Localização Eficiente de Sensores Colaborativos para Detecção e Prevenção de Intrusão em Ambientes Virtualizados*

Martin Andreoni Lopez , Diogo Menezes Ferrazani Mattos, Lyno Henrique Gonçalves Ferraz e Otto Carlos Muniz Bandeira Duarte

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

{martin, menezes, lyno, otto}@gta.ufrj.br

Abstract. *Internal users are the main causes of anomalous and suspicious behavior of a network. Even when traditional security middleboxes are present, internal attacks can lead the network to an outage or to a leakage of security information. This paper introduces BroFlow+, an Intrusion Detection and Prevention System (IDPS) that employs multiple sensors which collaborate in threat detection. BroFlow+ enhances the limited-view mechanism that works isolated, by a global network view, based on a collaborative-knowledge distributed scheme. The system provides an optimal sensors placement, maximizing network coverage using the minimal sensor number. The proposal uses Network Function Virtualization (NFV) and Software Defined Networking (SDN) to dynamically manage the sensors. BroFlow+ contributions are: i) an IDPS following the NFV model using high performance Bro tool sensors; ii) a heuristic for optimal location with the minimum number of sensors in the network; and iii) a systematical performance sensors analysis in a virtual environment. The results show that our proposal significantly reduces the sensors number, while keeping full coverage of network flows.*

Resumo. *Os usuários internos são os principais causadores de comportamentos suspeitos e anômalos em redes. Mesmo quando sistemas intermediários de segurança estão presentes, ataques internos podem levar a uma interrupção da rede ou vazamento de informações seguras. Esse artigo apresenta BroFlow+, um Sistema de Detecção e Prevenção de Intrusão (SDPI) que utiliza múltiplos sensores para a colaboração na detecção de ameaças. BroFlow+ melhora o mecanismo de funcionamento isolado com visão limitada da rede, por uma visão de rede global, com base em um esquema de conhecimento colaborativo distribuído. O sistema prove uma localização otimizada dos sensores colaborativos maximizando a cobertura da rede utilizando o menor número de sensores possível. A proposta utiliza a Virtualização de Funções de Rede (VFR) e as Redes Definidas por Software para gerenciar dinamicamente os sensores. As contribuições do BroFlow+ são: i) um IDPS que segue o modelo função de virtualização de rede com sensores Bro de alto desempenho; ii) uma heurística para localização otimizada com o mínimo número de sensores na rede; e iii) uma análise sistemática do desempenho dos sensores em um ambiente virtualizado. Os resultados obtidos mostram que a proposta reduz significativamente o número de sensores, enquanto mantém a cobertura total dos fluxos da rede.*

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

1. Introdução

O crescente número de ameaças e ataques aos sistemas de comunicação atuais, demonstram que os procedimentos tradicionais de segurança são ineficazes para coibir os comportamentos maliciosos que exploram as vulnerabilidades da rede. Quaisquer comportamentos maliciosos ou ataques podem dar origem a grandes desastres, afetando a confiabilidade, a integridade e a disponibilidade dos sistemas. Esses desastres podem não só ocorrer na Internet, como também, em sistemas de computação em nuvem ou redes de infraestruturas críticas, tais como as Redes Elétricas Inteligentes [Guimarães et al. 2013]. Assim, os Sistemas de Detecção e Prevenção de Intrusão (SDPI) (*Intrusion Detection and Prevention Systems* - IDPS) são ferramentas fundamentais na detecção de ataques internos e na identificação de novas vulnerabilidades [Liao et al. 2013]. No entanto, os SDPI tradicionais, como as ferramentas Snort e Bro, funcionam de maneira isolada ou com visão parcial da rede, dificultando a detecção e prevenção de ameaças que são até então desconhecidas para esse sistema ou de ataques que são distribuídos. Os Sistemas IDS colaborativos ou Redes IDS, são uma solução para esse problema. Os IDS colaborativos possuem diferentes sensores espalhados na rede que compartilham conhecimento e experiências entre eles. Isso aumenta a precisão geral da avaliação de uma intrusão, bem como a capacidade de detectar novos tipos de intrusão [Fung e Boutaba 2013].

O Future Internet Testbed with Security (FITS) [Moraes et al. 2014] é um ambiente propício para desenvolvimento e testes de mecanismos de segurança usando redes definidas por *software* e redes virtuais [Mattos e Duarte 2014]. Nessa plataforma de experimentação, atualmente funciona a ferramenta de detecção e prevenção de intrusão BroFlow [Andreoni Lopez et al. 2014]. BroFlow segue a abordagem tradicional de IDS com sensores que não se comunicam, onde o conhecimento de um ataque ou ameaça fica restrito à área de cobertura do sensor.

Este artigo propõe o BroFlow+ um sistema distribuído e colaborativo de detecção de intrusões. O sistema BroFlow+ estende o sistema BroFlow, proposto por Andreoni Lopez e Duarte para a detecção de intrusão em redes virtualizadas. O BroFlow+ usa os sensores BroFlow de forma distribuída comunicando-los com um controlador OpenFlow centralizado para a detecção de ataques distribuídos. Na proposta BroFlow+ o sensor BroFlow foi aperfeiçoado para apresentar maior desempenho e é proposta uma heurística para a determinação do número mínimo de sensores e a localização ótima dos sensores IDS colaborativos em ambientes virtuais. A heurística proposta baseia-se na colocação de sensores em nós com maior quantidade de tráfego. A proposta faz uso das ferramentas de Virtualização de Função de Rede e Redes Definidas por *Software* para gerenciar dinamicamente os sensores.

O objetivo do BroFlow+ é realizar a comunicação entre sensores aplicando o conceito de IDS colaborativo. Contudo, as propostas de IDS colaborativo não atacam o problema da localização otimizada dos sensores de tráfego. O BroFlow+ modela e propõe uma heurística para a localização otimizada de sensores para um IDS colaborativo, cujos objetivos são reduzir o número de sensores usados, maximizando a cobertura da rede.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. O sistema proposto é detalhado na Seção 3. O problema é formulado na Seção 4. A Seção 5 discute os resultados. Por fim, a Seção 6 conclui o artigo e apresenta os trabalhos futuros.

