Trabalho Final
Monitoramento e Filtragem de Fluxos

Conrado Boeira[1]

[1]Laboratório de Redes de Computadores
Redes de Computadores Avançadas
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

1. Introdução

Mesmo dentro de uma rede local, existem diversos fluxos de pacotes de rede que os usuários não tem conhecimento. Aplicativos enviando mensagens para servidores remotos e pacotes enviados de dispositivos como televisões inteligentes são alguns exemplos de fluxos que normalmente não recebem muita atenção. Porém, eles ainda podem afetar o funcionamento da rede e levar a problemas. Dessa maneira, é importante ter uma ferramenta para realizar o monitoramento desses fluxos e bloquear-los quando for necessário. Neste relatório é apresentada uma ferramenta desenvolvida usando a linguagem Python 3.8.6 e a biblioteca Socket Raw, que realiza justamente essa função, monitora todos pacotes IPv4 recebidos e permite que o usuário filtre fluxos que desejar.

Para executar a ferramenta basta usar o comando:

```
sudo python flowall.py <MAC roteador> <Lista de monitorados>
```

Onde, <MAC roteador> é o endereço físico do roteador com acesso a rede externa para qual o programa deve encaminhar os pacotes e <Lista de monitorados> é uma lista de endereços MAC separados por uma vírgula que define os hosts que devem ter seus fluxos monitorados.

Na Secção 2, é apresentada a topologia usada para o desenvolvimento e teste da ferramenta. Na Seção 3 e 4 é explicado o funcionamento da coleta de dados e do processo de filtro dos fluxos, respectivamente. Finalmente, na Secção 5 são apresentadas as considerações finais.

2. Topologia

No desenvolvimento da ferramenta, foi utilizada um ambiente que consiste em duas máquinas virtuais (VM), uma máquina host que recebe os pacotes das VMs, executa a ferramenta desenvolvida e pode encaminhar os pacotes para um roteador local com acesso a Internet. Um esquema da topologia usada pode ser visto na Figura 1.

Ambas VMs tem estaticamente definida a sua interface de rede (criada no modo bridge de maneira a permitir acesso à rede local) de maneira a ter sua rota default configurada para o IP da host. Dessa maneira, qualquer pacote enviado pelas VMs precisa ser processado pela ferramenta desenvolvida. Além disso, foi desativada a interface IPv6 das VMs de modo a obrigar toda comunicação a ser feita através do protocolo IPv4, que é monitorado no host.

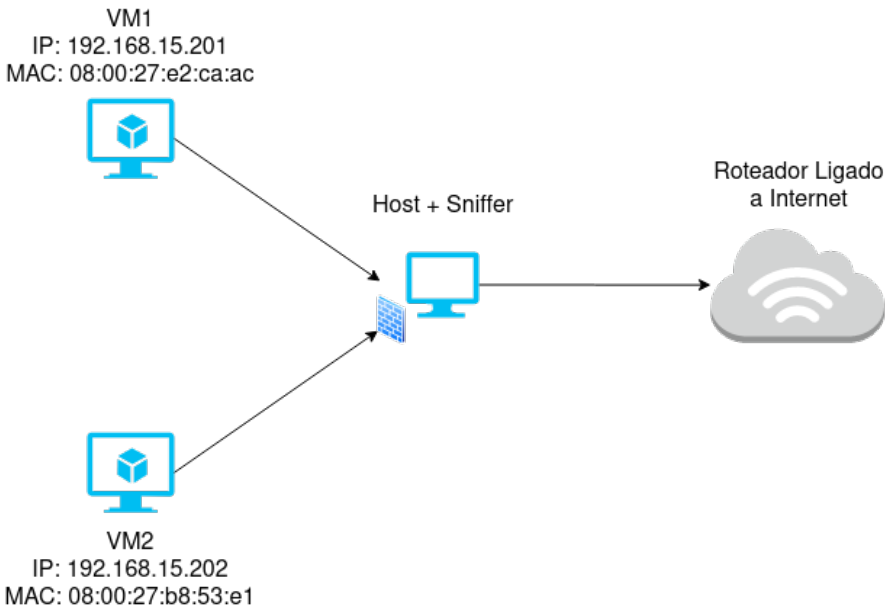


Figura 1. Topologia utilizada para o desenvolvimento e o teste da ferramenta.

