BroFlow: Um Sistema Eficiente de Detecção e Prevenção de Intrusão em Redes Definidas por Software*

Martin Andreoni Lopez, Ulisses da Rocha Figueiredo
Antonio Gonzalez Pastana Lobato e Otto Carlos Muniz Bandeira Duarte

[1]Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)

Abstract. *Intrusion Detection and Prevention Systems are fundamental to inspect real-time network traffic, seeking abnormal patterns caused by intruders or insider misuse, to ensure communication systems security. Moreover, this is the only effective mechanism to detect attacks from internal authenticated users. This paper proposes BroFlow, an Intrusion Detection and Prevention System based on Bro traffic analyzer, and on the global network-view feature of OpenFlow Application Programming Interface. BroFlow main contributions are: (i) intrusion detection through simple algorithms implemented by a modular and flexible architecture; (ii) immediate reaction to an attack and malicious packets dropping from its origin; and (iii) strategic sensor positioning for attack detection in an infrastructure network shared by multi-tenants. A system prototype was implemented and evaluated in the virtual environment Future Testbed Internet with Security (FITS). A system evaluation under attack shows that BroFlow guarantees the forwarding of benign packets at the maximal link rate and reduces, up to ten times, the maximal network delay caused by the attack, even when the attackers are legitimate tenants acting in collusion.*

Resumo. *Os Sistemas de Detecção e Prevenção de Intrusão são fundamentais para inspecionar o tráfego da rede em tempo real, em busca de padrões anômalos causados por intrusos ou abusos de usuários internos, para garantir a segurança dos sistemas de comunicação. Este artigo propõe o BroFlow, um Sistema de Detecção e Prevenção de Intrusão, baseado nas características da ferramenta de análise de tráfego Bro e na visão de rede global da Interface de Programação de Aplicação OpenFlow. As principais contribuições do BroFlow são: (i) detecção de intrusão através de algoritmos simples implementados através de uma arquitetura modular e flexível; (ii) reação imediata a um ataque e descarte dos pacotes atacantes desde a sua origem; e (iii) posicionamento estratégico de sensores para detecção de ataques em uma infraestrutura de rede compartilhada com diversos inquilinos. Um protótipo do sistema proposto foi implementado e avaliado no ambiente de redes virtuais Future Internet Testbed with Security (FITS). A avaliação do sistema sob ataque mostra que o BroFlow garante o encaminhamento de pacotes benignos na rede na taxa máxima do enlace e reduz, em até dez vezes, o atraso na rede provocado pelo ataque, mesmo quando os atacantes são inquilinos legítimos agindo em conluio.*

*Este trabalho foi realizado com recursos da CNPq, CAPES, FAPERJ, FINEP e FUNTTEL.

1. Introdução

Os ataques às redes de comunicação são cada vez mais comuns devido às vulnerabilidades que são criadas com o crescimento e complexidade das redes. Os impactos dos ataques variam de acordo com as ameaças e os riscos, afetando a distintos usuários que utilizam o mesmo ambiente de computação em nuvem, provocando desastres incalculáveis a infraestruturas críticas como, por exemplo, as redes elétricas inteligentes [Guimarães et al. 2013]. Como defesa contra essas ameaças, são usados diversos sistemas de segurança que agem como barreiras de proteção, tais como *firewall* e mecanismos de controle de acesso. No entanto, 70% dos ataques são originados por usuários internos e legítimos [Lynch 2006] da rede e, portanto, os métodos convencionais de segurança são completamente ineficazes. Logo, os Sistemas de Detecção e Prevenção de Intrusão (IDPS) são mandatórios para complementar os métodos convencionais de segurança, protegendo o sistema de ataques internos e externos.

A detecção de intrusão é o processo de monitorar e analisar eventos de um sistema em busca de sinais de possíveis incidentes de segurança. Os sistemas de Prevenção de Intrusão (*Intrusion Prevention System* -IPS) são considerados uma extensão dos sistemas de Detecção de Intrusão (*Intrusion Detection System* -IDS), pois atuam sobre o sistema uma vez que as ameaças ou atividades maliciosas são detectadas. Os ataques de negação de serviço (*Denial of Service* - DoS) visam consumir uma grande quantidade de recursos da vítima, de forma a interromper os serviços prestados ou evitar que usuários legítimos sejam atendidos com a qualidade de serviço adequada. Existem registros de ataques de negação de serviço por inundação que causaram perdas milionárias [Siris e Papagalou 2006]. O problema abordado neste artigo é o combate de ataques de DoS. Devido à falta de uma visão global de controle da rede, a detecção do ataque ocorre perto do destino, sendo mais difícil bloquear o ataque perto das origens. Uma maneira de obter uma visão global e, consequentemente, um melhor gerenciamento da rede é utilizar as Redes Definidas por Software (*Software Defined Networking* - SDN) que propõem a separação do plano de controle centralizado do plano de dados distribuído. O OpenFlow (OF) é a implementação de maior sucesso do paradigma SDN e seus elementos encaminhadores são chamados de comutadores OF.

Este artigo propõe o BroFlow, um sistema de Detecção e Prevenção de Intrusão (IDPS) distribuído para a defesa contra ataques de negação de serviço (DoS). São três as principais contribuições do sistema BroFlow: (i) a detecção em tempo real de ataques de negação de serviço através de algoritmos simples implementados em uma linguagem específica para eventos de rede; (ii) a reação imediata para a interrupção de ataques de negação de serviço; e (iii) o posicionamento estratégico de sensores na rede para garantir o correto funcionamento da infraestrutura física, mesmo em cenários em que os inquilinos legítimos agindo em conluio são considerados atacantes. Para tanto, o BroFlow é baseado na Interface de Programação de Aplicação de redes OpenFlow [McKeown et al. 2008] e na ferramenta de análise de tráfego Bro [Sommer 2003]. Foram implementados diferentes algoritmos para a detecção de anomalias que correspondem a ataques de negação de serviço por inundação. Os sensores do BroFlow enviam alarmes ao controlador POX através de canais seguros para que uma aplicação OpenFlow efetue as contramedidas aplicáveis, de modo a bloquear o ataque de DoS o mais perto possível da fonte e, assim, eliminar o fluxo malicioso no comutador OpenFlow de entrada do fluxo na rede.