2. Trabalhos Relacionados

A segurança e a privacidade em Redes Definidas por Software e em sistemas virtualizados, como os ambientes de Computação em Nuvem, são reconhecidamente frágeis, sendo hoje focos principais de estudos no meio acadêmico e industrial. Assim, há diversas propostas que visam melhorar a segurança em sistemas de redes virtualizados e Redes Definidas por Software.

O sistema XenFlow [Mattos e Duarte 2014] propõe o isolamento de recursos em uma rede SDN funcionando sobre a interface de programa de aplicação (*Application Programming Interface* –API) OpenFlow. O sistema XenFlow fornece segurança para ataques entre múltiplos inquilinos que compartilham uma mesma infraestrutura física da rede, como por exemplo, ataques de Negação de Serviço ou *Denial of Service* – (DoS). Ainda relacionado a isolamento, 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 bisbilhotagem (*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 DoS interno a uma rede virtual. Sistemas de Detecção e Prevenção de Intrusão são mandatórios e vêm sendo propostos para detectar ataques internos em SDN e Redes Virtuais.

Shanmugam *et al.* propõem um sistema distribuído de detecção e prevenção de intrusão para nuvem [Shanmugam et al. 2014]. Este artigo se serve das Redes Definidas por Software para distribuir os sensores de detecção num ambiente universitário. Os recursos são providos de forma elástica de acordo com uma maior ou menor demanda de processamento de pacotes correspondentes a diversos ataques.

Alruwaili e Gulliver propõem um sistema colaborativo de detecção e prevenção de intrusão na proteção dos serviços da computação em nuvem [Alruwaili e Gulliver 2014]. Neste artigo, existe uma camada de virtualização que monitora todos os recursos dos sensores que são instanciados em máquinas virtuais (MV). Além disso, as informações recolhidas pelos diferentes sensores espalhados na rede são enviadas a uma base de dados a qual pode ser consultada por todos os sensores. No entanto, essa proposta não faz nenhum estudo sobre a localização estratégica dos sensores para obter um consumo mínimo de recursos com a máxima cobertura da rede.

Uma das primeiras pesquisas na localização de sensores de detecção em pontos críticos de uma rede para a detecção de ataques distribuídos de negação de serviço (*Distributed Denial of Service* - DDoS) é analisada por [Islam et al. 2008]. A proposta minimiza o número dos nós sensores na rede. A heurística usada é baseada na quantidade de saltos, também chamado distância entre o nó atacante e o nó sensor detector. A proposta calcula o número mínimo de sensores para a detecção o mais perto possível das fontes atacantes num ataque distribuído. A desvantagem dessa proposta é o fato de o cálculo requisitar o conhecimento prévio da localização das fontes atacantes. Além disso, o algoritmo faz unicamente detecção de ataques de DDoS e não apresenta uma análise em relação ao comportamento dos sensores num ambiente virtual.

O problema da localização específica também existe nas Redes Definidas por *Software* para a localização e o número de controladores [Mattos et al. 2015, Muller et al. 2014]. Mattos *et al.* propõem um controlador distribuído para redes Redes Definidas por *Software* com consistência forte no plano de controle e localização otimizada dos controladores. Para tanto os autores utilizam a meta-heurística de arrefecimento simulado ou *Simulated Annealing*. Chen *et al.* propõem o cálculo da localização de sensores de IDS distribuídos através de uma técnica de otimização baseada em algoritmos genéticos (AG) [Chen et al. 2009]. O algoritmo desenvolvido tem como heurísticas a minimização do número de sensores e a maximização da taxa de detecção. Bouet *et al.* também realizam a otimização por AG e analisam o posicionamento dos sensores de inspeção profunda de pacotes (*Deep Packet Inspection* - DPI) virtualizados [Bouet et al. 2013]. A proposta minimiza o número de sensores e a carga analisada por cada nó. No entanto, essas propostas baseadas em AG requerem um tempo elevado de processamento para obter resultados, não garantindo a convergência devido ao uso do algoritmo genético [Ferraz et al. 2014].

3. O Sistema BroFlow+

O modelo de virtualização de redes consiste na flexibilização e na provisão de maior programabilidade de controle e gerência da rede através da separação entre o uso geral da realização física, o *hardware* de rede, e o uso específico do controle e da gerência que são realizadas em *software*. Assim, nas redes virtuais, um roteador físico hospeda diversos roteadores virtuais e uma rede física hospeda diversas redes virtuais. O modelo de Redes Definidas por *Software*, em particular, propõe a separação do plano de dados e o plano de controle da centralização lógica do controle. Os dispositivos de encaminhamento recebem informações pela chamada *Southbound API*, com sua implementação mais conhecida OpenFlow [Sezer et al. 2013]. A troca de informações entre o controlador com as aplicações é feita mediante a *Northbound API*.

Na Virtualização das Funções de Redes, as funções ou serviços são executados em máquinas virtuais sobre *hardware* convencional, de uso geral, evitando a complexidade e os custos das implementações reais [Xilouris et al. 2014]. Serviços de redes tais como comutadores, sensores IDS, firewalls, entre outros, podem ser instanciados sobre uma máquina virtual e migrados para outro equipamento de rede físico remoto, como por exemplo, para um Centro de Dados geograficamente distante.

Na Figura 1 é exemplificado o esquema de prestação do serviço das funções de redes. O esquema possui três atores principais: os usuários da rede, os provedores das funções de rede e os provedores da infraestrutura de rede. Uma vez que o usuário necessita de uma função de rede como, por exemplo, um *firewall*, realiza um pedido a uma interface de *front-end*. Os provedores de funções de rede recebem a solicitação e constroem as funções de rede requisitadas. Como consequência, existe um pedido de alocação de recursos aos provedores da infraestrutura. Quando os recursos são disponibilizados, o usuário final recebe a implementação da função.