3. Monitorador de Fluxos

A primeira parte da ferramente consiste na coleta de dados de todos fluxos que passam através da rede e a compilação destes dados de maneira a permitir ao usuário uma fácil visualização através de uma interface visual. Dessa maneira, primeiramente são coletados dados do endereço físico do emissor através do header do protocolo Ethernet. Isso é feito para que se possa comparar o endereço origem do pacote com a lista de endereços que devem ser monitorados. Caso seja o caso do endereço estar na lista, o pacote é destrinchado mais a fundo, sendo coletado dados sobre o endereço IPv4 de origem e destino, o protocolo utilizado e o número das portas de origem e destino caso estas existam. As informações são armazenadas e então o pacote é encaminhado para o roteador, dado que ele passe pelo filtro descrito na seção seguinte.

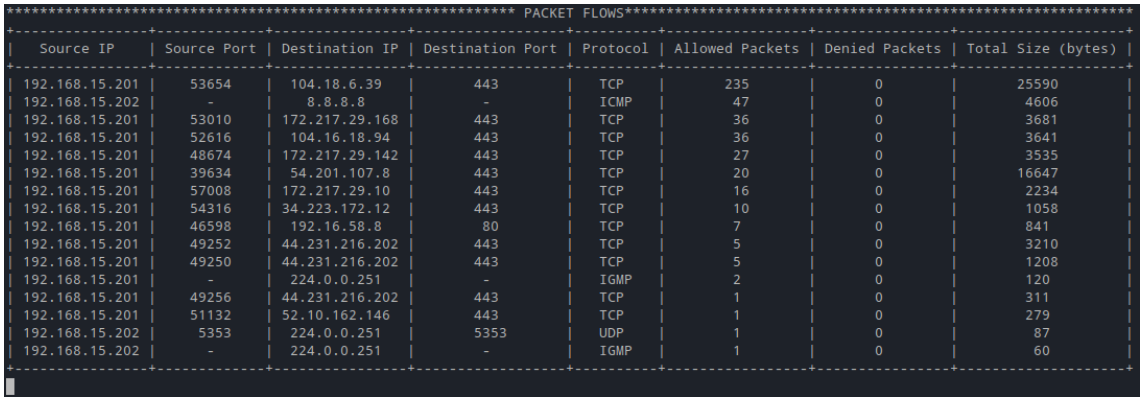
A Figura 2 demonstra a interface de usuário criada. Nela é possível ver os diversos fluxos, tanto como o número de pacotes encaminhados e bloqueados e a quantidade total de bytes recebidos. Nos resultados mostrados, uma das VMs está acessando o site *moodle.pucrs.br* enquanto a outra realiza um ping para o endereço 8.8.8.8 .

4. Regras de Filtragem

A qualquer momento durante a execução, o usuário tem acesso a três possíveis comandos: *show rules*, *deny* e *allow*. O primeiro destes exibe as regras atualmente ativas, sendo que inicialmente não existe nenhuma regra. O comando *deny* permite que o operador defina uma regra de bloqueio seguindo a sintaxe a seguir:

```
: deny <IP origem> <Porta origem> <IP destino> <Porta destino>
```

Além de poder fornecer os detalhes específicos de um fluxo assim, também é possível passar o caractere * em qualquer um dos campos para sinalizar que ele pode assumir qualquer valor. No casos dos protocolos que não usam portar, pode ser fornecido o número **-1** como identificador das portas.