A maioria das propostas de Sistemas de Detecção de Intrusão visa identificar a ocorrência de ataques e notificar, através de alarmes, os administradores da rede. A proposta BroFlow de detecção e prevenção de intrusão utiliza a visão global da rede para agir diretamente no controle do encaminhamento dos fluxos na rede, bloqueando de forma eficaz o fluxo malicioso ainda na sua origem. Também é possível redirecionar o fluxo malicioso para uma réplica do servidor atacado ou para um servidor de pote de mel *(honeypot)*. A eficiência e flexibilidade oferecida pela ferramenta Bro de análise de tráfego permite ao sistema proposto a criação de políticas eficientes de segurança individuais para cada inquilino da rede. Com a ferramenta Bro é possível detectar e bloquear os ataques de maneira local, enquanto que o controle e gerenciamento global da rede fornecida pelo OpenFlow permite aplicar esse bloqueio em todos os comutadores da rede eliminando o ataque desde a origem. Um protótipo do sistema BroFlow foi desenvolvido, avaliado e testado na plataforma *Future Internet Testbed with Security* (FITS), que é uma plataforma de teste de redes baseada na técnica de virtualização. Os resultados mostram a eficácia do sistema para detectar e reagir ante ataques de DoS por inundação.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A ferramenta Bro é detalhada na Seção 3. O sistema proposto é apresentado na Seção 4. A Seção 5 discute os resultados da avaliação do sistema. Por fim, Seção 6 conclui o artigo.

2. Trabalhos Relacionados

As Redes Definidas por Software (*Software Defined Networks* - SDN) têm o controle centralizado e a visão global da rede, o que faz com que elas sejam eficazes para a detecção e reação a ataques de segurança, em particular aos ataques de negação de serviço por inundação. A programabilidade do protocolo OpenFlow permite o gerenciamento dinâmico dos fluxos nos comutadores da rede. Essa característica permite o seu uso na área de segurança de redes e vem sendo abordada por diferentes pesquisadores [Porras et al. 2012] [Shin et al. 2013].

Mattos *et al.* propõem o isolamento da comunicação entre redes virtuais que compartilham uma mesma infraestrutura física [Mattos et al. 2013]. Esta proposta visa assegurar a confidencialidade da rede virtual de cada inquilino sobre a infraestrutura física através do isolamento de recursos e de tráfego, prevenindo ataques de bisbilhotamento (*eavesdroping*). No entanto, garantir o isolamento entre redes virtuais previne o ataque de uma rede virtual sobre outra rede virtual no mesmo roteador físico, mas não visa detectar a ocorrência de ataques de negação de serviço interno a uma rede virtual.

Porras *et al.* abordam o problema de potenciais conflitos no estabelecimento de regras de fluxos entre diferentes aplicações nos controladores OpenFlow. Como solução, propõem uma extensão de segurança nos controladores NOX-OpenFlow [Porras et al. 2012]. Nessa extensão, cada aplicação tem políticas com diferentes prioridades atribuídas, dependendo da sua função. Ao serem aplicadas nos fluxos do controlador, as políticas voltadas para as aplicações de segurança OpenFlow são prioritárias. Shin *et al.* propõem um arcabouço de segurança para o controlador NOX-OpenFlow, o sistema FRESCO [Shin et al. 2013] provê uma linguagem de programação e módulos de software para desenvolver aplicações de segurança. Esses módulos permitem a integração de serviços de segurança sobre uma nova arquitetura de controladores

NOX-OpenFlow.

Os ataques de negação de serviço em redes SDN são estudados por Braga *et al.*, que propõem um método de detecção de ataques de negação de serviço distribuídos (*Distributed Denial of Service* -DDoS) usando um controlador NOX-OpenFlow [Braga et al. 2010]. Esse trabalho consiste em monitorar os comutadores de uma rede durante determinado período, extraindo características dos fluxos que entram nas tabelas do comutador, tais como média de pacotes por fluxo, média de bytes por fluxo, entre outras. Esses atributos são enviados a um classificador dentro do controlador NOX, baseado em redes neurais, que determinam se o tráfego é normal ou se corresponde a um ataque. As redes neurais requerem um treinamento prévio com conjuntos de dados artificiais, o que é uma limitação importante na área de IDS [Sommer e Paxson 2010]. Além disso, o processamento do controlador da rede é sobrecarregado pelo processo de detecção.

Existem propostas de Sistemas de Detecção e Prevenção de Intrusão (IDPS) em redes virtuais. NICE [Chung et al. 2013] implementa um IDPS em ambientes de nuvem, criando um modelo analítico baseado em grafos dos ataques que, dependendo do estado do ataque, realiza, como contramedida, a adaptação da topologia da rede para evitar o ataque. Esta proposta é baseada no protocolo OpenFlow sobre comutadores Open vSwitch (OVS). Esse trabalho evoluiu para o SnortFlow [Xing et al. 2013], que consiste em um IDPS baseado no Snort, um IDS *open source* de detecção por assinatura, e no OpenFlow para o encaminhamento dos pacotes. O analisador Snort é instalado no domino de gerencia do hipervisor XEN, que é conectado em um comutador ligado às maquinas virtuais de uma máquina física para inspecionar o tráfego das maquinas virtuais. Essa proposta carece do sincronismo entre o agente Snort e controlador para ter uma visão global da rede. Além disso, a ferramenta Snort só utiliza o método de detecção por assinatura, não detectando ataques frente a pequenas variações dos ataques.

O IPSFlow [Nagahama et al. 2012] é outra proposta baseada em SDN e propõe um mecanismo de bloqueio automático de tráfego malicioso utilizando o protocolo OpenFlow. Na proposta, o tráfego é duplicado para a análise no IDS, gerando fluxos novos na rede. Além disso, a análise dos fluxos é feita de maneira seletiva, porém existe uma grande possibilidade de não inspecionar fluxos maliciosos durante a seleção, já que durante um ataque de DoS, todos os campos dos pacotes são similares aos benignos.