Um aspecto importante da implementação das Funções de Virtualização de Redes é o processo de orquestração. O orquestrador gerencia e fornece recursos sob demanda automatizando e coordenando os processos e os serviços requeridos. Além disso, o orquestrador permite um monitoramento das funções, da infraestrutura física e também é

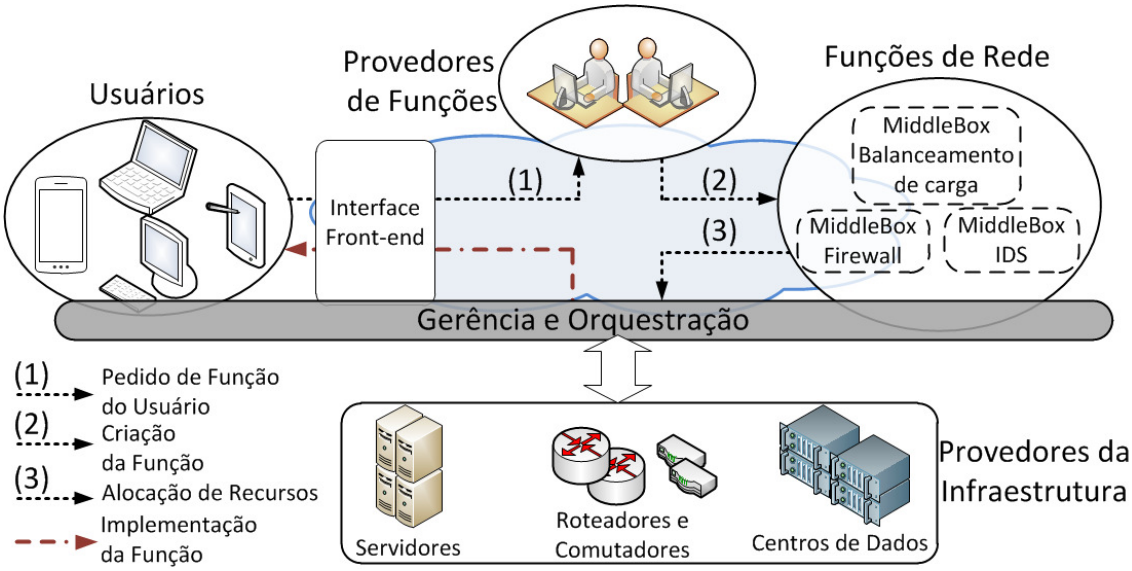


Figura 1. Sequência de ações para disponibilização de uma função de rede no modelo de da Virtualização de Funções de Redes. 1) Pedido de uma função pelo usuário. 2) Criação da função pelo provedor de função de rede. 3) Alocação de recursos na infraestrutura física. 4) Implementação da função pelo orquestrador.

quem gerencia o processo de faturação. Essa camada é localizada acima dos provedores da infraestrutura, e deve ser capaz de gerenciar diferentes fornecedores de infraestrutura. Assim, a camada do orquestrador é executada no plano de controle SDN para controlar o gerenciamento dos recursos de rede.

O objetivo principal do BroFlow+ é a colocação estratégica de sensores IDS de forma a obter uma visão global da rede e com isto atingir melhores resultado devido a um procedimento colaborativo em Redes Definidas por Software (*Software Defined Networking* - SDN). Uma versão otimizada da ferramenta Bro, com acelerador tratamento de pacotes através memória em anel no *kernel*, é utilizada para inspecionar o tráfego mediante diversos sensores. Esses sensores são localizados estrategicamente gerando alarmes quando uma anomalia é detectada. Uma aplicação rodando no controlador de rede OpenFlow recebe esses alarmes e aciona contramedidas para bloquear o ataque de maneira global. O BroFlow+ utiliza a visão global fornecida pelo OpenFlow para bloquear ataques. O sistema é baseado em um ambiente de virtualização de redes híbrido, composto por Máquinas Virtuais (MV) Xen executando sobre uma matriz de comutação OpenFlow. As MV se interconectam através de comutadores OpenFlow, implementados pelo comutador programável por *software* Open vSwitch (OVS), instanciados nas máquinas físicas. A configuração e o controle dos comutadores OpenFlow é realizada pelo controlador POX.

No sistema BroFlow+ existem dois tipos de sensores, como mostra a Figura 2. Os sensores BroFlow de Rede Virtual (RV) e o sensor BroFlow da Infraestrutura. Os sensores de RV monitora tanto os roteadores virtuais como uma estação específica. Em cada um dos sensores de Rede Virtual 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.

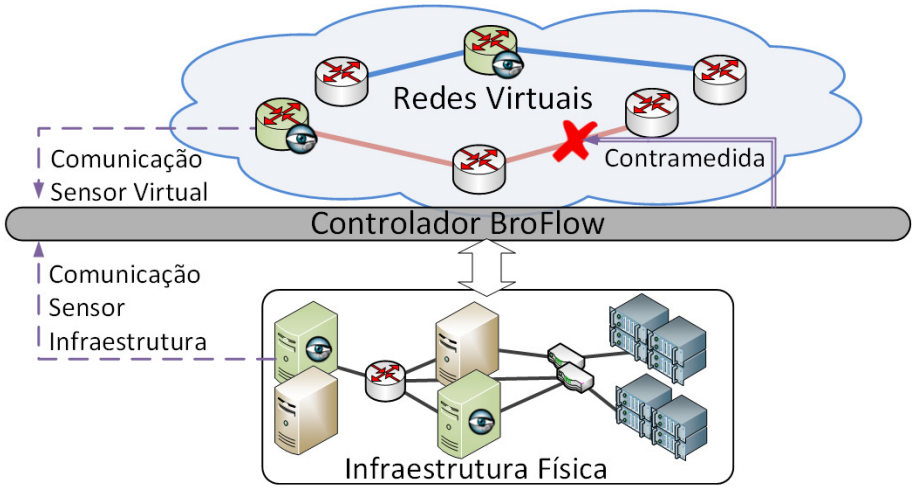


Figura 2. Arquitetura de Sistema. A comunicação existe entre os sensores virtuais ou físicos e o controlador BroFlow. Quando um ataque é detectado, o controlador envia uma mensagem de contramedida para bloquear o ataque.