PACKET FLOWS

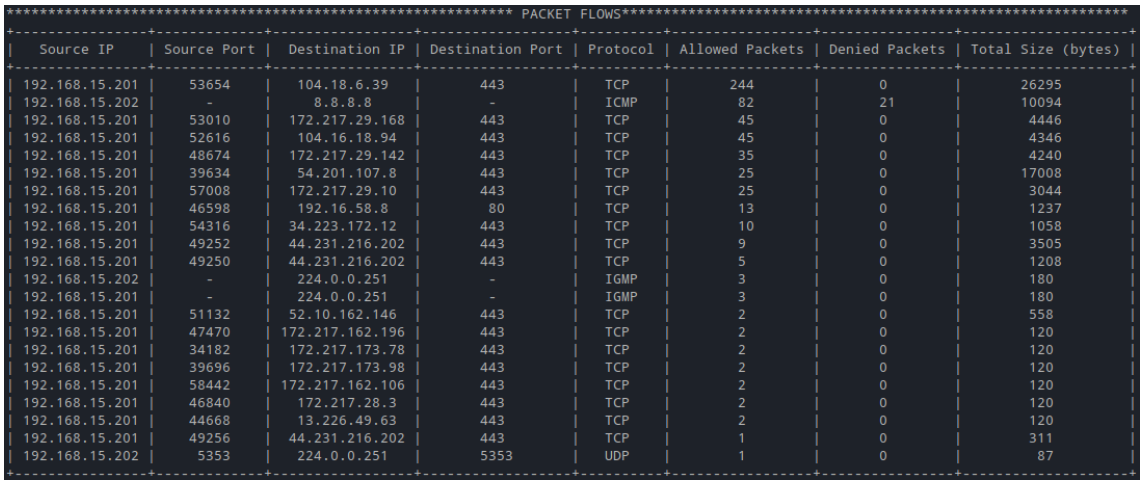
Source IP	Source Port	Destination IP	Destination Port	Protocol	Allowed Packets	Denied Packets	Total Size (bytes)
192.168.15.201	53654	104.18.6.39	443	TCP	235	0	25590
192.168.15.202	-	8.8.8.8	-	ICMP	47	0	4606
192.168.15.201	53010	172.217.29.168	443	TCP	36	0	3681
192.168.15.201	52616	104.16.18.94	443	TCP	36	0	3641
192.168.15.201	48674	172.217.29.142	443	TCP	27	0	3535
192.168.15.201	39634	54.201.107.8	443	TCP	20	0	16647
192.168.15.201	57008	172.217.29.10	443	TCP	16	0	2234
192.168.15.201	54316	34.223.172.12	443	TCP	10	0	1058
192.168.15.201	46598	192.16.58.8	80	TCP	7	0	841
192.168.15.201	49252	44.231.216.202	443	TCP	5	0	3210
192.168.15.201	49250	44.231.216.202	443	TCP	5	0	1208
192.168.15.201	-	224.0.0.251	-	IGMP	2	0	120
192.168.15.201	49256	44.231.216.202	443	TCP	1	0	311
192.168.15.201	51132	52.10.162.146	443	TCP	1	0	279
192.168.15.202	5353	224.0.0.251	5353	UDP	1	0	87
192.168.15.202	-	224.0.0.251	-	IGMP	1	0	60

Figura 2. Interface da ferramenta exibindo o fluxo de pacotes observados.

A Figura 3 monstra o que acontece, no mesmo caso apresentado no exemplo da seção anterior, após usar o comando:

```
: deny * * 8.8.8.8 *
```

Como é possível ver, o fluxo ICMP criado pela VM de endereço 192.168.15.202 é bloqueado, levando a uma estagnação no número de pacotes encaminhados e um aumento no total de pacotes bloqueados.



PACKET FLOWS

Source IP	Source Port	Destination IP	Destination Port	Protocol	Allowed Packets	Denied Packets	Total Size (bytes)
192.168.15.201	53654	104.18.6.39	443	TCP	244	0	26295
192.168.15.202	-	8.8.8.8	-	ICMP	82	21	10094
192.168.15.201	53010	172.217.29.168	443	TCP	45	0	4446
192.168.15.201	52616	104.16.18.94	443	TCP	45	0	4346
192.168.15.201	48674	172.217.29.142	443	TCP	35	0	4240
192.168.15.201	39634	54.201.107.8	443	TCP	25	0	17008
192.168.15.201	57008	172.217.29.10	443	TCP	25	0	3044
192.168.15.201	46598	192.16.58.8	80	TCP	13	0	1237
192.168.15.201	54316	34.223.172.12	443	TCP	10	0	1058
192.168.15.201	49252	44.231.216.202	443	TCP	9	0	3505
192.168.15.201	49250	44.231.216.202	443	TCP	5	0	1208
192.168.15.202	-	224.0.0.251	-	IGMP	3	0	180
192.168.15.201	-	224.0.0.251	-	IGMP	3	0	180
192.168.15.201	51132	52.10.162.146	443	TCP	2	0	558
192.168.15.201	47470	172.217.162.196	443	TCP	2	0	120
192.168.15.201	34182	172.217.173.78	443	TCP	2	0	120
192.168.15.201	39696	172.217.173.98	443	TCP	2	0	120
192.168.15.201	58442	172.217.162.106	443	TCP	2	0	120
192.168.15.201	46840	172.217.28.3	443	TCP	2	0	120
192.168.15.201	44668	13.226.49.63	443	TCP	2	0	120
192.168.15.201	49256	44.231.216.202	443	TCP	1	0	311
192.168.15.202	5353	224.0.0.251	5353	UDP	1	0	87

Figura 3. Exemplo de execução onde é usada uma regra de bloqueio para fluxos com destino ao IP 8.8.8.8.

O comando *allow* funciona com a mesma sintaxe que o *deny*. Porém, ele realiza o processo inverso. Ele olha pela lista de regras e, caso exista uma regra definida exatamente igual aos parâmetros recebidos, ele apaga ela, assim permitindo que o fluxo definido volta a ser encaminhado.

5. Conclusão

Neste trabalho foi apresentada uma ferramenta para monitoramento da rede local. Ela permite que o usuário veja, em tempo real, dado dos fluxos de pacotes e possa bloqueá-los com facilidade.