Exercício 2 de MC833 — Programação em Redes de Computadores

Raul Rabelo Carvalho, 105607, turma A

11 de Março de 2014

1 Quais são as interfaces nas quais o tcpdump pode escutar/capturar dados? Essas interfaces são as mesmas mostradas pelo comando ifconfig?

Para verificar quais interfaces de rede estão disponíveis para captura de pacotes ao `tcpdump`, deve-se usar o comando `tcpdump -D`. Na máquina sabbath da sala CC302 nenhuma interface está disponível para captura de pacotes, pois os alunos não podem executar comandos como root. O comando `ifconfig -a`, no entanto, lista duas interfaces: lo e em1, como mostrado abaixo.

Em um laptop conectado a um *NAT router* de uma rede doméstica pela interface sem fio, temos para o comando `sudo tcpdump -D` sete interfaces listadas, como mostrado abaixo.

```
rrcarvalho@Tarkin ~/MC823/exercicio2 $ sudo tcpdump -D
1.nflog (Linux netfilter log (NFLOG) interface)
2.nfqueue (Linux netfilter queue (NFQUEUE) interface)
3.dbus-system (D-Bus system bus)
4.enp2s0
5.wlp3s0
6.any (Pseudo-device that captures on all interfaces)
7.lo
```

As duas primeiras linhas da saída mostram *kernel hooks* do projeto Netfilter[1] para filtragem de pacotes no Linux; a terceira linha é um *Unix socket* usado para comunicação entre processos. A partir da quarta linha temos as interfaces de rede; sendo `enp2s0` a interface Ethernet do laptop, enquando wlp3s0 é a interface sem fio; na sétima linha é mostrado a interface associada ao localhost da máquina.

O comando `ifconfig -a`, no entando, lista somente três interfaces:

```
rrcarvalho@Tarkin ~/MC823/exercicio2 $ ifconfig -a
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
       ether dc:0e:a1:c7:9f:29 txqueuelen 1000 (Ethernet)
       RX packets 0 bytes 0 (0.0 B)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 0 bytes 0 (0.0 B)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
       inet 127.0.0.1 netmask 255.0.0.0
       inet6 ::1 prefixlen 128 scopeid 0x10<host>
       loop  txqueuelen 0 (Local Loopback)
       RX packets 434 bytes 33476 (32.6 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 434 bytes 33476 (32.6 KiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet 10.17.23.103 netmask 255.255.255.0 broadcast 10.17.23.255
       inet6 fe80::5ec9:d3ff:fe14:2f4a prefixlen 64 scopeid 0x20<link>
       ether 5c:c9:d3:14:2f:4a txqueuelen 1000 (Ethernet)
       RX packets 9995 bytes 3852001 (3.6 MiB)
       RX errors 0 dropped 1299 overruns 0 frame 0
```

[1] http://netfilter.org/

```
        TX packets 1437 bytes 199562 (194.8 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

As interfaces, neste caso, são todas de rede e são a interface Ethernet (enp2s0), o localhost (lo) e a interface sem fio (wlp3s0).

2 Qual é o endereço IP do nós maple e willow na rede em questão?

Para descobrir os endereços IP das duas máquinas da comunicação registrada em `tcpdump.dat`, é possível utilizar a opção `-n` do `tcpdump`. Deve-se executar o comando `tcpdump -r tcpdump.dat > outfile.txt` e procurar uma linha com os nomes para os quais se quer o endereço IP; no caso a terceira linha contém os dois nomes:

```
01:34:41.473518 IP willow.csail.mit.edu.39675 > maple.csail.mit.edu.commplex-link: Flags
      [S], seq 1258159963, win 14600, options [mss 1460,sackOK,TS val 282136473 ecr
      0,nop,wscale 7], length 0
```

Em seguida, executa-se `tcpdump -r tcpdump.dat -n > enderecos.txt` e verifica-se a mesma terceira linha:

```
01:34:41.473518 IP 128.30.4.222.39675 > 128.30.4.223.5001: Flags [S], seq 1258159963, win
      14600, options [mss 1460,sackOK,TS val 282136473 ecr 0,nop,wscale 7], length 0
```

Temos, então, que a máquina com o nome willow está no endereço IP `128.30.4.222` e a máquina maple está em `128.30.4.223`. Note que a mesma informação poderia ser obtida usando o programa `nslookup`.

3 Qual é o endereço MAC dos nós maple e willow?

Ainda que seja possível se obter os endereços IP das máquinas willow e maple consultando um servidor DNS, endereços MAC não estão disponíveis externamente à rede Ethernet local. Ainda sim, o registro `tcpdump.dat` contém os endereços MAC de cada uma das máquinas de origem e destino dos pacotes capturados pelo `tcpdump`. A opção `-e` exibe estas informações.

Seria possível obter o endereço MAC na terceira linha novamente, mas será empregado um filtro para listar somente os pacotes ARP, que é o protocolo de manutenção da rede Ethernet.