O Sensor BroFlow de Infraestrutura é colocado paralelamente ao controlador para proteger a infraestrutura física da rede e as redes virtuais que ela hospeda. Assim, o BroFlow fornece segurança tanto para os provedores da infraestrutura como aos usuários das redes virtuais. Logo, o controlador BroFlow, atuando como orquestrador, cumpre o requisito de fornecer o serviço de segurança para todas as redes que ele monitora.

Cada um dos sensores BroFlow contém um *daemon* da ferramenta de código aberto para o monitoramento e análise em tempo real do tráfego de rede Bro [Sommer 2003]. As políticas BroFlow+ são implementadas diretamente no Bro, sendo abstraídos os algoritmos de detecção em políticas descritas na linguagem `Bro`.

O BroFlow+ utiliza como base a detecção e as contramedidas utilizadas no BroFlow anteriormente apresentado, contribuindo com a possibilidade de distribuir nós pelo modelo NFV. Para isso é utilizada a heurística selecionando os nós com maior cobertura da rede. O comportamento dos sensores BroFlow é assumido como benigno. Porém, a colaboração é feita de modo em que são trocadas mensagens de eventos anômalos entre os sensores. A comunicação é feita pela biblioteca `broccoli`, uma abreviatura de *Bro Client Communication Library*. Esta biblioteca permite a comunicação entre pares de sensores IDS ou sensores com outras aplicações. Assim, os próprios sensores detectam eventos, e enviam a todos seus pares ou um conjunto deles. Logo, um ataque à infraestrutura física, realizado por Máquinas Virtuais atuando em conluio, pode ser detectado pelos sensores localizados nas redes virtuais e, uma vez que eles se comunicam com o sensor da infraestrutura física, uma contramedida pode ser tomada para evitar o ataque. Um sensor que esteja monitorando uma máquina física pode comunicar a distintos sensores na rede o evento ou estado da máquina. Esse evento pode não corresponder a um ataque de maneira local, mas pode ser um ataque distribuído por *bots*, sendo detectado rapidamente na rede vista como um todo. Portanto, é utilizado o conceito de colaboração por pares, diferenciado de outras ferramentas IDS que utilizam modelos hierárquicos [Kholidy et al. 2013].

Outro cenário possível de ser analisado é no qual os sensores IDS não possuem confiança plena em todos os nós. Dessa forma, é necessário um modelo de confiança para avaliar quais informações trocadas pelos sensores são comprometidas ou sofreram ataques de homem no meio (*Man in the Middle*). Esse modelo serve para evitar que um nó sensor atacado envie informações falsas ou tendenciosas aos seus pares. Uma alternativa a esse problema é proposto em [Fung e Boutaba 2013]. No entanto, nessa proposta não é analisado o comportamento em um ambiente virtual.

4. Modelagem e Otimização do Problema de Colocação Estratégica de Sensores IDPS

Essa seção modela formalmente o problema de colocação de sensores na rede e provê uma heurística mediante a metodologia da solução gulosa no problema de localização, minimizando o número de sensores e maximizando a cobertura atingida por cada um deles.

Seja um grafo finito conexo $G = (N, A)$, onde N é o conjunto não vazio de nós e A é o conjunto de arestas não direcionadas de G.

Seja $n \in N$ e $s \in S \subseteq N$, onde S é o conjunto de nós nos quais serão implementados os sensores IDS colaborativos. Além disso, definimos uma função auxiliar $X(n)$ que determina se um nó n possui um sensor

$$X(n) = \begin{cases} 1, & \text{se } n \in S \\ 0, & \text{caso contrario} \end{cases} \tag{1}$$

Seja t_{ij}^n a parcela de tráfego da origem i para o destino j que passa pelo nó n. Assim, o total de tráfego que passa pelo nó n é $t^n = \sum_{i \neq j} t_{ij}^n$, e o total de tráfego de i até j é $t_{ij} = \sum_{n \in N} t_{ij}^n$. Caso exista um sensor em n, ele é capaz de analisar todo o tráfego t^n. Dessa maneira, o total de tráfego analisado pelos sensores IDS é $T = \sum_{n \in N} \sum_{i \neq j \in N} X(n) t_{ij}^n$, ou de forma equivalente $\sum_{n \in S} \sum_{i \neq j \in N} t_{ij}^n$.

A função objetivo $F(x)$, representa o custo global a minimizar, composta por dois custos: o número de sensores na redes e a cobertura do tráfego da rede analisado por cada nó para a detecção de um ataque. Logo,

$$\min F(S) = F_{sensor}(S) + F_{traf}(S) \tag{2}$$

onde $F_{sensor}(S)$ é a função da relação entre os nós sensores e o total de nós na rede que é expresso por

$$F_{sensor}(S) = \frac{1}{|N|} \sum_{n \in N} X(n) \tag{3}$$

e $F_{traf}(S)$ é uma função da porcentagem de tráfego em relação ao total do tráfego da rede que cada nó sensor analisa

$$F_{traf}(S) = \frac{\sum_{n \in N} \sum_{i \neq j \in N} t_{ij}^n X(n)}{\sum_{n \in N} \sum_{i \neq j \in N} t_{ij}^n} \tag{4}$$

de modo que $0 \leq F_{sensor}(S) \leq 1$ e $0 < F_{sensor}(S) + F_{traf}(S) \leq 1$. Esse problema pode ser reduzido ao problema de Conjunto de Cobertura (*Set Covering Problem* - SCP) que é um problema NP-difícil.

