Integrando IPTables ao RouteFlow Server

Dorival M. Machado Junior
Universidade Federal de São Carlos
SP 310 - km 235
São Carlos - Brasil
dorivaljunior@gmail.com

Viviani Akemi Kasahara
Universidade Federal de São Carlos
SP 310 - km 235
São Carlos - Brasil
vivi_akemi@yahoo.com.br

Renata Rodrigues de Oliveira
Universidade Federal de São Carlos
SP 310 - km 235
São Carlos - Brasil
professoraxrenata@gmail.com

Pablo Botton da Costa
Universidade Federal de São Carlos
SP 310 - km 235
São Carlos - Brasil
pablo.botton.costa@gmail.com

Rogers S. Cristo
Universidade Federal de São Carlos
SP 310 - km 235
São Carlos - Brasil
krigisk@gmail.com

Emerson Barea
Universidade Federal de São Carlos
SP 310 - km 235
São Carlos - Brasil
emerson.barea@gmail.com

1. INTRODUÇÃO

RouteFlow é um projeto de livre cooperação criado para prover roteamento de IPs a redes definidas por software do tipo OpenFlow. Da mesma maneira que o OpenFlow, o RouteFlow não possui camada física, já que virtualiza os switchs que trabalham com OpenFlow e emula seu funcionamento em máquinas virtuais [1].

Apesar de possuir realizar com eficiência a tarefa de roteamento, o RouteFlow possui uma lacuna no contexto de segurança. Em sua estrutura de código não há serviço próprio ou auxiliar que realize o monitoramento de IPs, necessário para realizar bloqueios em uma eventual tentativa de invasão.

Este trabalho apresenta uma solução de alteração ao RouteFlow onde a implementação de segurança torna-se possível através da inserção de regras IPTables.

2. ESTRUTURA ROUTEFLOW

O cenário básico da estrutura RouteFlow é composto por um controlador OpenFlow, neste caso chamado de RFProxy, um servidor RouteFlow (RFServer) e um conjunto de máquinas virtuais que reproduzem a rede física e suas respectivas conexões [1].

A estrutura de código é dividida em alguns arquivos principais.A Figura 1 ilustra o fluxo de comunicação entre estes arquivos. Além disso, é demonstrado abaixo uma explicação sucinta de cada arquivo pertencente a esta estrutura.

- **FlowTable.cc** - responsável por detectar rotas no ambiente virtual.
- **rfclient.cc** - responsável por inicializar um rfclient (switch do plano virtual – vm lxc).
- **rfserver.py** - servidor que recebe as mensagens dos rfclients.
- **rfproxy.py** – controlador responsável pela comunicação com as máquinas do plano físico.
- **RFProtocol** – responsável por definições de tipos de mensagem IPC para comunicação entre os componentes do RouteFlow.

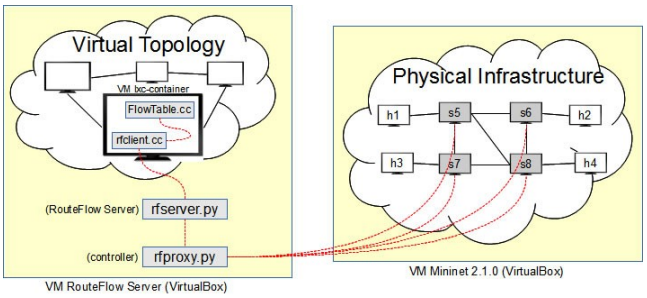

Figura 1. Estrutura de código RouteFlow.

3. INTEGRAÇÃO IPTABLES

O IPTables é uma ferramenta de edição da tabela de filtragem de pacotes, ou seja, ele é capaz de analisar o cabeçalho (header) e tomar decisões sobre os destinos destes pacotes [2].

Neste trabalho a geração de regras IPTables é feita manualmente, porém é possível inserir regras de outras formas, como:

a) Através da linha de comando em si, caso em que o administrador insere diretamente a regra via terminal.

b) Na inicialização, através de script de firewall, onde as regras são previamente definidas e inseridas em um script a ser executado durante a inicialização da máquina virtual ou quando solicitado em algum momento.

c) Através de uma ferramenta IPS/IDS que faça a inserção de regras baseando-se em um fluxo de dados. De certo modo enquadra-se na primeira já descrita e também pode enquadrar-se na segunda, pois é possível a criação automática de script de firewall com base nas regras existentes.

As regras IPTables não precisam ser escritas obrigatoriamente nas LXCs (i.e. máquinas virtuais RouteFlow). Neste trabalho utiliza-se as LXCs apenas por comodidade de testes. Para a utilização de outra máquina qualquer é necessário enviar apenas os IPs a serem bloqueados, arquivo este que no ambiente do projeto deve ser editado em "/root/blocked_ip.txt". O rfserver.py faz a leitura deste arquivo para promover as regras ao rfproxy.

3.1 Modificação RFProtocol

De modo a possibilitar a troca de mensagens geradas pelo IPTables entre os demais arquivos estruturais do RouteFlow, o arquivo RFProtocol foi modificado com algumas inserções de classes responsáveis por essa habilitação.

A primeira inserção feita, nomeada "SuspectFlow", possui a tarefa de encaminhar as regras geradas pelo IPTables do RFServer até RFProxy. Assim o RFProxy direciona a respectiva mensagem para a camada física, neste projeto emulada pela máquina virtual Mininet.

Outra modificação realizada é a inserção do "IptablesRegister", utilizado para monitorar os IPs quando as máquinas são iniciadas. Deste modo, sempre que o sistema é iniciado uma busca é feita para verificar se há algum IP suspeito em fluxo pela rede.

