



## Projeto 2 Segurança e Confiabilidade 22/23

### Parte I: *iptables*

Grupo SegC-003

Temos como máquina segura onde corre o servidor **TintolmarketServer** a máquina MServer, vamos então ver os comando executados para configurar a mesma de modo a satisfazer as restrições e serviços pedidos.

Executámos os seguintes comandos na máquina MServer:

- **Preparação**

1)

```
$ sudo /sbin/iptables -P OUTPUT DROP
```

```
$ sudo /sbin/iptables -P INPUT DROP
```

Estes comandos definem a política de INPUT e OUTPUT como DROP (bloqueio). Isto significa que todas as conexões de entrada e saída serão bloqueadas a menos que seja criada uma regra específica que diga o contrário. Isto inclui **ping** e **ssh**, que como esperado não funcionam depois da execução destes comandos. Para testar isto basta tentar fazer **ping** no MServer de qualquer máquina.

2)

```
$ sudo /sbin/iptables -A INPUT -i lo -j ACCEPT
```

```
$ sudo /sbin/iptables -A OUTPUT -o lo -j ACCEPT
```

Temos também os comandos que adicionam a regra que permite o tráfego proveniente da interface de *loopback* (lo). Esta interface permite que haja comunicação entre membros do mesmo sistema. Ou seja, por exemplo, sem estes comandos não é possível realizar um **ping** ao próprio sistema.

3)

```
$ sudo /sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
$ sudo /sbin/iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -m conntrack ! --ctorigsrc 10.101.204.4 -j ACCEPT
```

Por último na preparação, temos ainda estes comandos para ambas as *chains* de INPUT e OUTPUT que permitem conexões que já estão **estabelecidas** ou **relacionadas**. Conexões **estabelecidas** são conexões que foram iniciadas localmente com sucesso. Conexões **relacionadas** são aquelas que estão associadas a conexões estabelecidas, como respostas a solicitações ou relacionadas a um protocolo específico.

- **Implementação das Regras pedidas no MServer**

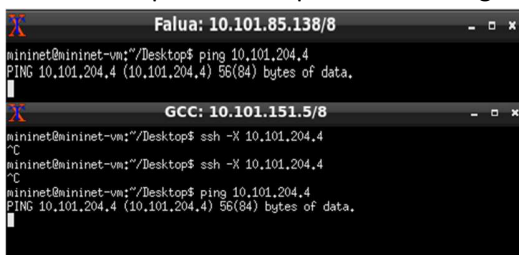
(i) Responde a *pings* apenas com origem na máquina **gcc** (10.101.151.5) e nas máquinas da sub-rede 10.101.85.0/24;

Para implementar esta regra escolhemos usar os seguintes comandos:

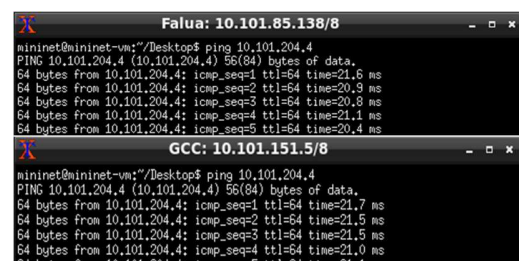
```
$ sudo /sbin/iptables -A INPUT -p icmp --icmp-type echo-request
-s 10.101.151.5 -j ACCEPT

$ sudo /sbin/iptables -A INPUT -p icmp --icmp-type echo-request
-s 10.101.85.0/24 -j ACCEPT
```

Estes comandos especificam que nos referimos ao protocolo ICMP com a opção `-p ICMP` e mais especificamente aos *pings* do tipo `-icmp-type echo-request` só aceitando os *pings* recebidos da máquina **gcc** e da **sub-rede** pois previamente usámos o comando `$ sudo /sbin/iptables -P OUTPUT DROP` que dá DROP a todas as conexões exceptuando as que esta nova regra permite. Podemos testar isto com o método abaixo.



1. Antes de executar os comandos é impossível fazer *ping* no MServer através das máquinas da sub-rede e da máquina gcc



2. Depois de inseridos os comandos já conseguimos fazer *ping*

(II) Aceita ligações de clientes Tintolmarket de qualquer origem para o servidor TintolmarketServer;

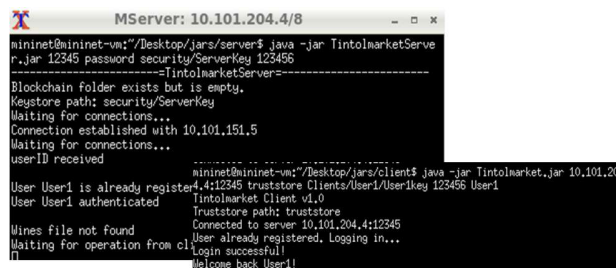
Para implementar esta regra usámos o seguinte comando:

```
$ sudo /sbin/iptables -A INPUT -p tcp --dport <porta do
servidor> -j ACCEPT
```

Este comando limita o tipo de tráfego que é aceite puramente a TCP mas apenas na porta especificada no comando. Neste caso usaremos a porta **12345** porque foi a que foi definida como *default* no Projeto 1 da UC. Para testar isto vamos tentar ligar-nos ao servidor antes e depois de termos executado o comando.