Uma maneira eficiente de resolver esse problema é usando o algoritmo guloso. O algoritmo guloso ordena a lista de nós de acordo com a quantidade de caminhos que passam pelos nós. Assim, espera-se que nós centrais que concentram boa parte do tráfego sejam escolhidos primeiro. Esses nós são escolhidos para colocar sensores sucessivamente. Esse procedimento é repetido até que todos os caminhos existentes na rede possuam sensores analisando, ou quando não é possível colocar mais sensores. Assim, nesse artigo é utilizado o algoritmo guloso para encontrar a máxima cobertura de rede possível com o mínimo número de sensores de maneira rápida e eficiente. Como heurística é utilizado o cálculo da quantidade de caminhos nos quais o nó está presente em relação ao total de caminhos é definida por $C(n)$

$$C(n) = t^n = \sum_{i \neq j} t_{ij}^n \tag{5}$$

Além disso, definimos uma métrica para medir a porcentagem de caminhos que o conjunto de sensores é capaz de analisar. Para isso, definimos uma função $Y_{i,j}(n)$ que determina se os tráfego de um par de nós i, j é analisado por n

$$Y_{i,j}(n) = \begin{cases} 1, & \text{se } X(n) = 1 \text{ e } t_{ij}^n > 0; n \in N \\ 0, & \text{caso contrario} \end{cases} \tag{6}$$

Assim, a métrica porcentagem de tráfego analisado é H

$$H(S) = \frac{\sum_{i \neq j \in N} Y_{i,j}(n)}{\sum_{i \neq j \in N} t_{ij}} \tag{7}$$

Onde a Equação 7 relaciona a quantidade de caminho que cada nó sensor colaborativo analisa em relação ao total de caminhos na rede. O valor obtido para cada nó n mostra o percentual de caminhos inspecionados pelo nó.

O algoritmo 1 mostra o calculo da heurística proposta. Primeiramente como entrada do algoritmo o valor de cobertura alvo H^* de rede a analisar. Em seguida, é calculada a Árvore de Cobertura Mínima do grafo G para evitar laços. Finalmente é calculada a heurística para cada nó utilizando a Equação 7. Quando o nó n é escolhido como sensor IDS é parcialmente removido da topologia, sendo adicionado no conjunto de nós S.

5. Experimentações e Resultados

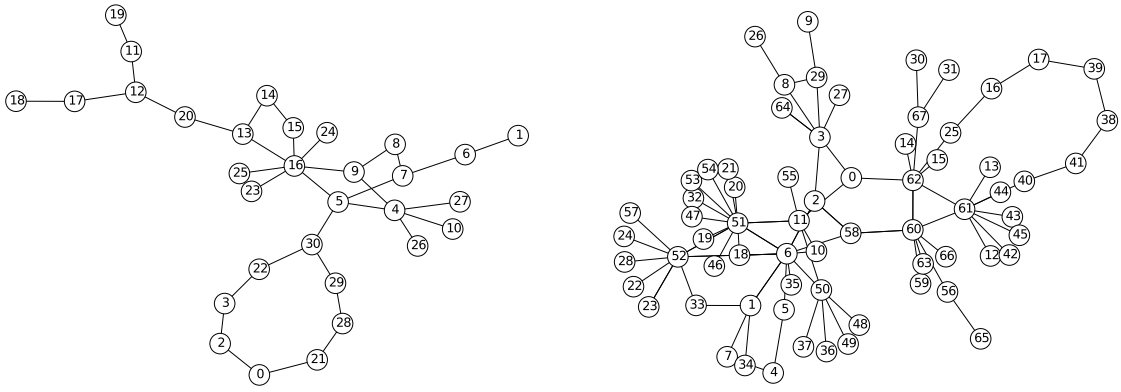
Para avaliar a proposta foram realizadas as seguintes experimentações: avaliação da heurística mediante simulação em distintas topologias de diferentes tamanhos; análise sobrecarga de processamento dos sensores de Detecção de Intrusão. Os experimentos foram realizados em um servidor com processador Intel Xeon X5690 com 24 núcleos, com frequência de 3.47GHz de *clock* e com 48 GB de memória RAM.

Algoritmo 1: Cálculo do número mínimo de sensores.

Seja G grafo finito conexo
Calcule $ACM(G)$, nós da *Árvore de Cobertura Mínima*
Entrada: H^* valor da cobertura da rede a obter
calcular $C(n), \forall n \in ACM(G)$
ordene $ACM(G)$ por $C(\cdot)$
for $n \in ACM(G)$ **do**
    calcular H
    if $H \leq H^*$ **then**
        remover nó n da topologia;
        adicionar n ao S
    end
end

5.1. Avaliação do Algoritmo de localização de Sensores

Para avaliar a heurística em relação ao número e a localização dos sensores de IDS foram usadas duas topologias reais obtidas do *topology zoo* [1]. A primeira topologia analisada é o *backbone* de internet da Rede Nacional de Ensino e Pesquisa (RNP), que possui 31 nós com 34 enlaces espalhados geograficamente em todos os estados do Brasil. A segunda topologia foi a Rede de Ciência e Energia *Energy Sciences Network* ou ESnet. Essa é uma rede de alta velocidade do Departamento de Energia dos Estados Unidos e é gerenciada por uma equipe no Lawrence Berkeley National Laboratory. Essa topologia tem conectividade entre 68 nós e 92 enlaces diferentes dentro dos Estados Unidos.



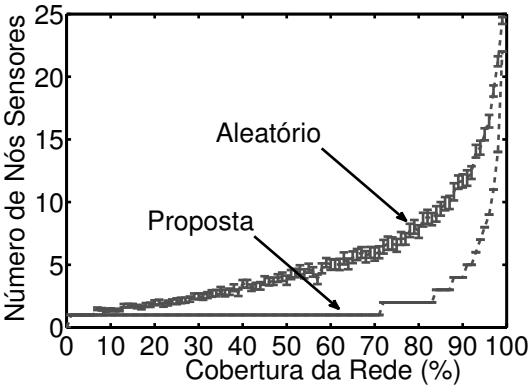
(a) Topologia RNP com 31 nós e 34 enlaces.