A empresa Radware comercializa um equipamento [Radware 2014] para a prevenção de ataques de DoS em SDN. O equipamento implementa um controlador de código proprietário que gerencia uma rede de comutadores OpenFlow e utiliza lógica nebulosa para determinar o grau do ataque, baseado no comportamento da taxas de pacotes, transmissão e abertura de conexões, além do comportamento dos protocolos, de acordo com padrões previamente obtidos. Com esses valores é possível diminuir a alta taxa de falsos positivos obtidos pela lógica nebulosa. A ferramenta precisa de um período de aprendizagem do comportamento benigno para obter a base de dados para o treinamento.

Este artigo propõe o sistema BroFlow que é um Sistema de Detecção e Prevenção de Intrusão (IDPS) distribuído em Redes Definidas por Software (*Software Defined Networking* - SDN) para a defesa de ataques de negação de Serviço (DoS). O BroFlow

utiliza a ferramenta Bro para analisar o tráfego e inspecionar com eficiência todos os pacotes de um fluxo. O Bro provê uma sinalização de eventos em alto nível e uma linguagem própria para estabelecer políticas de segurança de uma maneira fácil. Logo, o Bro permite a implementação de diferentes algoritmos para a detecção por anomalias de ataques de negação de serviço por inundação. Na proposta, a ferramenta Bro de análise de tráfego é utilizada nos sensores localizados estrategicamente que geram alarmes quando uma anomalia é detectada. Além disso, esses alarmes são enviados a um controlador de rede OpenFlow que aciona uma contramedida para bloquear o ataque de maneira global, aliviando ao controlador do processo da detecção. O BroFlow utiliza a visão global fornecida pelo OpenFlow para bloquear os ataques de DoS em todos os enlaces desde o destino até a origem.

3. A Ferramenta Bro de Análise de Tráfego

O Bro [Sommer 2003] é uma ferramenta de código aberto para o monitoramento e análise em tempo real do tráfego de rede. Essa ferramenta escuta a comunicação entre dois pontos da rede, reconstruindo a semântica das conexões e armazenando os estados para cada uma delas. A arquitetura do Bro, representadas na Figura 1, é dividida em três camadas principais: a captura dos pacotes; a abstração da atividade da rede em eventos; e a interpretação das políticas, na qual o usuário define os seus próprios *scripts*.

A separação dos mecanismos de captura e da interpretação das políticas é uma das principais vantagens do Bro, pois lhe confere flexibilidade e simplicidade. O Bro possui mais de 30 analisadores de diferentes protocolos, tais como FTP, HTTP, ICMP, UDP, TCP, DNP3, entre outros, e as políticas de segurança estabelecem que ação que deve ser executada pelo mecanismo de segurança. Logo, se uma atividade suspeita é observada, os analisadores disparam um evento de forma assíncrona que são enviados à camada superior, na qual são interpretados baseados nas políticas descritas pelos usuários numa linguagem específica, linguagem BRO. Portanto, as políticas de segurança estão separadas da detecção de cada ameaça.

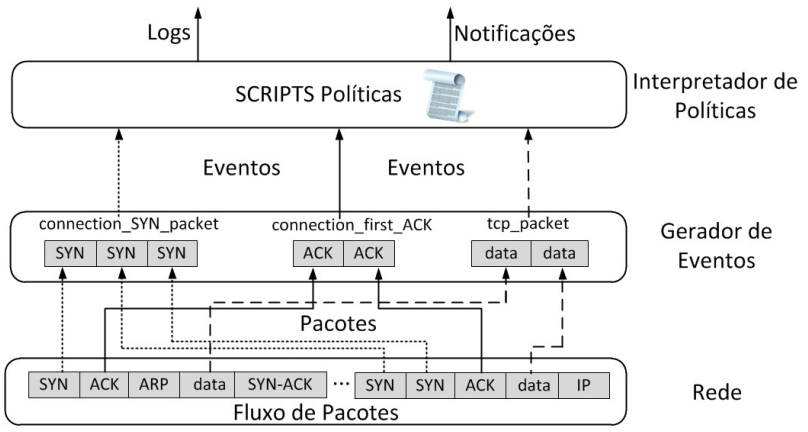


Figura 1. A arquitetura Bro de três camadas: captura dos pacotes na rede; gerador de eventos, onde é analisada a semântica dos pacotes; interpretação de políticas, na qual os eventos são tratados pelos scripts do usuário.

O gerador de eventos do Bro define eventos para as atividades da rede, o que representa uma abstração dos pacotes em um nível de informação mais elevado. Depois que o fluxo de pacotes é capturado, o gerador de eventos realiza a verificação de *checksum*

para assegurar a integridade e checar se os cabeçalhos estão bem formados. No caso das verificações falharem, os pacotes são descartados e são gerados eventos indicando o problema encontrado. No caso de sucesso das verificações de integridade, o gerador de eventos procura o estado associado aos endereços IP e às portas de origem e destino do pacote capturado e, se não encontrar nenhum estado associado, cria um novo estado. Em seguida, os pacotes são ordenados por conexões, reagrupados por fluxos de dados TCP/UDP e decodificados pelos protocolos da camada de aplicação. Um exemplo da geração de eventos são as três trocas de mensagens da conexão do TCP (*three way handshake*), correspondente a um evento de estabelecimento de conexão bem sucedido (`conection_established`) com informações tais como host, IP, porta, tempo de início, entre outros, que são enviados à camada de políticas.

A camada de interpretação de políticas do Bro confere flexibilidade ao sistema BroFlow, pois as políticas de segurança são especificadas mediante scripts descritos na linguagem BRO, que permite a definição da forma de gerenciar os eventos gerados previamente. A definição das políticas depende do ambiente de segurança que, para o mesmo evento, pode exigir ações mais frouxas e liberais, como em um ambiente universitário, ou mais rígidas, quando se trata de setores sensíveis de uma empresa. Outra vantagem é a inspeção e análise posterior, pois toda vez que uma violação das políticas é detectada, um alerta é disparado, gerando notificações em tempo real ou *logs* para análise posterior.

