Aula 11 Snifando com TCPDump

O tcpdump é uma ferramenta utilizada para monitorar e capturar pacotes trafegados em uma rede, ou seja, é um sniffer de rede. Dependendo da sua execução pode desempenhar diversas tarefas entre os quais mostrar os cabeçalhos dos pacotes, tipos de pacotes, origem, destino e no final ainda exibe informações como pacotes capturados, recebido e descartados (dorp).

11.1 Instando tcpdump

É uma ferramenta que faz parte de praticamente todas distribuições Linux. Para verificar se o TCPdump está instalado:

```
$ tcpdump --version
tcpdump version 4.9.3
libpcap version 1.8.1
OpenSSL 1.1.1  11 Sep 2018
```

Caso não esteja instalado, basta executar o comando (para a família Debian):

```
$ sudo apt-get install tcpdump
```

11.2 Usando o tcpdump

Após instalado para verificar o tráfego no host a qual foi instalado executar o comando a seguir, onde:
-i : interface a qual se quer executar o

```
$ sudo tcpdump -i wlo1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
15:53:59.889304 IP ec2-34-208-234-71.us-west-2.compute.amazonaws.com.https > hk.42120: Flags [.], ack 2153043663, win 422, options [nop,nop,TS val 2810400515 ecr 728356549], length 0
15:53:59.901488 IP hk.49700 > _gateway.domain: 63088+ PTR? 103.1.168.192.in-addr.arpa. (44)
15:53:59.923012 IP _gateway.domain > hk.49700: 63088 NXDomain 0/0/0 (44)
1195 packets captured
1808 packets received by filter
613 packets dropped by kernel
```

O tcpdump permite realizar combinações diversas para aplicar diversos tipos de filtros, como por exemplo capturar vários tipos de protocolo, como tcp, ip, ip6 arp e etc., como no exemplo do comando a seguir, onde:
-n: orienta o tcpdump a não resolver nomes, apresentando somente o endereço IP.
icmp: indica o protocolo a ser apresentado na saída do comando, neste caso o protocolo icmp.

```
$ sudo tcpdump -n -i wlo1 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
17:38:41.569194 IP 192.168.1.103 > 195.168.1.1: ICMP echo request, id 8982, seq 1, length 64
17:38:42.589727 IP 192.168.1.103 > 195.168.1.1: ICMP echo request, id 8982, seq 2, length 64
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Também permite salvar a captura dos pacotes em um arquivo com um formato geral ou específico, para ser utilizado para leitura posterior com uma ferramenta genérica ou até mesmo com outras aplicações como por exemplo o Wireshark. O comando a seguir permite que as capturas sejam salvas em arquivo, onde:
-w captura_interface_wlo1.cap: orienta o TCPdump a escrever os pacotes capturados em um arquivo; neste caso, o arquivo captura_wlo1.cap. A extensão não é necessárias em sistemas UnixLike (Unix, Linux e derivados) pois o arquivo é lido não pela sua extensão, mas pele seu tipo, que vem especificado no cabeçalho do arquivo. Assim, a extensão “.cap” é apenas para lembrar que o arquivo é do tipo de captura.

```
$ sudo tcpdump -i wlo1 -w captura_wlo1.cap
tcpdump: listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
2190 packets captured
2206 packets received by filter
0 packets dropped by kernel
```

Pode-se usar comando Linux de redirecionamento para gravar saídas de comandos em arquivos, onde:
> : redirecionador simples, apaga tudo que tem no arquivo e escreve de novo, ou seja, o que estiver no arquivo anteriormente será apgado. Caso se queira registros novos é recomendado usar este;

>> : redirecionador duplo, insere novos registros a partir da última linha do último registro, ou seja, vai acumulando os redirecionamentos. Caso se queira acumular registros, como por exemplo logs, este é recomendado.

Exemplo:

$ sudo tcpdump -i wlo1 > captura_trafego

11.3 Filtrando conteúdos

Para realizar a leitura e análise dos arquivos de captura pode-se utilizar todas as ferramentas do Linux, principalmente comandos de filtragem utilizando Shell Scripts.

O próprio tcpdump realiza leitura e filtragem de arquivos utilizando sempre comandos do Linux, no entanto, também é possível fazer a chamada ao arquivo, sua leitura e filtragem usando apenas comandos Linux.

Um exemplo simples do uso de comando para filtragem de arquivos é o uso do comando “grep” que realiza buscas por ocorrências de strings.

Um exemplo usando o tcpdump para leitura de arquivos é o comando a seguir, onde:
-r : indica ao tcpdump para realizar a leitura doarquivo indicado.

$ sudo tcpdump -r captura_interface_wlo1.cap | grep http
17:50:48.178503 IP 177.43.70.25.static.host.gvt.net.br.**https** > hk.58398: Flags [.], seq 2302096:2303536, ack 3036, win 1422, options [nop,nop,TS val 426414685 ecr 1705838865], length 1440
17:50:48.178513 IP 177.43.70.25.static.host.gvt.net.br.**https** > hk.58398: Flags [P.], seq 2303536:2303759, ack 3036, win 1422, options [nop,nop,TS val 426414687 ecr 1705838865], length 223
17:50:48.180510 IP hk.58398 > 177.43.70.25.static.host.gvt.net.br.**https**: Flags [.], ack 2303759, win 24576, options [nop,nop,TS val 1705838920 ecr 426414685], length 0

Um exemplo seria executar o comando para captura de pacotes e depois filtrar por pacotes TCP salvos em um arquivo chamado captura_wlo1.cap:

$ sudo tcpdump -i wlo1 -w captura_trafegocat

A seguir, para filtrar os dados coletados pode-se utilizar tanto o tcpdump:

$ cat captura_trafego | grep pass
20:38:42.970748 IP hk.39991 > _gateway.domain: 56629+ A? passwordsleakcheck-pa.googleapis.com. (54)