(b) Topologia ESnet com 68 nós e 92 enlaces.

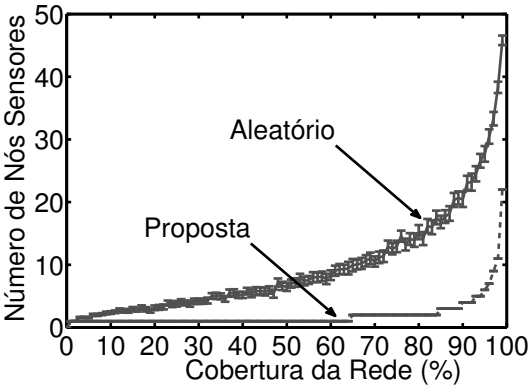
Figura 3. Topologias reais usadas para a avaliação da eficiência do sistema proposto em relação ao cálculo do número mínimo de sensores IDS utilizados para proteger a rede. 1) Topologia da RNP. 2) Topologia da ESnet.

A Figura 4 mostra percentual de tráfego coberto em função do número de sensores IDS calculados pelo sistema proposto. O sistema determina a melhor posição que cada sensor deve ser colocado para garantir a melhor cobertura da rede. Observa-se que o

[1] www.topology-zoo.com



(a) Avaliação da heurística na topologia de 31 nós.



(b) Avaliação da heurística na topologia de 68 nós.

Figura 4. Eficiência do sistema proposto em relação ao número de sensores de IDS necessários para cobrir todo o tráfego da rede.

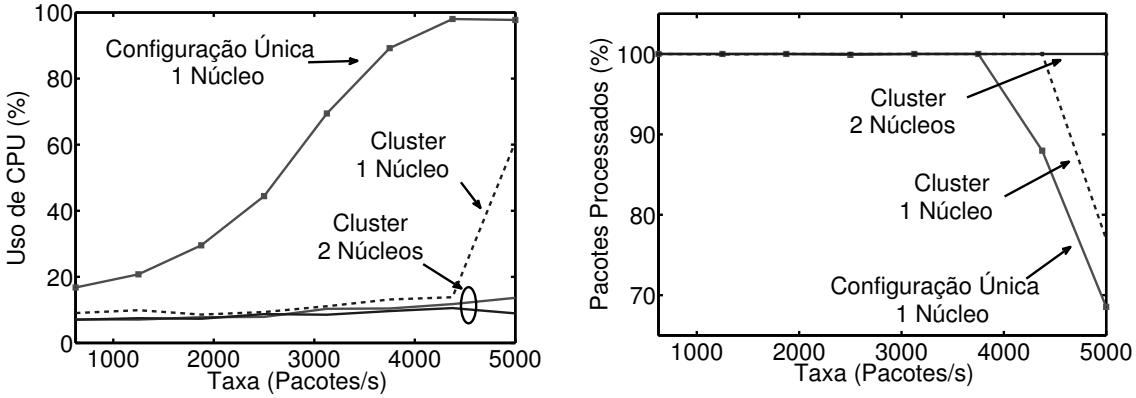
sistema apresenta uma boa eficiência, pois com menos de 10% dos nós sendo usados como sensores IDS mais de 80% do tráfego total da rede já é coberto. Também observa-se que sistema apresentou maior eficiência para a rede com maior número de nós, pois chega-se com poucos sensores a grandes coberturas da rede. Para se visualizar a eficiência do sistema proposto é apresentada a curva de cobertura em relação ao número de sensores usando uma estratégia aleatória. O método aleatório escolhe randomicamente um nó da lista de nós da rede.

A Figura 4(a) que escolha criteriosa de onde colocar o sensor de IDS é fundamental e obtém-se 70% de cobertura de rede com apenas um nó. Já para uma cobertura de rede do 95% são necessários sete nós, enquanto o método aleatório utiliza o dobro de nós, em média um valor de 15 nós.

Na Figura 4(b) apresenta os resultados para a topologia com 67 nós, mais de duas vezes a quantidade de nós considerados na topologia anterior. No entanto é importante destacar que o sistema proposto apresenta uma cobertura de até o 65% do tráfego da rede com apenas um nó sensor de IDS, enquanto uma escolha aleatória necessitaria de dez nós. Já no 95% de cobertura de rede o sistema proposto usa unicamente cinco nós. Para uma cobertura total da rede o sistema proposto requer 22 nós sensores enquanto o aleatório utiliza em média 46 nós. É interessante destacar que o algoritmo proposto é determinístico. O método aleatório, como era de esperar, obtém diferentes saídas para uma mesma entrada. Os resultados obtidos possuem um intervalo de confiança do 95%.

5.2. Avaliação dos Recursos Consumidos pelo Sensor Colaborativo IDS no Ambiente Virtual

Uma vez que os sensores são estrategicamente localizados em pontos específicos da rede, é necessário instanciar a ferramenta Bro neles. Nesta seção é analisado o desempenho de um sensor da ferramenta de análise de tráfego Bro sendo executado em um no sistema BroFlow no ambiente virtual na rede de teste FITS. O sensor Bro do Sistema de Detecção de Intrusão utilizado captura foi configurado na rede e ele analisa em tempo real os pacotes espelhados. Portanto, há uma sobrecarga associada a esta função. Assim, avaliou-se o consumo de recursos da análise de pacotes pelo Sistema de Detecção de In-



(a) Uso de CPU em relação a taxa de pacotes processada pelo nó sensor de IDS implementado com a ferramenta de análise de tráfego Bro em configuração única e *cluster* com um e dois núcleos

(b) Taxa de pacotes analisados pelo sensor de IDS implementado com a ferramenta de análise tráfego Bro na configuração única e *cluster* com um e dois núcleos.