O que diferencia o Bro dos demais analisadores de tráfego, como Snort, `tcpdump`, netFlow, entre outros, é a descrição de políticas na sua própria linguagem. A linguagem BRO, similar à linguagem C, foi projetada levando em conta o uso de redes. Assim, existem tipos de dados específicos para endereços IP, portas, intervalos de tempo, entre outros tipos característicos do ambiente de redes. Quando comparado com outros monitores de redes, tais como o `tcpdump`, baseado em caracteres ASCII, os tipos de dados trazidos pela linguagem BRO facilitam o entendimento e evitam "erros simples" [Sommer 2003]. Dessa forma, com a linguagem específica do Bro, é possível escrever de maneira fácil e intuitiva, *scripts*, denominados políticas, para que a detecção de intrusão seja por anomalias ou assinaturas.

4. O Sistema Proposto

O sistema proposto, denominado BroFlow, segue o paradigma de redes definidas por software, no qual as funções de controle da rede podem ser programadas e as tabelas de encaminhamento dos fluxos são atualizadas nos comutadores através do protocolo OpenFlow [McKeown et al. 2008]. No sistema, o Open vSwitch [Pfaff et al. 2009], que é um comutador programável por *software*, é utilizado como comutador OpenFlow. Além de apresentar uma tabela de encaminhamento que pode ser atualizada por um controlador OpenFlow, o Open vSwitch oferece as funcionalidades de descartar pacotes, alterar campos do cabeçalho, entre outras. A configuração e o controle dos comutadores OpenFlow é realizada pelo controlador POX[1]. O POX foi escolhido entre outros controladores, pela simplicidade na programação mantendo seu alto desempenho. Todas as contramedidas são baseadas nas ações do protocolo OpenFlow, que permitem encaminhar, redirecionar, modificar ou descartar pacotes dinamicamente em cada um dos fluxos.

A arquitetura do sistema BroFlow considera um ambiente de virtualização de re-

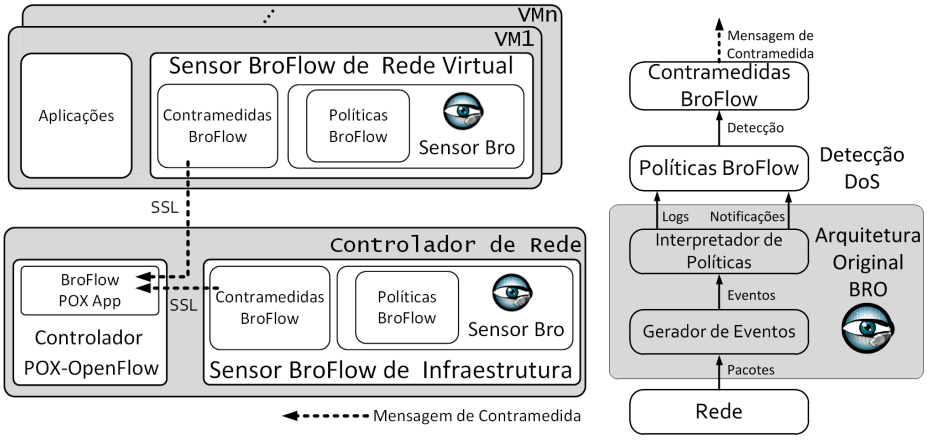
[1] http://www.noxrepo.org/pox/about-pox/

des híbrido, composto por Máquinas Virtuais (MV) Xen executando sobre uma matriz de comutação OpenFlow [Mattos e Duarte 2012]. Nesse ambiente de virtualização, as MV se interconectam através de comutadores OpenFlow, implementados pelo comutador programável Open vSwitch (OVS), instanciados nas máquinas físicas.

Dado que o sistema possui sensores, resulta o problema de otimização da localização dos sensores na rede. Para a detecção do ataque é possível colocar um número de sensores reduzido em vez de colocar sensores em todos os roteadores. Essa vantagem é outorgada pela combinação da ferramenta Bro com a visão global fornecida pelo OF, de modo que é possível ter um controle global da rede mediante o monitoramento in-situ dos sensores BroFlow. Assim, surge a pergunta onde colocar estrategicamente os sensores tendo o melhor desempenho da rede e mantendo a precisão da detecção no momento do bloqueio dos ataques de DDoS. Esse problema está fora do escopo desse artigo e será analisado em trabalhos futuros.

4.1. Os Sensores BroFlow

No sistema BroFlow existem dois tipos de sensores. Os sensores BroFlow de Rede Virtual (BFRV) e o sensor BroFlow da Infraestrutura (BFI), como ilustrado na Figura 2(a). Os sensores de BroFlow de Rede Virtual devem estar distribuídos em todos os roteadores virtuais ou localizados em pontos estratégicos dentro dela. Os sensores BFRV podem monitorar tanto os roteadores virtuais como uma estação específica, tal como um servidor de aplicação devido à sua importância na rede. Em cada um dos sensores BFRV são estabelecidas políticas específicas e independentes para cada rede virtual. Esta facilidade provida pelo sistema BroFlow é importante dentro de um ambiente de nuvem, pois a política persiste mesmo quando ocorre a migração de um roteador virtual, uma vez que o sensor BroFlow também migra junto com o roteador.



(a) Arquitetura do BroFlow.

(b) Módulos Sensores BroFlow.

Figura 2. a) A arquitetura do sistema BroFlow. b) O sensor BroFlow e a sequência de atividades até o envio de mensagem de contramedida para a aplicação BroFlow POX no controlador.

O Sensor BroFlow de Infraestrutura é colocado paralelamente ao controlador para proteger a infraestrutura física da rede, ou seja, a rede física subjacente às redes virtuais monitoradas pelo BroFlow. Um exemplo de ataque à infraestrutura é o ataque de

inundação por ARP, também conhecido como envenenamento ARP. Esse ataque consiste em atacar comutadores com pacotes ARP para saturar as tabelas de encaminhamento com endereços MAC falsos. Isso provoca a negação de serviço na memória dos comutadores da infraestrutura. O Sensor da Infraestrutura BroFlow detecta e impede esse tipo de ataque, protegendo a rede física e as redes virtuais que ela hospeda.

