdump arp
  Tcpdump ether proto \\arp

- Captura todo o tráfego ip
  Tcpdump ip
  Tcpdump ether proto \\ip

- Captura todo os pacotes udp
  Tcpdump udp
  Tcpdump ip proto \\udp

Combinando os Filtros

Pode-se combinar as expressões anteriores com a ajuda dos operadores **not**, **and** e **or** (correspondem a negação, E lógico e o OU lógico), criando assim, filtros mais seletos.

Exemplos:

- Captura todo o tráfego WEB (TCP porta 80)

  Tcpdump tcp and port 80

- Captura todas os pacotes DNS

  Tcpdump udp and port 53

- Captura o tráfego da porta telnet ou ssh

  Tcpdump tcp and \(port 22 or port 23\)

- Captura todo o tráfego da rede, exceto os pacotes referentes a aplicação WWW (web)

  Tcpdump tcp and not port 80

Exercícios:

1) Verificar qual o seu IP usando ifconfig.

2) Fazer captura de pacotes usando simplesmente:

$ tcpdump –i eth0

OBS: Neste modo de funcionamento a placa captura qualquer pacote no seu segmento de rede.

3) Analise o comando abaixo, identificando os parâmetros que estão sendo utilizados, e em seguida realize uma captura de pacotes com este comando:

$ tcpdump –i eth0 –tnXS –s 1500 tcp and host <IP_local>

5) Implemente um comando tcpdump para capturar uma seqüência de pacotes trocados durante uma sessão na porta 80, entre seu computador e o servidor web do CEFET-SC. Analise os números de seqüência e reconhecimento utilizados pela máquina cliente e servidora. Procure identificar, entre os pacotes capturados, a seqüência de pacotes que definem as três fases da abertura da conexão TCP realizada pela aplicação telnet, observando os valores dos flags TCP (S e A).

6) Faça um filtro utilizando o tcpdump para capturar tráfego icmp na porta 53 (DNS) da sua máquina