Figura 5. Desempenho de um sensor de IDS implementado com a ferramenta de análise tráfego Bro em um nó da rede de testes FITS. Quando há sobrecarga a ferramenta Bro deixa de analisar alguns dos pacotes mas não deixa de encaminhá-los evitando ser vítima de um ataque de negação de serviço. A porcentagem de pacotes analisados durante a sobrecarga é mostrada.

trusão para se determinar qual dos aspectos, banda ou processamento, é o mais crítico para a detecção do ataque de negação de serviço por inundação. É importante ressaltar que o processamento exigido pela análise dos pacotes depende da política de segurança e de que tipos de ameaças são analisados. A técnica de Inspeção Profunda de Pacotes (*Deep Packet Inspection*–DPI) requer uma quantidade considerável de processamento. Para a avaliação foram geradas taxas crescentes de pacotes TCP-SYN mediante a ferramenta `hping3`, e foi analisado tanto o processamento gasto pela máquina de DPI, quanto o percentual de pacotes analisados pelo sistema Bro. Na avaliação, os sensores foram inicializados executando uma política escrita na linguagem `Bro` para a detecção de pacotes TCP-SYN mediante o algoritmo de limiar adaptativo[Siris e Papagalou 2006].

A ferramenta Bro de análise de tráfego convencional não pode executar em *multi-threading*, ele só usa um único núcleo da CPU, ou pode fazer uso da técnica *multi-threading* mediante a utilização da tecnologia de *cluster* com a melhora no desempenho da biblioteca `PF_RING` em vez da biblioteca nativa `libpcap`. A `PF_RING` funciona como acelerador da velocidade de captura no tratamento de pacotes através de memória em anel no *kernel*, melhorando altamente o desempenho. Assim, o experimento foi avaliado com o uso de ambas tecnologias durante um ataque de DoS diretamente sobre a máquina de análise. Na ferramenta convencional a máquina virtual foi configurada de forma a ter acesso a somente um núcleo, para evitar desperdício de recursos. Já na configuração em *cluster* foram avaliadas o uso de só um núcleo e da mesma maneira com dois núcleos. Como mostra a Figura 5(a) no caso da configuração *cluster* com dois núcleos ele gera dois traços, isto é devido a que na configuração *cluster* o Bro utiliza um núcleo do processador durante a execução. Também é importante ressaltar que nesta técnica foram analisados os consumos dos processos de CPU para os trabalhadores, que são os que inspecionam o tráfego. Não foram analisados os processos do *proxy* e o gerenciador dado que eles

podem ser executados em computadores distintos. Neste experimento foram utilizados as duas técnicas de modo de uso do Bro tanto a configuração única ou *standalone* como a configuração por *cluster*.

A Figura 5(a) mostra o consumo de CPU pelos processos Bro em uma Máquina Virtual. Na configuração com um único núcleo o sistema chega a saturação utilizando todos os recursos de CPU da máquina para aproximadamente 3600 pacotes por segundo. No entanto, o desempenho na configuração em *cluster* é bem melhor que não chega a satura para essa mesma quantidade 3600 pacotes por segundo. Na configuração com dois núcleos o desempenho é muito melhor quase sendo desprezível o aumento de CPU na taxa máxima de 5000 pacotes por segundo. Na Figura 5(b) é mostrado a porcentagem de pacotes analisada pelo Bro. Deve ser ressaltado que um ataque comum a ferramentas de IDS é sobrecarrega-las até a parada do sistema, No entanto, a ferramenta Bro é robusta a ataques de negação de serviço contra si mesma, pois ela deixa de analisar pacotes mas não trava o sistema. Comparando esses valores com os da Figura 5(a), se observa que a quantidade de pacotes analisados sofre uma queda quando o processamento é máximo, este efeito é notório com a configuração de um núcleo único chegando a analisar, no máximo, apenas 70% dos pacotes na taxa de 5000 pacotes por segundos. Os resultados são bem melhores na configuração em *cluster*, já que com um núcleo só o Bro analisa 80% dos pacotes a uma taxa de 5000 pacotes por segundo. Na configuração de com dois núcleos todos os pacotes são analisas a uma taxa de 5000 pacotes por segundo. Os resultados são se semelhantes aos obtidos por Weaver e Sommer [Weaver e Sommer 2007], no entanto a ferramenta BroFlow executa o sistema Bro rodando sobre máquinas Virtuais.

Segundo os desenvolvedores do Bro é possível utilizar a ferramenta em links ethernet de 10 Gigabits[2], para analisar altíssimas taxas de pacotes, uma possível solução que o Grupo de Teleinformática e Automação (GTA) defende para evitar a sobrecarga de processamento nas Máquinas Virtuais que usam IDS é através da elasticidade [Lobato et al. 2014]. Esta proposta, combina os recursos das Virtualização e as Redes Definidas por *Software* para aliviar a sobrecarga das instâncias DPI. Os recursos das Máquinas Virtuais são monitorados constantemente mediante uma aplicação. A aplicação consegue detectar sobrecarga ou descargas das instâncias IDS. Assim a proposta consegue brindar elasticidade ao sistema, criando ou destruindo dinamicamente instâncias de máquinas IDS. O sistema é beneficiado por fazer um uso máximo dos recursos, sem ter recursos ociosos. Também o IDS consegue ser mais robustos já que uma vez que uma sobrecarga de processamento é detectada, uma Máquina Virtual é instanciada e mediante as SDN, o tráfego é distribuído entre as instâncias conseguindo inspecionar uma taxa maior.