3. Com o servidor Tintolmarket a correr no MServer não é possível ser feita a ligação do cliente a este porque ainda não definimos a regra que aceitará esta ligação



4. Nesta imagem já é possível fazer a ligação porque o servidor já aceita conexões por TCP na porta especificada (ex:12345)

(III) Aceita ligações ssh apenas da máquina gcc e das sub-redes em que se encontram as máquinas DC1, DC2 e DC3 (com máscara 255.255.255.224).

Para implementar esta regra usámos os seguintes comandos:

```
$ sudo /sbin/iptables -A INPUT -p tcp --dport 22 -s 10.101.151.5 -j ACCEPT

$ sudo /sbin/iptables -A INPUT -p tcp --dport 22 -s 10.121.52.0/27 -j ACCEPT

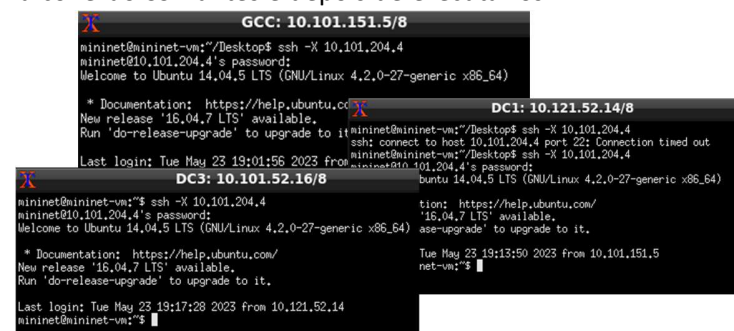
$ sudo /sbin/iptables -A INPUT -p tcp --dport 22 -s 10.101.52.16/27 -j ACCEPT
```

Estes comandos implementam regras que permitem o tráfego SSH (protocolo TCP, porta 22) vindo das **sub-redes** (2º e 3º) mencionadas e da máquina **gcc** (1º) com a máscara pedida. Para testarmos isto vamos tentar realizar uma conexão SSH antes e depois de executar os comandos.



Two terminal windows are shown. The first, titled 'DC1: 10.121.52.14/8', shows a user at 'mininet@mininet-vm' attempting to connect via SSH to '10.101.204.4'. The second, titled 'DC3: 10.101.52.16/8', shows the user navigating to the desktop and then attempting the same SSH connection, which fails.

5. Com o servidor SSH aberto em MServer as máquinas DC1 e DC3 não conseguem fazer a conexão. Iriamos verificar o mesmo na máquina gcc.



Two terminal windows are shown. The first, titled 'GCC: 10.101.151.5/8', shows a user at 'mininet@mininet-vm' successfully connecting via SSH to '10.101.204.4'. The second, titled 'DC3: 10.101.52.16/8', shows a user at 'mininet@mininet-vm' successfully connecting via SSH to '10.101.204.4'.

6. Com o servidor SSH aberto em MServer já conseguimos efetuar as ligações SSH como esperado.

(IV) A máquina segura onde corre o servidor **TintolmarketServer** (MServer) apenas pode fazer:

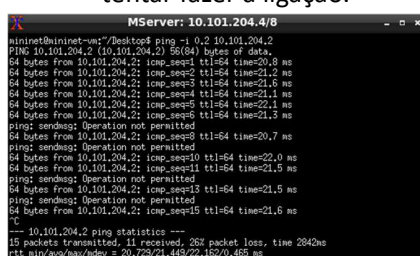
- ✓ Ping às máquinas da sua sub-rede local. Contudo, a frequência dos pings deve ser limitada a um máximo de 3 pings/segundo;
- ✓ SSH à máquina gcc.

Para implementar estas regras usámos os seguintes comandos:

```
$ sudo /sbin/iptables -A OUTPUT -p icmp --icmp-type echo-request -m limit --limit 3/s --limit-burst 3 -d 10.101.204.0/24 -j ACCEPT

$ sudo /sbin/iptables -A OUTPUT -p tcp --dport 22 -d 10.101.151.5 -j ACCEPT
```

A primeira regra permite o envio de pacotes ICMP, ou seja, *pings* para a sub-rede local, com o limite pedido de 3 *pings* por segundo. A segunda permite que a máquina segura estabeleça ligações TCP na porta 22, ou seja, ligações SSH com a máquina gcc (10.101.151.5). Para testarmos a primeira iremos ver se fazendo *pings* com um intervalo menor que 33 milissegundos (tempo máximo para ter 3 por segundo) temos alguns que são impedidos. Já para a segunda, basta abrir o servidor SSH na máquina gcc e tentar fazer a ligação.



A terminal window titled 'MServer: 10.101.204.4/8' shows the output of a 'ping -i 0.2 10.101.204.2' command. It displays a series of ping results with times ranging from 20.8 ms to 21.6 ms. At the bottom, it shows a summary: '15 packets transmitted, 11 received, 26% packet loss, time 2842ms'.

7. Testando o primeiro comando com um ping à máquina MClient com um intervalo de 2 milissegundos. Como vemos à esquerda, há alguns que são bloqueados pois violariam a regra dos 3 pings por segundo. O valor máximo seria 3.3 milissegundos.



Two terminal windows are shown. The first, titled 'GCC: 10.101.151.5/8', shows a user at 'mininet@mininet-vm' successfully connecting via SSH to '10.101.151.5'. The second, titled 'MServer: 10.101.204.4/8', shows a user at 'mininet@mininet-vm' successfully connecting via SSH to '10.101.151.5'.

8. Depois de executar os comandos a cima já conseguimos realizar a conexão SSH.