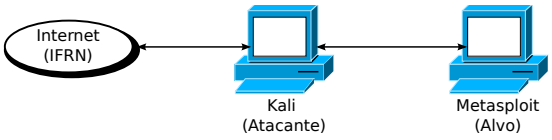
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Professor: Macêdo Firmino
Disciplina: Segurança de Computadoores
Prática 06: Ferramenta TCPDump

Olá turma, hoje iremos conhecer uma ferramenta de analisador de tráfego que roda em terminal chamado de tcpdump. Ele ele é ideal para rodar em servidores remotos ou dispositivos para os quais não há interface gráfica disponível (como é o nosso caso), para coletar dados que possam ser analisados posteriormente. Vamos lá preparados??

Configurando o Ambiente

Para estudarmos estes conceitos e ferramentas, e para as aulas posteriores, iremos utilizar duas máquinas virtuais (Kali Linux e Metasploit). A Metasploit será a máquina que iremos utilizar como alvo e o Kali Linux será utilizado para gerarmos os ataques.



Tcpdump

O tcpdump é um programa analisador de tráfego (*sniffer*) em linha de comando e software livre. Ele permite que o usuário exiba pacotes de diversos protocolos TCP/IP. O Tcpdump funciona na maioria dos sistemas operacionais do tipo Linux, Solaris , BSDs e macOS. Nesses sistemas, o tcpdump usa a biblioteca libpcap para capturar pacotes. O tcpdump para Windows é chamada de WinDump, ele usa WinPcap, que é a versão do libpcap para Windows.

O Tcpdumo é uma ferramenta poderosa e versátil que inclui muitas opções e filtros. O Tcpdump está incluído em várias distribuições do Linux. Verifique se o tcpdump está instalado em seu sistema com o seguinte comando para localiza-lo:

```
which tcpdump
```

Se o tcpdump não estiver instalado, você pode instalá-lo, mas usando o gerenciador de pacotes da sua distribuição. Por exemplo:

```
sudo apt install tcpdump
```

Capturando Pacotes

Para capturar pacotes o tcpdump requer permissões de root, portanto, nos exemplos a seguir, a maioria dos comandos são utilizados com o sudo.

Para começar, use o comando tcpdump –list-interfaces (ou -D) para mostrar quais interfaces estão disponíveis para captura:

```
sudo tcpdump -D
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -D
[sudo] password for kali:
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.docker0 [Up, Disconnected]
5.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
6.nflog (Linux netfilter log (NFLOG) interface) [none]
7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
8.dbus-system (D-Bus system bus) [none]
9.dbus-session (D-Bus session bus) [none]
```

No exemplo acima, você pode ver todas as interfaces disponíveis em minha máquina. Quando se executa o tcpdump sem passar nenhum parâmetro o Linux captura, por default, os pacotes presentes na eth0.

Existe uma interface especial chamada de "any" permite a captura em todas as interface ativa. Vamos usá-lo para começar a capturar alguns pacotes. Capture todos os pacotes em qualquer interface executando este comando:

```
sudo tcpdump -i any
```

O Tcpdump continua a capturar pacotes até receber um sinal de interrupção. Você pode interromper a captura pressionando [Ctrl+C].

Se quisermos capturar um número limitado de pacotes, podemos utilizar a opção -c, indicando o número de pacotes que desejamos capturar. Por exemplo:

```
sudo tcpdump -i eth0 -c 10
```

Tcpdump é capaz de capturar e decodificar muitos protocolos diferentes, como TCP, UDP, ICMP e muitos mais. Por exemplo, iremos analisar um segmento TCP capturado pelo tcpdump que tem a seguinte aparência:

```
20:48:44.259152 IP 10.0.2.15.57816 >
200.137.1.195.https: Flags [S], seq
1938058989, win 64240, options [...],
length 0
```

O primeiro campo 20:48:44.259152, representa o timestamp do pacote recebido de acordo com o relógio local. A opção -t faz com que o tcpdump não imprima esta marca de tempo.

Em seguida, temos o endereço IP de origem do segmento (10.0.2.15) e sua respectiva porta de origem (57816). Na sequência temos o endereço IP de destino (200.137.1.195), com seu respectivo valor de porta de destino (https). O tcpdump quando possível converte via DNS o endereço IP e o nome da porta. Para evitar esta conversão podemos utilizar a opção -n.

Após a origem e o destino, você pode encontrar os TCP Flags [S.]. Os valores típicos para este campo incluem: A (ACK), R (RST), S (SYN), F (FIN), U (URG) e P (PSH). Quando no flag temos o ponto [.], significa uma mensagem ACK.