6. Conclusão

Nesse artigo foi apresentada o BroFlow+ um sistema de Detecção e Prevenção de Intrusão colaborativo seguindo o concepto das Virtualização das Funções de Rede. O BroFlow+ apresenta um método para a localização estratégica de sensores. Para isso foi desenvolvida uma modelagem matemática obtendo uma heurística que leva em consideração o mínimo número de sensores de intrusão atingido a máxima cobertura da rede. A avaliação da heurística foi analisada em diferentes topologias reais. Os resultados mostram que com uma cobertura alta de rede o sistema proposto tem um grande ganho em

[2]https://www.bro.org/documentation/faq.html

relação à escolha aleatória. Também foi avaliado o desempenho das instâncias dos sensores colaborativos de intrusão sendo executados sobre máquinas virtuais. Além disso, um protótipo do sistema funciona atualmente na rede de testes FITS. Como trabalhos futuros, pretende-se integrar na heurística os requerimentos do Acordo de Nível de Serviço ou *Service Level Agreement* para obter um modelo mais eficiente de otimização orientado ao provedor da infraestrutura na nuvem. Também como foi estabelecido previamente a comunicação e a confiança entre os distintos sensores colaborativos.

Referências

[Alruwaili e Gulliver 2014] Alruwaili, F. F. e Gulliver, A. (2014). CCIPS: A cooperative intrusion detection and prevention framework for cloud services. *International Journal of Latest Trends in Computing*, 4(4):151–158.

[Andreoni Lopez et al. 2014] Andreoni Lopez, M., Figueiredo, U. d. R., Lobato, A. G. P. e Duarte, O. C. M. B. (2014). BroFlow: Um sistema eficiente de detecção e prevenção de intrusão em redes definidas por software. *XXXIV CSBC - WPerformance'14*, páginas 1919–1932.

[Bouet et al. 2013] Bouet, M., Leguay, J. e Conan, V. (2013). Cost-based placement of virtualized deep packet inspection functions in SDN. Em *IEEE Military Communications Conference, MILCOM*, páginas 992–997. IEEE.

[Chen et al. 2009] Chen, H., Clark, J. A., Tapiador, J. E., Shaikh, S. A., Chivers, H. e Nobles, P. (2009). A multi-objective optimisation approach to IDS sensor placement. Em *Computational Intelligence in Security for Information Systems*, páginas 101–108. Springer.

[Ferraz et al. 2014] Ferraz, L. H. G., Mattos, D. M. F. e Duarte, O. C. M. B. (2014). A two-phase multipathing scheme with genetic algorithm for data center network. IEEE Global Communications Conference - GLOBECOM.

[Fung e Boutaba 2013] Fung, C. e Boutaba, R. (2013). Design and management of collaborative intrusion detection networks. Em *Proceedings of the 13th IFIP/IEEE Integrated Network Management Symposium (IM 2013)*, Ghent, Belgium.

[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.

[Islam et al. 2008] Islam, M., Nadeem, K. e Khan, S. (2008). Efficient placement of sensors for detection against distributed denial of service attack. Em *International Conference on Innovations in Information Technology, IIT*, páginas 653–657.

[Kholidy et al. 2013] Kholidy, H., Erradi, A., Abdelwahed, S. e Baiardi, F. (2013). HA-CIDS: A hierarchical and autonomous IDS for cloud systems. Em *V International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, páginas 179–184.

[Liao et al. 2013] Liao, H.-J., Lin, C.-H. R., Lin, Y.-C. e Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16 – 24.

[Lobato et al. 2014] Lobato, A. G. P., Figueiredo, U. d. R., Andreoni Lopez, M. e Duarte, O. C. M. B. (2014). Uma arquitetura elástica para prevenção de intrusao em redes virtuais usando redes definidas por software. *XXXII SBRC'14*, páginas 427,440.

[Mattos e Duarte 2014] Mattos, D. M. F. e Duarte, O. C. M. B. (2014). XenFlow: Seamless migration primitive and quality of service for virtual networks. Em *Global Communications Conference (GLOBECOM), 2014 IEEE*, páginas 2326–2331, Austin, Texas, USA.

[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.

[Mattos et al. 2015] Mattos, D. M. F., Lopez, M. E. A., Ferraz, L. H. G. e Duarte, O. C. M. B. (2015). Controlador resiliente com distribuição eficiente para redes definidas por software. Em *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2015*.

[Moraes et al. 2014] Moraes, I. M., Mattos, D. M., Ferraz, L. H. G., Campista, M. E. M., Rubinstein, M. G., Costa, L. H. M., de Amorim, M. D., Velloso, P. B., Duarte, O. C. e Pujolle, G. (2014). FITS: A flexible virtual network testbed architecture. *Computer Networks*.

[Muller et al. 2014] Muller, L. F., Oliveira, R. R., Luizelli, M. C., Gaspary, L. P. e Barcellos, M. P. (2014). Survivor: an enhanced controller placement strategy for improving SDN survivability. Em *Global Communications Conference (GLOBECOM), 2014 IEEE*, Austin, Texas, USA.

[Sezer et al. 2013] Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M. e Rao, N. (2013). Are we ready for SDN? implementation challenges for software-defined networks. *Communications Magazine, IEEE*, 51(7):36–43.

[Shanmugam et al. 2014] Shanmugam, P. K., Subramanyam, N. D., Breen, J., Roach, C. e Van der Merwe, J. (2014). DEIDtect: towards distributed elastic intrusion detection. Em *Proceedings of the 2014 ACM SIGCOMM workshop on Distributed cloud computing*, páginas 17–24. ACM.

[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.

[Weaver e Sommer 2007] Weaver, N. e Sommer, R. (2007). Stress testing cluster Bro. Em *DETER workshop*, páginas 1–4.

[Xilouris et al. 2014] Xilouris, G., Trouva, E., Lobillo, F., Soares, J., Carapinha, J., McGrath, M., Gardikis, G., Paglierani, P., Pallis, E., Zuccaro, L. et al. (2014). T-NOVA: A marketplace for virtualized network functions. Em *IEEE European Conference on Networks and Communications (EuCNC)*, páginas 1–5. IEEE.