Apesar de "IptablesRegister" realizar o monitoramento inicial, viu-se necessário o monitoramento em tempo real. Esta tarefa é implementada com a inserção do código nomeado "IptablesRegisterDynamic" ao RFProtocol. A Figura 2 descreve as linhas de código de cada modificação.

SuspectFlow	**IptablesRegister**	**IptablesRegister**
i8 mod	i64 vm_id	i64 vm_id
i64 vm_id	i32 vm_port	i32 vm_port
i64 dp_id	mac hwaddress	string blocked_ip
i32 vm_port	string blocked_ip	
match[] matches		
string ip_src		
string gateway		

Figura 2. Inserções de código RFProtocol.

3.2 Integrando IPTABLES

No mundo atual onde cada vez mais a globalização está presente em nossas vidas, cada vez mais sistemas complexos e distribuídos são desenvolvidos necessita-se cada vez mais de segurança. O IPTABLES é um método muito utilizado em sistemas para a filtragem de pacotes em uma rede.

O IPTABLES é uma ferramenta de filtragem de pacotes, através da análise de tabelas de filtragem e cabeçalhos de pacotes, ele toma decisões de bloqueio por exemplo. O método de filtragem utilizado no IPTABLES é o *stateful*.

Para o funcionamento correto do IPTABLES necessita-se a criação das tabelas de regras, para isso existem três formas:

- **Linha de comando**: no qual um administrador insere regras via terminal de comandos;
- **Inicialização**: através de escripts de firewall, no qual as regras são previamente definidas em um scrip a ser executado durante a inicialização da VM ou quando solicitado em algum momento;
- **Ferramenta IPS/IDS**: o qual faz inserções de regras baseando-se em fluxo de dados.

Neste projeto foram utilizadas LXC's para o ambiente de testes das regras IPTABLES, mas estas regras podem ser escritas em um arquivo de texto, onde este contém somente os IP's a serem bloqueados, uma *blacklist*, onde o RFServer faz a leitura deste arquivo para promover as regras ao RFProxy.

omo ideia geral o IPTABLES, verifica a existência de bloqueio para um IP, caso exista esse IP na *blacklist* o switch mininet não conseguirá enviar nem receber pacotes provenientes deste IP. Com o bloqueio no switch do IP de origem existe uma redução no trafego, pois o trafego proveniente do IP bloqueado não trafega pela rede, permitindo também melhoras em questões de segurança pois estrafego ilícito não trafega na rede, permitindo assim que essas mensagens não alterem ou burlem as regras IPTABLES existentes.

3.3 Modificações de arquitetura

Agora, descrevemos as modificações feitas no código do Routeflow para o seu funcionamento com o **IPTABLES**.

Primeiramente, cada máquina virtual quando iniciada cria seus registros de informações no **RFServer**, enviando mensagens do tipo:

- **PortRegister**: através dessa o RFServer faz o registro da máquina virtual, colocando-o na tabela **RFTable**.
- **IpTablesRegister**: essa mensagem, contém informações dos **IP's** que serão bloqueados pelas regras do **IPTABLES**.
  - Ao receber a mensagem **IpTablesRegister**, o **RFServer** cria o arquivo blocked_ip e adiciona os **IP's** recebidos de cada **VM**(H1,H2,H3 e H4).
  - O **RFServer** aguarda até que ocorra o evento on_datapath_up (evento que ocorre quando os nó são iniciados no mininet), então o **RFProxy** envia uma mensagem **DatapathPortRegister** ao **RFServer** informando que o switch foi inicializado no mininet.
  - Tendo a mensagem enviada pelo **RFProxy** sido recebida pelo **RFServer**, o arquivo do blocked_ip é lido e então são enviados os comandos OpenFlow para que o instale nos switch's da **mininet** as regras para serem serem feitos "**DROP**" dos packet's desses IP's lidos no blocked _ip, o tipo de mensagem utilizada para o envio é **SuspectFlow** e envia novamente ao **RFProxy**.
  - Tendo O **RFProxy** receebido essa mensagem, ele constrói o comando **OpenFlow** e envia ao switch através do **datapath_id**.
  - Agora com as regras instalad nos switch's da mininet, elas podem ser veficadas pelo comando "**dpctl dump-flows**", ao tentar mandar pacotes para alguns dos Ipś que foram bloqueados é feito um "**DROP**" dos pacotes.

Estas alterações foram feitas para que o **Routeflow** faça uma verificação estática dos IP's contidos na ***blacklist***. Mas, agora apresentaremos as modificações que tornam esse sistema de verificação de regras **IPTABLES** de forma dinâmica.

- Criamos uma thread na **FlowTable** que fica monitorando se o arquivo **blocked_ip_dynamic** foi alterado.
    - Se este arquivo for alterado por qualquer agente, então a thread irá ler o arquivo e enviar uma mensagem **IpTablesRegisterDynamic** para o **RFServer** com o **IP** da regra.
    - Caso o **RFServer** receba a mensagem **IpTablesRigisterDynamic**, então deverá fazer todos os passos já descritos nesta seção, que são mandar mensagens **SuspectFlow** ao **RFProxy** com o IP. O **RFProxy** deverá enviar o comando correto para instalação das regras nos switch's.

Assim o **IPTABLES** é integrado ao **RouteFlow**, onde de forma dinâmica são verificadas a existência de **IP's** a serem bloqueados, para que não exista trafego de pacotes provenientes destes, garantindo assim a segurança contra diversos tipos de ataque.

4. REFERÊNCIAS

[1] Routeflow - Virtual IP Routing Services over OpenFlow networks. http://routeflow.github.io/RouteFlow/

[2] Dominando o IPTables - IPTABLES e suas principais características. http://www.vivaolinux.com.br/artigo/Dominando-o-iptables-(parte-1).