Em seguida é o número de sequência dos dados contidos no pacote. Ele representa o número do primeiro byte de dados deste segmento tcp. Neste exemplo, a sequência é seq 1.938.058.989. Ele é utilizado para reordenar os pacotes no receptor.

Depois surge o tamanho da janela (64240), que representa o número de bytes disponíveis no buffer de recebimento, seguido pelas opções do TCP.

Por fim, temos o comprimento do pacote (0), que representa o comprimento, em bytes, dos dados do payload. Como o exemplo mostrado é de estabelecimento de conexão, não foi enviado dados e o comprimento foi de zero.

Filtrando Pacotes

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.

Um dos recursos mais poderosos do tcpdump é sua capacidade de filtrar os pacotes capturados usando uma variedade de parâmetros, como endereços IP de origem e destino, portas, protocolos etc. Filtrando os pacotes iremos restringir os resultados e facilitar a solução de problemas específicos.

Podemos filtrar pacotes baseados no:

Protocolo

Podemos especificar um protocolo na linha de comando. Por exemplo, capturando pacotes icmp, tcp, udp, ip, arp, http, ftp, dns, entre outros.

Por exemplo, se desejarmos filtrar somente as mensagens icmp podemos utilziar o filtro:

```
sudo tcpdump icmp
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:14:37.059724 IP 10.0.2.15 > 10.0.2.4: ICMP echo request, id 62070, seq 1, length 64
21:14:37.060102 IP 10.0.2.4 > 10.0.2.15: ICMP echo reply, id 62070, seq 1, length 64
21:14:38.086400 IP 10.0.2.15 > 10.0.2.4: ICMP echo request, id 62070, seq 2, length 64
21:14:38.086798 IP 10.0.2.4 > 10.0.2.15: ICMP echo reply, id 62070, seq 2, length 64
21:14:39.107041 IP 10.0.2.15 > 10.0.2.4: ICMP echo request, id 62070, seq 3, length 64
21:14:39.107508 IP 10.0.2.4 > 10.0.2.15: ICMP echo reply, id 62070, seq 3, length 64
^C
```

Endereço IP (host)

Podemos limitar a captura apenas a pacotes relacionados a um host (IP) específico. Por exemplo, se desejarmos filtrar somente as mensagens IP do host (200.137.1.195) podemos utilziar o filtro:

```
sudo tcpdump host 200.137.1.195
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump host 200.137.1.195
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:30:03.183336 IP 10.0.2.15.33044 > 200.137.1.195.https: Flags [S], seq 111855400, win 642
40, options [mss 1460,sackOK,TS val 326431232 ecr 0,nop,wscale 7], length 0
21:30:03.189460 IP 200.137.1.195.https > 10.0.2.15.33044: Flags [S.], seq 114052, ack 11185
5401, win 32768, options [mss 1460], length 0
21:30:03.189502 IP 10.0.2.15.33044 > 200.137.1.195.https: Flags [.], ack 1, win 64240, leng
th 0
21:30:03.193976 IP 10.0.2.15.33044 > 200.137.1.195.https: Flags [P.], seq 1:665, ack 1, win
 64240, length 664
21:30:03.200360 IP 200.137.1.195.https > 10.0.2.15.33044: Flags [P.], seq 1:246, ack 665, w
in 32104, length 245
```

Neste caso, o tcpdump captura tanto pacotes de origem como de destino para este determinado IP. Entretanto, podemos especificar de onde e para onde vão os pacotes que queremos capturar, através das opções: src e dst. Por exemplo, se queremos capturar somente os pacotes que tem destino o endereço 200.137.1.195 e origem 10.0.2.15, basta aplicar um dos filtros abaixo:

```
sudo tcpdump dst 200.137.1.195 and src 10.0.2.15
```

Endereço de Rede

Podemos filtrar a captura apenas para pacotes relacionados a uma determinada rede (net). Por exemplo, se desejarmos filtrar somente as mensagens de hosts pertencentes a rede (10.0.2.0/24) podemos utilizar o filtro:

```
sudo tcpdump net 10.0.2.0/24
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump net 10.0.2.0/24
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:38:47.185407 IP 10.0.2.4 > 10.0.2.15: ICMP echo request, id 42010, seq 1, length 64
21:38:47.185474 IP 10.0.2.15 > 10.0.2.4: ICMP echo reply, id 42010, seq 1, length 64
21:38:47.236037 IP 10.0.2.15.43676 > 10.0.2.1.domain: 28245+ PTR? 15.2.0.10.in-addr.arpa. (
40)
21:38:47.249471 IP 10.0.2.1.domain > 10.0.2.15.43676: 28245 NXDomain 0/0/0 (40)
21:38:47.249920 IP 10.0.2.15.49762 > 10.0.2.1.domain: 14923+ PTR? 4.2.0.10.in-addr.arpa. (3
9)
21:38:47.263475 IP 10.0.2.1.domain > 10.0.2.15.49762: 14923 NXDomain 0/0/0 (39)
21:38:47.339316 IP 10.0.2.15.48427 > 10.0.2.1.domain: 35790+ PTR? 1.2.0.10.in-addr.arpa. (3
9)
21:38:47.350706 IP 10.0.2.1.domain > 10.0.2.15.48427: 35790 NXDomain 0/0/0 (39)
```