Cada um dos sensores BroFlow contém um *daemon* da ferramenta Bro, o qual executa em paralelo às demais aplicações, com um consumo mínimo de recursos do sistema. Cada Sensor BroFlow tem aplicações de detecção de ataques e envio de alarmes definidos por dois módulos: o Módulo de Políticas BroFlow e o Módulo de Contramedidas. O módulo de Políticas BroFlow é implementado diretamente no *daemon* Bro, sendo abstraídos os algoritmos de detecção em políticas descritas na linguagem BRO.

Além disso, a aplicação BroFlow POX, que executa no controlador, gerencia os alarmes e as contramedidas recebidas. A Figura 2(b) detalha os módulos dos sensores BroFlow. Os pacotes capturados são enviados ao gerador de eventos, que os verifica, ordena e converte em eventos, que são enviados em seguida ao interpretador de políticas. O gerador de eventos e o interpretador de políticas fazem parte da arquitetura original da ferramenta Bro. O módulo de políticas BroFlow possui a inteligência para decidir se os eventos gerados pelo Bro constituem realmente um ataque e, em caso positivo, qual ação a aplicação BroFlow POX deve realizar.

4.1.1. As Políticas BroFlow

Na Figura 3 é apresentada a arquitetura de políticas de segurança do sistema BroFlow, composta de três módulos principais: inspeção de eventos de rede, detecção de ataques e contramedidas. O Módulo de Inspeção de Eventos de Rede analisa os eventos de rede, fornecidos em tempo real pelo sistema Bro com as informações pertinentes aos fluxos estabelecidos durante a recepção de pacotes. As políticas são descritas na linguagem BRO e o sistema BroFlow oferece três políticas para detecção de ataques de negação de serviço por inundação de pacotes: TCP-SYN, ICMP, e UDP. Assim, toda vez que um pacote relativo a esses eventos é detectado pelo Módulo de Inspeção de Eventos de Rede, o Módulo de Detecção de Ataques é invocado. Nesse módulo, são implementados os diversos algoritmos, abstraídos a políticas na linguagem BRO, que decidem se existe ou não um ataque para que um alarme seja emitido. Nos exemplos implementados, a detecção é feita por estabelecimento de limiares de rampa ou adaptativo. No caso de desejar outro algoritmo de detecção, sua implementação é facilitada pela programação de alto nível da linguagem BRO. Quando um ataque é detectado, o Módulo de Contramedida é invocado. Nesse módulo qualquer programa externo pode ser chamado para realizar as contramedidas necessárias para evitar o ataque.

O protótipo do BroFlow implementa dois tipos de algoritmos de detecção de ataques por inundação: por rampa e por limiar adaptativo [Siris e Papagalou 2006]. O algoritmo de detecção por rampa efetua o somatório de pacotes durante um determinado período de tempo e emite um alarme se um limiar especificado é ultrapassado. O Algoritmo 1 usa a técnica de limiar adaptativo e tem como objetivo diminuir os falsos positivos ou “flash crowds”, que excedem rapidamente os valores médios. O algoritmo detecta mudanças nas estatísticas do tráfego, baseado em medições de tráfego de rede em inter-

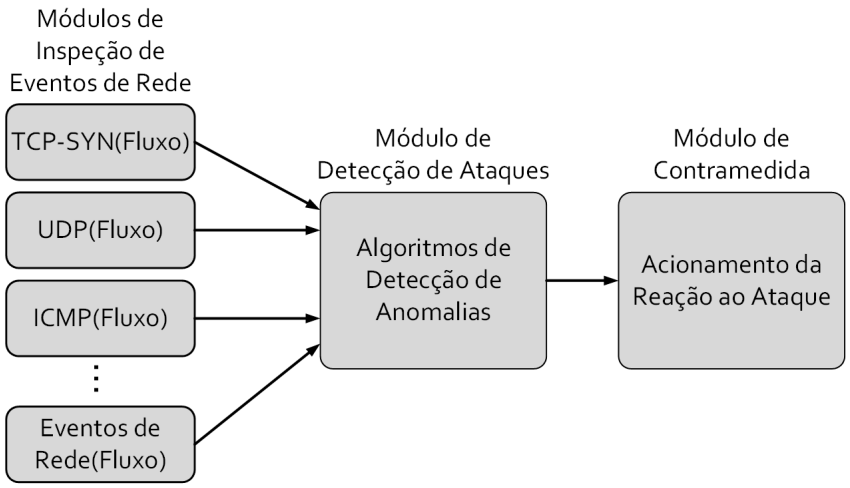


Figura 3. Arquitetura de políticas de segurança do sistema proposto composta por três módulos sequenciais: inspeção de eventos de rede, detecção de ataques e contramedidas.

valos consecutivos de tempo *T*.

Toda vez que o limiar é ultrapassado, um contador *k* é incrementado. Se o contador for maior que um, indica que houve violação do limiar por períodos consecutivos de tempo, o que caracteriza uma anomalia considerada ataque. Os parâmetros de ajuste do algoritmo são a amplitude do fator α, para calcular o limiar, o número de violações do limiar consecutivas *k*, o fator média móvel exponencial ponderada (*Exponential Weighted Moving Average* - EWMA) β e o tamanho em segundos do intervalo de tempo *T*.

Algoritmo 1: Detecção de DoS pelo método de limiar adaptativo.