```
rrcarvalho@Tarkin ~/MC823/exercicio2 $ tcpdump -r tcpdump.dat -e arp
reading from file tcpdump.dat, link-type EN10MB (Ethernet)
01:34:41.473036 00:16:ea:8e:28:44 (oui Unknown) > Broadcast, ethertype ARP (0x0806), length
      42: Request who-has maple.csail.mit.edu tell willow.csail.mit.edu, length 28
01:34:41.473505 00:16:ea:8d:e5:8a (oui Unknown) > 00:16:ea:8e:28:44 (oui Unknown),
      ethertype ARP (0x0806), length 42: Reply maple.csail.mit.edu is-at 00:16:ea:8d:e5:8a
      (oui Unknown), length 28
```

No primeiro pacote, temos uma máquina no endereço MAC `00:16:ea:8e:28:44` requisitando o endereço físico da máquina maple. Como a comunicação capturada ocorre somente entre as máquinas maple e willow, podemos assumir que este endereço MAC é da máquina willow. Mas essa informação pode ser confirmada observando o terceiro pacote mais adiante.

Já no segundo pacote, alguma máquina responde que maple está no endereço MAC `00:16:ea:8d:e5:8a`; no caso, a própria maple foi a máquina que respondeu ao pedido da willow.

Para verificar o endereço MAC da máquina willow, observa-se no terceiro pacote listado pelo comando `tcpdump -r tcpdump.dat -e > ethernet.txt` que `00:16:ea:8e:28:44` é de fato o endereço de willow.

```
01:34:41.473518 00:16:ea:8e:28:44 (oui Unknown) > 00:16:ea:8d:e5:8a (oui Unknown),
     ethertype IPv4 (0x0800), length 74: willow.csail.mit.edu.39675 >
     maple.csail.mit.edu.commplex-link: Flags [S], seq 1258159963, win 14600, options [mss
     1460,sackOK,TS val 282136473 ecr 0,nop,wscale 7], length 0
```

4 Qual é a porta TCP usada pelos nós maple e willow? Qual é o tipo de porta que está sendo utilizada pela fonte nesta conexão?

Usando novamente a opção `-n`, pode-se obter o número das portas utilizadas por maple e willow. Sem esta opção, mesmo algumas portas acima de 1024 são associadas a uma aplicação oficial e assim mostradas. No caso específico da conexão estudada, a porta de destino da conexão (na máquina maple) é 5001, registrada na IANA como `commplex-link`, mas que, no entanto, pode não ser a aplicação usada nesta conexão, já que esta é uma porta associada a múltiplas aplicações não oficiais.

```
01:34:41.473518 IP 128.30.4.222.39675 > 128.30.4.223.5001: Flags [S], seq 1258159963, win
     14600, options [mss 1460,sackOK,TS val 282136473 ecr 0,nop,wscale 7], length 0
```

A porta na máquina willow, a qual originou a conexão, é 39675. Esta é uma porta bem acima das *well-known ports*, escolhida aleatoriamente pela máquina para identificar unicamente esta conexão; esta porta não está associada a nenhuma aplicação específica.

5 Quantos kilobytes foram transferidos durante essa sessão TCP? Qual foi a duração da sessão? Baseado nas respostas anteriores, qual é a vazão (em Kilobytes/segundos) do fluxo TCP entre willow e maple?

Para se descobrir quantos bytes foram enviados na conexão, nota-se que o `tcpdump` usa uma numeração relativa para exibir o campo SEQ do cabeçalho TCP. O primeiro pacote de push da conexão é:

```
01:34:41.474166 IP willow.39675 > maple.commplex-link: Flags [P.], seq 1:25, ack 1, win
     115, options [nop,nop,TS val 282136474 ecr 282202089], length 24
```

Foi usado o comando `tcpdump -r tcpdump.dat -N > short.txt` para se obter a lista de pacotes de onde a linha acima foi retirada.

Assim, basta localizar o último pacote ACK reconhecendo o recebimento de um intervalo de bytes. O último pacote ACK desta sequência é:

```
01:34:44.329956 IP maple.commplex-link > willow.39675: Flags [.], ack 1572890, win 820,
     options [nop,nop,TS val 282204945 ecr 282139320], length 0
```

que reconhece o recebimento do pacote:

```
01:34:44.311921 IP willow.39675 > maple.commplex-link: Flags [FP.], seq 1572017:1572889,
     ack 1, win 115, options [nop,nop,TS val 282139311 ecr 282204927], length 872
```

que é o pacote com as flags FIN e PUSH ativadas.

Portanto, pode-se afirmar que foram enviados 1 572 889 bytes em um intervalo de 2,865497 segundos. O tempo foi calculado desde o início do *three-way handshake* até que willow reconheça o encerramento da sessão; os dois pacotes referentes a este cálculo estão listados abaixo.