Endereço de Portas

Podemos filtrar a captura apenas para pacotes relacionados a uma determinada porta (port). Por exemplo, se desejarmos filtrar somente as mensagens que tem como origem ou destino a porta 443 (https), podemos utilizar o filtro:

```
sudo tcpdump port https
```

ou

```
sudo tcpdump port 443
```

Se desejarmos filtrar a captura para os pacotes destinados ou de origem a uma determinada porta podemos utilizar a opção: dst e src, respectivamente. Por exemplo, capturar os pacotes destinados a porta 23, temos:

```
tcpdump dst port 23
```

ou, de origem na porta 23, temos

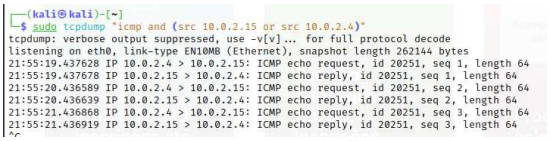
```
tcpdump src port 23
```

Você também pode combinar filtros usando os operadores lógicos “and” e “or” para criar expressões mais complexas. Por exemplo, para filtrar pacotes de origem do host 192.168.122.98 e na porta do serviço HTTP, usamos o comando:

```
sudo tcpdump src 192.168.122.98 and port 80
```

Podemos ainda, criarmos expressões mais complexas agrupando filtros com parênteses. Nesse caso, coloque toda a expressão de filtro entre aspas para evitar que o shell as confunda com expressões do shell. Por exemplo, para filtrarmos todos os pacotes icmp que se originaram no endereço IP 10.0.2.15 ou 10.0.2.4, temos:

```
sudo tcpdump "icmp and (src 10.0.2.15 or
 src 10.0.2.4)"
```



Salvando Capturas

Outro recurso útil fornecido pelo tcpdump é a capacidade de salvar a captura em um arquivo para analisar os resultados posteriormente. Para salvar os pacotes em um arquivo em vez de exibi-los na tela, use a opção -w e o nome do arquivo com a extensão pcap. Por exemplo, o comando abaixo salva a saída da captura para um arquivo chamado file.pcap.

```
sudo tcpdump -w file.pcap
```

O tcpdump cria um arquivo em formato binário. Para ler o conteúdo do arquivo, execute tcpdump com a opção -r. Como você não está mais capturando os pacotes diretamente da interface de rede, sudonão é necessário ler o arquivo. Por exemplo:

```
tcpdump -r file.pcap
```

É possível utilizarmos qualquer um dos filtros que discutimos para filtrar o conteúdo do arquivo, assim como faria com dados em tempo real. Além disso, podemos utilizar o Wireshark para ler os arquivos (.pcap) capturados pelo tcpdump.

Atividades (Entregar no Google Sala de Aula)

1. Utilize o netcat, nas máquinas Kali e Metasploit, para estabelecer uma conexão, enviar dados e finalizar conexão. Ao mesmo tempo faça uso do tcpdump para capturar as informações trocadas entre as duas máquinas. Depois faça um relatório mostrando quais foram as mensagens trocadas e quais informações foram enviadas nas respectivas mensagens. Explicando o passo a passo do estabelecimento da conexão até o término.

2. Utilize o Nmap no Kali para determinar se a porta 80 (http) está aberta no Metasploit. Ao mesmo tempo utilize o tcpdump para capturar os pacotes trocados entre as máquinas. Depois faça um relatório mostrando quais foram as mensagens trocadas e quais informações foram enviadas nas respectivas mensagens. Explicando como o Nmap determina como uma porta estaria aberta.

   ```
   sudo nmap -sS -p 80 10.0.2.4
   ```

3. Faça o mesmo do item 2, porém utilizando o Nmap para tentar determinar o software, e sua respectiva versão, que está rodando na porta 80. Faça um relatório mostrando as mensagens trocadas.

   ```
   sudo nmap -sV -p 80 10.0.2.4
   ```