$\alpha =$ porcentagem acima da média que é considerada anômala
$\beta =$ valor Média Móvel Exponencial Ponderada (EWMA)
$\mu_n =$ média estimada de pacotes no intervalo *n*
$\lambda_n =$ quantidade de pacotes no intervalo *n*
T = duração do intervalo em segundos
k = número de intervalos consecutivos em que o limiar foi violado
while *período* $<$ *T segundos* **do**
  $\lambda_n = \lambda_n + 1$
end
$\mu_n = \beta\mu_{n-1} + (1-\beta)\lambda_n$
if $\mu_n > (\alpha+1)\lambda_n$ **then**
  incremento de *k*;
  if *k*> *1* **then**
    ativa alerta;
    mensagem para controlador;
  end
end

4.2. As Contramedidas BroFlow

O módulo de contramedida realiza a comunicação com a aplicação BroFlow POX no controlador de rede OpenFlow. O módulo de contramedidas, escrito na linguagem *Python*, traduz as informações geradas pelas sensores BroFlow e encaminha as mensagem de alarme para a aplicação BroFlow POX App. Quando um ataque é detectado pelo módulo de políticas nos sensores BroFlow, uma mensagem de alarme é enviada para o controlador POX-OpenFlow. Esses módulos de comunicação usam o protocolo *Secure Socket Layer (SSL)* em interfaces de redes dedicadas, garantindo comunicações encriptadas e autenticadas. As mensagens são enviadas no formato JSON e contém as informações do fluxo, como endereços IP e portas, de origem e de destino, obtidas através da ferramenta Bro, e também o endereço MAC de destino e a contramedida a ser tomada.

Os campos nas mensagens enviadas são as informações que o sistema Bro consegue monitorar de um pacote. Como esses campos não exaurem as possibilidades de campos de um fluxo completo OpenFlow, os demais campos são completados com valores coringas (*wildcard*). Embora essa definição instale fluxos gerais nos comutadores, o uso de campos coringas não gera ambiguidade, pois uma conexão TCP é normalmente definida por quatro campos do pacote somente: IP e portas de origem e destino. Assim, como os quatro campos que identificam uma conexão TCP são bem definidos, não há ambiguidade nos fluxos considerados maliciosos.

4.3. A Aplicação BroFlow POX App

A aplicação BroFlow POX é uma aplicação do controlador POX-OpenFlow que recebe as mensagens de alarme provenientes dos diferentes sensores BroFlow e executa as contramedidas necessárias para respondê-los. Assim, ao receber a mensagem de alarme proveniente do modulo de contramedidas, a aplicação BroFlow POX verifica em sua tabela de fluxos qual o fluxo corresponde ao conteúdo da mensagem de alarme. Em seguida, a aplicação indica ao controlador POX-OpenFlow a ação de contramedida a ser executada em todos os comutadores da rede. No protótipo BroFlow, as contramedidas possíveis, no caso de uma ameaça, correspondem às ações OpenFlow especificas: *drop*, de descarte do pacote e *output*, de encaminhamento do pacote para uma porta definida. Portanto, com essas duas ações do protocolo OpenFlow sobre os comutadores da rede, as contramedidas definidas pelo BroFlow podem ser, o *bloqueio* de um fluxo específico ou *desvio* de um fluxo para uma outra estação.

Assim, no caso de um ataque de negação de serviço (DoS), os sensores BroFlow detectam o ataque de acordo com a política de segurança enviando uma mensagem de alarme indicando a contramedida a ser realizada. Vale lembrar, que a comunicação entre sensores e o controlador é executada em uma rede isolada da rede de comunicação compartilhada pelos inquilinos.

Além disso, se a política de segurança do BroFlow detectar um ataque de varredura de portas, o fluxo malicioso pode ser redirecionado para um servidor de pote de mel (*honeypot*), para estudar as características do ataque. Todas as contramedidas são aplicadas sob regime de quarentena, ou seja, cada vez que uma contramedida é aplicada nos comutadores, um temporizador é ativado. Quando o temporizador estoura, a contramedida é apagada e todas as análises de ataques são realizadas novamente. Assim, é possível detectar se o ataque foi encerrado, deixando de utilizar recursos nas contramedidas.

5. Desenvolvimento e Avaliação do Protótipo

Um protótipo do BroFlow foi implementado e avaliado na plataforma de teste de redes virtuais `Future Internet Testbed with Security` (FITS) [Guimaraes et al. 2013]. O FITS [2] possui nós distribuídos geograficamente em universidades brasileiras e europeias para o desenvolvimento de experimentações em redes de nova geração. O FITS permite a criação e o gerenciamento de diferentes redes virtuais. As principais características da plataforma FITS são o isolamento de redes virtuais, o acesso seguro do gerenciamento da rede e a diferenciação da Qualidade de Serviço entre redes virtuais.

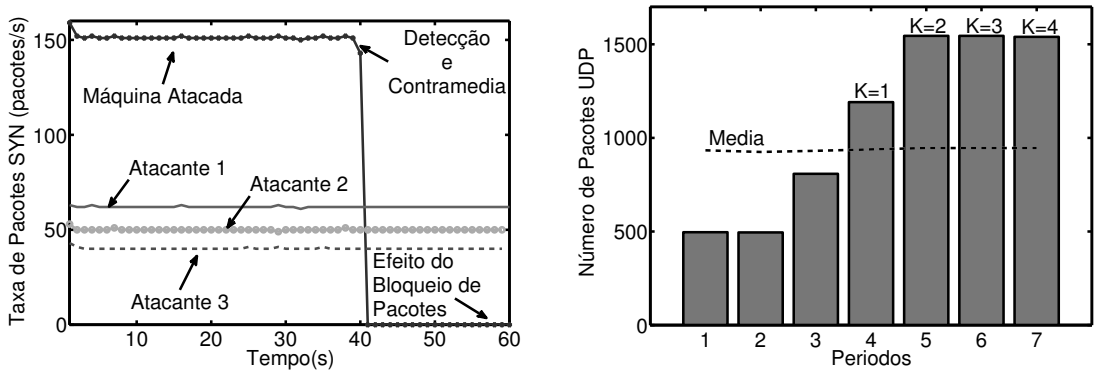
O primeiro experimento analisa ataques de DoS por inundação de pacotes SYN em redes virtuais que são hospedadas em roteadores físicos. No ambiente virtualizado, também existem ameaças de segurança às redes virtuais, percebida por ataques aos roteadores virtuais, e ameaça à infraestrutura física, percebida por ataques ao roteador físico. O cenário considerado é um roteador físico hospedando quatro roteadores virtuais pertencentes a quatro redes virtuais diferentes. Em cada uma das máquinas virtuais correspondentes aos roteadores virtuais é instalado um sensor BroFlow, que analisa o tráfego de cada rede virtual separadamente. Além disso, um sensor BroFlow de infraestrutura é instalado no roteador físico.