```
01:34:41.473518 IP willow.39675 > maple.commplex-link: Flags [S], seq 1258159963, win
     14600, options [mss 1460,sackOK,TS val 282136473 ecr 0,nop,wscale 7], length 0
.
.
.
01:34:44.339015 IP willow.39675 > maple.commplex-link: Flags [.], ack 2, win 115, options
     [nop,nop,TS val 282139339 ecr 282204955], length 0
```

A taxa de transferência foi, então, de aproximadamente `548,9062 kB/s`.

6 Qual é o round-trip time (RTT), em segundos, entre willow e maple baseado no pacote 1473:2921 e seu acknowledgment? Veja o arquivo outfile.txt e encontre o RTT do pacote 13057:14505. Porque esses dois valores são diferentes?

Para o cálculo do RTT do pacote 1473:2921, foram usados os pacotes:

```
01:34:41.474225 IP willow.39675 > maple.commplex-link: Flags [.], seq 1473:2921, ack 1, win
     115, options [nop,nop,TS val 282136474 ecr 282202089], length 1448
.
.
.
01:34:41.482047 IP maple.commplex-link > willow.39675: Flags [.], ack 2921, win 159,
     options [nop,nop,TS val 282202095 ecr 282136474], length 0
```

O RTT calculado para estes pacotes foi de `RTT1 = 7,822 ms`.

Para o cálculo do RTT do pacote 13057:14505, foram usados os pacotes:

```
01:34:41.474992 IP willow.39675 > maple.commplex-link: Flags [.], seq 13057:14505, ack 1,
     win 115, options [nop,nop,TS val 282136474 ecr 282202090], length 1448
.
.
.
01:34:41.499373 IP maple.commplex-link > willow.39675: Flags [.], ack 14505, win 331,
     options [nop,nop,TS val 282202114 ecr 282136474], length 0
```

O RTT, neste caso, foi `RTT2 = 4.381 ms`.

A diferença entre estes valores se deve ao atraso da máquina maple em começar a enviar os pacotes ACK referentes aos bytes transmitidos.

7 Identifique os procedimentos three-way handshake e connection termination. Coloque as mensagens envolvidas em uma tabela e, para cada um dos procedimentos, inclua a fonte, o destino, o protocolo e informações das mensagens.

Os pacotes do *three-way handshake* são:

```
01:34:41.473518 IP willow.csail.mit.edu.39675 > maple.csail.mit.edu.commplex-link: Flags
     [S], seq 1258159963, win 14600, options [mss 1460,sackOK,TS val 282136473 ecr
     0,nop,wscale 7], length 0
01:34:41.474055 IP maple.csail.mit.edu.commplex-link > willow.csail.mit.edu.39675: Flags
     [S.], seq 2924083256, ack 1258159964, win 14480, options [mss 1460,sackOK,TS val
     282202089 ecr 282136473,nop,wscale 7], length 0
01:34:41.474079 IP willow.csail.mit.edu.39675 > maple.csail.mit.edu.commplex-link: Flags
     [.], ack 1, win 115, options [nop,nop,TS val 282136474 ecr 282202089], length 0
```

Pacote	Fonte	Destino	Protocolo	Informação
1	willow.csail.mit.edu	maple.csail.mit.edu	TCP	SYN
2	maple.csail.mit.edu	willow.csail.mit.edu	TCP	SYN-ACK
3	willow.csail.mit.edu	maple.csail.mit.edu	TCP	ACK

Os pacotes do término da conexão são:

```
01:34:44.339007 IP maple.csail.mit.edu.commplex-link > willow.csail.mit.edu.39675: Flags
     [F.], seq 1, ack 1572890, win 905, options [nop,nop,TS val 282204955 ecr 282139320],
     length 0
01:34:44.339015 IP willow.csail.mit.edu.39675 > maple.csail.mit.edu.commplex-link: Flags
     [.], ack 2, win 115, options [nop,nop,TS val 282139339 ecr 282204955], length 0
```

Pacote	Fonte	Destino	Protocolo	Informação
1	maple.csail.mit.edu	willow.csail.mit.edu	TCP	FIN, ACK 1572890
2	willow.csail.mit.edu	maple.csail.mit.edu	TCP	ACK