Exercício 2: TCPDUMP

NOME	RA
Renan Camargo de Castro	147775

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

Nem todas as interfaces podem ser usadas para capturar dados pelo tcpdump. Para listar as interfaces, deve-se utilizar o comando "sudo tcpdump -D". É bom lembrar que tcpdump -D não funciona como usuário normal. Segue exemplo:

```
➜  ~  ifconfig | expand | cut -c1-8 | sort | uniq -u | awk -F: '{print $1;}'
awdl0
bridge0
en0
en1
en2
gif0
lo0
p2p0
stf0
utun0
➜  ~  tcpdump -D
➜  ~  sudo tcpdump -D
Password:
1.en0
2.awdl0
3.bridge0
4.utun0
5.en1
6.en2
7.p2p0
8.lo0 [Loopback]
➜  ~
```

2. Qual é o endereço IP do nós maple e willow na

rede em questão?

Podemos utilizar o comando do tcpdump, o -n que lista os endereços ips. Para facilitar a leitura, iremos utilizar algumas ferramentas extras:

```
➜  exercicio2 git:(master) ✗ tcpdump -r tcpdump.dat | awk -F: '{print $3;}' | cut
-c14- | sort | uniq | tail -n2
reading from file tcpdump.dat, link-type EN10MB (Ethernet)
maple.csail.mit.edu.commplex-link > willow.csail.mit.edu.39675
willow.csail.mit.edu.39675 > maple.csail.mit.edu.commplex-link
➜  exercicio2 git:(master) ✗ tcpdump -r tcpdump.dat -n | awk -F: '{print $3;}' | c
ut -c14- | sort | uniq | tail -n2
reading from file tcpdump.dat, link-type EN10MB (Ethernet)
128.30.4.222.39675 > 128.30.4.223.5001
128.30.4.223.5001 > 128.30.4.222.39675
➜  exercicio2 git:(master) ✗
```

Podemos ver que os ips são: 128.30.4.222(maple) e 128.30.4.223(willow), que são os únicos que trocam dados nesse dump.

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

Para responder essa pergunta, iremos utilizar o tcpdump com o dump fornecido e filtrar pelos pacotes arp, pedindo para o programa listar os cabeçalhos da camada de link. O comando -e faz isso, como dito no man, o -e: "Print the link-level header on each dump line.". Logo:

```
➜  exercicio2 git:(master) ✗ 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, lengt
h 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
```

Concluimos que os endereços MAC de maple é 00:16:ea:8d:e5:8a, pois ela pediu o endereço mac de willow, que por sua vez possui mac igual a 00:16:ea:8e:28:44.

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? (e.g., as portas até 1024 são chamadas de reserved ports ou well-known ports.)

Podemos utilizar o mesmo comando novamente, o tcpdump com -n. Nesse caso:

```
➜  exercicio2 git:(master) ✗ tcpdump -r tcpdump.dat -n | awk -F: '{print $3;}' | c
ut -c14- | sort | uniq | tail -n2
reading from file tcpdump.dat, link-type EN10MB (Ethernet)
128.30.4.222.39675 > 128.30.4.223.5001
128.30.4.223.5001 > 128.30.4.222.39675
```

Sabemos que as conexões entre eles são usadas das portas:

porta 39675(maple) - não possui associação com nenhuma aplicação conhecida, não faz parte das well-known ports.

porta 5001(willow) - está associada a aplicações como:

- commplex-link
- threat
- backdoorsetup
- socketsdestroie
- trojan

É possível ver que essa porta era comum para vírus e trojans se comunicarem com backdoors.

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 saber quantos kilobytes foram transferidos, analisamos o último ACK antes do encerramento da

conexão. Para facilitar a análise e diminuir a saída do tcpdump, iremos utilizar a saída com o comando:

```
tcpdump -r tcpdump.dat -N > short.txt
```

Analisando o arquivo gerado, vemos que o último ACK válido é o seguinte:

```
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
```

Para saber o pacote correspondente, temos que verificar o pacote que transfere os bits até 1572890 - 1, que é 1572889 (isso no segundo parametro de seq). Achamos o pacote:

```
01:34:44.311921 IP willow.39675 > maple.commplex-link: Flags [FP.], seq 1572017:15
72889, ack 1, win 115, options [nop,nop,TS val 282139311 ecr 282204927], length 872
```

Logo, sabemos que na comunicação foram transmitidos 1572889 bytes. Agora, para sabermos o tempo, basta pegarmos o primeiro pacote e o encerramento da conexão:

```
inicio:
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

fim:
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
```

Temos 01:34:44.339015 - 01:34:41.473518 = 2,865497 segundos

Logo, dividindo, temos: 1572,889 KB / 2,865497 s = 548,906 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?

Identificamos o pacote:

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

Seu ack é de número 2921, logo:

```
01:34:41.482047 IP maple.commplex-link > willow.39675: Flags [.], ack 2921, win 15
9, options [nop,nop,TS val 282202095 ecr 282136474], length 0
```

Temos um RTT de 7,822 ms.

Agora, para o próximo pacote:

```
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
```

Seu ack é de número 14505, logo:

```
01:34:41.499373 IP maple.commplex-link > willow.39675: Flags [.], ack 14505, win 3
31, options [nop,nop,TS val 282202114 ecr 282136474], length 0
```

RTT: 24,381 ms

Esses valores são diferentes, pois provavelmente houve uma demora no envio do ack.

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 e informações das mensagens.

Um pacote de conexão possui a flag S, logo, identificamos o three-way como sendo os pacotes:

```
* 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
* 01:34:41.474055 IP maple.commplex-link > willow.39675: Flags [S.], seq 292408325
6, ack 1258159964, win 14480, options [mss 1460,sackOK,TS val 282202089 ecr 282136
473,nop,wscale 7], length 0
* 01:34:41.474079 IP willow.39675 > maple.commplex-link: Flags [.], ack 1, win 115
, options [nop,nop,TS val 282136474 ecr 282202089], length 0
```

#	Fonte	Destino	Conexão	Flags
1	willow	maple	TCP	S
2	maple	willow	TCP	S,ACK
3	willow	maple	TCP	ACK

Para o connection termination, vemos que são os últimos pacotes com F, que é a tag de Fin, e o ACK da finalização:

```
* 01:34:44.339007 IP maple.commplex-link > willow.39675: Flags [F.], seq 1, ack 15
72890, win 905, options [nop,nop,TS val 282204955 ecr 282139320], 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
```

#	Fonte	Destino	Conexão	Flags
1	maple	willow	TCP	F, ACK
2	willow	maple	TCP	ACK