Para realizar o ataque de inundação de SYN foi desenvolvido um *script*, no qual é possível regular uma taxa constante de pacotes SYN do ataque em 45, 50 ou 55 pacotes por segundo. Define-se, a critério de teste, o valor do limiar como 100 pacotes SYN por segundo. Esse valor representa a taxa de pacotes SYN permitida em cada uma das redes. Logo, o valor limiar das redes não é superado. No entanto, como os inquilinos legítimos agem em conluio, o limiar estabelecido é superado no roteador físico da infraestrutura em mais de 50%, pois a taxa agregada de conexões por segundo das redes virtuais é totalmente encaminhada pelo roteador físico que as hospeda. Nesse experimento, o limiar das redes virtuais não é extrapolado individualmente, mas o limiar agregado é considerado um ataque pelo sensor da infraestrutura. A cada instante em que o limiar é superado, um contador é incrementado para uso no algoritmo de limiar adaptativo. Sendo assim, do ponto de vista do sensor de infraestrutura, há a ocorrência de um ataque de negação de serviço à infraestrutura física de redes virtuais.

A Figura 4(a) mostra o experimento com os três atacantes enviando pacotes SYN com três taxas constantes diferentes. No momento da detecção, aproximadamente aos 40 segundos, uma mensagem é gerada pelo sensor do BroFlow presente no roteador físico do roteador atacado. Esse valor de 40 segundos é devido à implementação do algoritmo de detecção por limiar adaptativo. Como foi mencionado na Seção 4, esse algoritmo incrementa um contador k, quando a taxa média do período anterior for violado. Assim, se a taxa média de pacotes SYN for ultrapassada só uma vez, a contramedida não é disparada assumindo que é um falso positivo, mas se a taxa for violada quatro vezes consecutivas a contramedida é enviada à aplicação BroFlow POX.

O segundo experimento, ilustrado na Figura 4(b), avalia o algoritmo de detecção de anomalias por limiar adaptativo no mesmo cenário anterior, mas com um ataque de inundação UDP, utilizando a ferramenta de ataques DoS `Trin00`. Esse método di-

[2] http://www.gta.ufrj.br/fits/

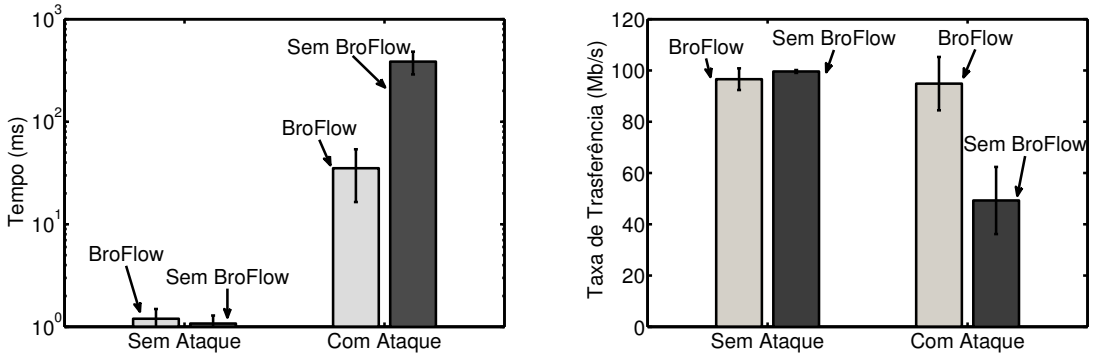


(a) Ataques de inundação de pacotes à taxa constante e efeito da ação de bloqueio.

(b) Número de pacotes UDP por período T e incremento do valor k.

Figura 4. a) Três ataques de inundação de SYN com uma taxa constante, detecção do ataque e efeito da ação de contramedida de bloqueio em 40 segundos. b) Comportamento da variável k do algoritmo de detecção por limiar adaptativo, durante um ataque de inundação UDP. O sensor de infraestrutura física identifica ataques de DoS distribuídos em redes virtuais distintas.

vide a recepção de pacotes em intervalos T de 10 segundos. A taxa média de pacotes UDP recebidos é estabelecida em 1000 pacotes por segundo. A cada vez que o valor da média do intervalo anterior é superado, um valor k é incrementado. Nesse experimento, o valor limiar de $k = 4$ foi estabelecido. Assim, sempre que os fluxos de pacotes UDP recebidos superarem a média de pacotes do intervalo anterior, o valor k é incrementado. Quando esse valor supera 4, uma mensagem de alarme é enviada ao controlador, indicando que a rede está sofrendo um ataque de inundação UDP. O controlador então realiza o bloqueio do fluxo malicioso com um tempo mínimo de reação de 40 segundos. No experimento, os valores do algoritmo por limiar adaptativo adotados foram $\mu_0 = 1000, \beta = 0.75, \alpha = 0.2$, e $T = 10$ segundos, sendo os mesmos valores utilizados por Siris e Papagalou [Siris e Papagalou 2006] durante a avaliação dos seus algoritmos.



(a) Atraso médio de comutação de pacotes.

(b) Comparação da taxa de transferência.

Figura 5. Avaliação do desempenho do atraso e da taxa de transferência do sistema BroFlow com e sem ataque. O atraso na rede é reduzido em até 10 vezes em um cenário com ataques.

A Figura 5 mostra tanto a sobrecarga introduzida pelo BroFlow, quanto a sua

eficácia em bloquear um ataque de negação de serviço. A Figura 5(a) compara os atrasos médios de comutação de pacotes do sistema sem ataque e durante o ataque. Observa-se que o atraso inserido pelo sistema BroFlow é insignificante quando não há ataques. Já na presença de um ataque, o sistema BroFlow diminui o atraso médio devido aos descartes dos pacotes de ataque. Na Figura 5(b) é mostrada a taxa de transferência de pacotes. Observa-se que o BroFlow praticamente não sobrecarrega o sistema na condição sem ataque de negação de serviço atingindo a taxa máxima de 100 Mb/s. Durante o ataque de DoS, a taxa de transferência cai de 50% da taxa máxima, enquanto a taxa máxima permanece praticamente inalterada com o sistema BroFlow. Os resultados mostrados são a média durante 5 execuções com um intervalo de confiança de 95%.

6. Conclusão

Nesse artigo foi apresentado o BroFlow, um sistema de detecção e prevenção intrusão, em especial para ataques de negação de serviço por inundação de pacotes, para redes definidas por software. O BroFlow une a eficiência, a flexibilidade e a simplicidade de elaboração de políticas de segurança, providas pela ferramenta Bro de análise de tráfego, com as características da visão global e da agilidade de ação em toda a rede oferecidas pelo OpenFlow. As contribuições do artigo são evidenciadas através do desenvolvimento de um protótipo. O protótipo do sistema demonstra o funcionamento dos diferentes algoritmos de detecção de intrusão por anomalia. Ao tomar ações imediatas para interromper o ataque, o protótipo mostrou ser altamente efetivo na detecção de ataques de negação de serviço por inundação e também eficaz na ação de bloqueio contra os ataques. O sistema descarta os pacotes de ataque ainda na sua origem o que permite aumentar a disponibilidade da rede. O posicionamento estratégico dos sensores de ataques permite ao sistema reduzir em até dez vezes o atraso na rede sob ataque e garante o encaminhamento de pacotes úteis na taxa máxima do enlace, ao custo de acrescentar um atraso praticamente desprezível na rede em um cenário sem ataques. Vale ressaltar, que os experimentos com o protótipo evidenciam que o posicionamento estratégico de sensores de ataques na infraestrutura física assegura a proteção da infraestrutura física contra ataques provenientes de inquilinos legítimos. Como trabalhos futuros, está previsto a proposta de novos algoritmos para a detecção de um maior espectro de ataques e introduzir novas contramedidas que se sirvam da característica peculiar de controle global para correlacionar os alarmes de diferentes sensores BroFlow para a detecção de ataques distribuídos. Além disso, como foi estabelecido previamente, se abordará o problema de otimização na localização dos sensores BroFlow.

Referências

[Braga et al. 2010] Braga, R., Mota, E. e Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. Em *IEEE 35th Conference on Local Computer Networks*, páginas 408–415.

[Chung et al. 2013] Chung, C., Khatkar, P., Xing, T., Lee, J. e Huang, D. (2013). NICE: Network intrusion detection and countermeasure selection in virtual network systems. *IEEE Transactions on Dependable and Secure Computing*, 10(4):198–211.

[Guimaraes et al. 2013] Guimaraes, P. H. V., Ferraz, L. H. G., Torres, J. V., Mattos, D. M. F., Murillo P., A. F., Andreoni L., M. E., Alvarenga, I. D., Rodrigues, C. S. C. e Duarte, O.

C. M. B. (2013). Experimenting content-centric networks in the future internet testbed environment. Em *IEEE ICC Workshops*, páginas 1383–1387.

[Guimarães et al. 2013] Guimarães, P. H. V., Murillo P., A. F., Andreoni L., M. E., Mattos, D. M. F., Ferraz, L. H. G., Pinto, F. A. V., Costa, L. H. M. K. e Duarte, O. C. M. B. (2013). Comunicação em redes elétricas inteligentes: Eficiência, confiabilidade, segurança e escalabilidade. Em *Minicursos do SBRC'13*, páginas 101–164.

[Lynch 2006] Lynch, D. M. (2006). Securing against insider attacks. *Information Systems Security*, 15(5):39–47.

[Mattos e Duarte 2012] Mattos, D. M. F. e Duarte, O. C. M. B. (2012). QFlow: Um sistema com garantia de isolamento e oferta de qualidade de serviço para redes virtualizadas. Em *XXX SBRC'12*.

[Mattos et al. 2013] Mattos, D. M. F., Ferraz, L. H. G. e Duarte, O. C. M. B. (2013). Um mecanismo para isolamento seguro de redes virtuais usando a abordagem híbrida Xen e OpenFlow. Em *XIII SBSeg'13*, páginas 128–141.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. e Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *SIGCOMM Computer Communication*.

[Nagahama et al. 2012] Nagahama, F. Y., Farias, F., Aguiar, E., Gaspary, L., Granville, L., Cerqueira, E. e Abelém, A. (2012). IPSFlow uma proposta de sistema de prevenção de intrusão baseado no framework openflow. Em *III WPEIF-SBRC'12*, páginas 42–47.

[Pfaff et al. 2009] Pfaff, B., Pettit, J., Koponen, T., Amidon, K., Casado, M. e Shenker, S. (2009). Extending networking into the virtualization layer. Em *8th ACM Workshop on Hot Topics in Networks-HotNets*. Citeseer.

[Porras et al. 2012] Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M. e Gu, G. (2012). A security enforcement kernel for OpenFlow networks. Em *Proceedings of the first workshop on Hot topics in software defined networks*, páginas 121–126. ACM.

[Radware 2014] Radware (2014). DefenseFlow resources. `http://www.radware.com/Products/DefenseFlow/`. Acessado em abril de 2014.

[Shin et al. 2013] Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G. e Tyson, M. (2013). FRESCO: Modular composable security services for software-defined networks. Em *Proceedings of Network and Distributed Security Symposium*.

[Siris e Papagalou 2006] Siris, V. A. e Papagalou, F. (2006). Application of anomaly detection algorithms for detecting syn flooding attacks. *Computer communications*, 29(9):1433–1442.

[Sommer 2003] Sommer, R. (2003). Bro: An open source network intrusion detection system. Em *DFN-Arbeitstagung über Kommunikationsnetze*, páginas 273–288.

[Sommer e Paxson 2010] Sommer, R. e Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. Em *IEEE Symposium on Security and Privacy*, páginas 305–316.

[Xing et al. 2013] Xing, T., Huang, D., Xu, L., Chung, C.-J. e Khatkar, P. (2013). SnortFlow: A OpenFlow-based intrusion prevention system in cloud environment. Em *2nd GENI Research and Educational Experiment Workshop*, páginas 89–92.