AS-Defender: Mitigando o sequestro de prefixos IP no entorno de um AS de trânsito

Marcio Vinicius de Queiroz Santos[1], **Sidney Cunha de Lucena**[1]

[1]Programa de Pós-Graduação em Informática Aplicada
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro, RJ - Brasil

{marciovinicius.santos, sidney}@uniriotec.br

Abstract. *The BGP protocol, despite its importance, was conceived without any security mechanism to guarantee the authenticity of an advertisement. Thus, attackers can deviate Internet traffic by advertising fake routes, performing the attack known as prefix hijacking. On the other hand, software-defined networks have conferred great power of innovation to network administrators through programmability and logically-centralized control of the network. This work presents AS-Defender, a SDN-based solution that aims to combat IP prefix hijacking surrounding an Autonomous System, by detecting and mitigating attacks targeting its neighbors. Our experiments showed the effectiveness of the solution and how variables present in an inter-domain communication environment can influence its efficiency.*

Resumo. *O protocolo BGP, apesar da sua importância, foi concebido sem qualquer mecanismo de segurança para garantir a autenticidade de um anúncio. Diante disso, atacantes podem desviar o tráfego na Internet mediante o anúncio de rotas falsas, realizando o ataque conhecido como sequestro de prefixo. Por outro lado, as redes definidas por software (SDN) têm conferido um grande poder de inovação aos administradores de rede, através da programabilidade e do controle logicamente centralizado da rede. Este trabalho apresenta o AS-Defender, uma solução baseada em SDN que visa combater o sequestro de prefixos IP no entorno de um sistema autônomo, detectando e mitigando ataques direcionados a seus vizinhos. Nossos experimentos mostraram a eficácia da solução e como variáveis presentes em um ambiente de comunicação interdomínios podem influenciar sua eficiência.*

1. Introdução

O BGP (*Border Gateway Protocol*) [Traina 1995] é o protocolo de roteamento responsável por propagar informações de alcançabilidade das redes e conectar os sistemas autônomos (AS) na Internet. Entretanto, ao receber um anúncio de prefixo IP, um AS não possui a capacidade de validar a autenticidade da origem desse anúncio. Assim, um atacante com controle de um roteador na malha BGP poderia anunciar prefixos que não lhe pertencem, sequestrando o tráfego a eles destinados para fins ilícitos [Horn 2009]. Com a evolução das pesquisas sobre redes definidas por *software* (SDN) [ONF 2012] surgiram novas perspectivas de inovação, incluindo aquelas ligadas a roteamento interdomínios. A definição do comportamento da rede através da programação de um controlador logicamente centralizado possibilita uma resposta rápida e flexível a eventos, modificando o

funcionamento da rede conforme necessário. Diante disso, ao identificar um sequestro de prefixo, uma aplicação poderá reprogramar os *switches* da rede de modo que o fluxo de dados possa ser direcionado para o caminho desejado.

Baseado nisso, este trabalho propõe o *AS-Defender*[1], um protótipo de aplicação SDN para detectar e mitigar o sequestro de prefixos IP direcionado aos ASes vizinhos de um AS de trânsito que rode essa solução. De forma geral, o protótipo usa as mensagens *UPDATE* do BGP para detectar o sequestro de prefixo e então adotar ações de mitigação na tentativa de redirecionar a maior quantidade de tráfego possível, com destino à faixa IP sequestrada, de volta ao vizinho atacado. Isso é garantido através de uma *thread* que monitora os prefixos anunciados pelos ASes vizinhos, permitindo detectar se algum atacante anunciou um desses prefixos. Verificando-se um sequestro, a mitigação é executada para cada vizinho afetado através da reprogramação da rede segundo o paradigma SDN.

Os experimentos realizados mostraram que um AS de trânsito que faz uso desse protótipo é capaz de proteger seus ASes vizinhos contra o sequestro de prefixo. Além disso, foi estudada a eficiência da solução de mitigação com relação a diversos aspectos possíveis em cenários de ligação entre ASes. Verificou-se que o *AS-Defender* é especialmente interessante para casos onde o AS de trânsito que dele faz uso se comporta de forma similar a um Ponto de Troca de Tráfego (PTT), sendo que os benefícios da mitigação podem se estender para além dos ASes vizinhos.

As demais seções deste trabalho são organizadas da seguinte forma: a Seção 2 discorre sobre os trabalhos relacionados, a Seção 3 descreve com mais detalhes o AS-Defender, a Seção 4 apresenta a validação experimental da proposta, e a Seção 5 traz conclusões e trabalhos futuros.

2. Trabalhos Relacionados

O problema do ataque de sequestro de prefixos perdura por muitos anos e, portanto, há várias iniciativas que visam sua detecção ou sua mitigação [Zhang et al. 2008, Chi et al. 2008]. Verifica-se, no entanto, uma quantidade maior de trabalhos ligados à detecção desse tipo de ataque do que na sua mitigação em si.

Dentre os trabalhos que propõem formas para a detecção do sequestro de prefixos, tem-se o PHAS [Lad et al. 2006]. Trata-se de um sistema de notificação, em tempo real, que alerta ASes previamente registrados sobre o sequestro de seus prefixos. O PHAS examina dados de roteamento BGP coletados por algumas organizações na Internet. Outra iniciativa é o DARSHANA [Balu et al. 2016], uma solução de monitoramento que detecta o sequestro de rotas BGP através de informações coletadas do plano de dados das redes, possuindo mecanismos de redundância que possibilitam essa detecção mesmo em caso de contramedidas adotadas pelo atacante. Esse monitoramento envolve ações como a análise de latência da rede, o cálculo da similaridade dos caminhos e a verificação do atraso de propagação. Um outro exemplo é o ARGUS [Shi et al. 2012], que tem como proposta entregar um sistema ágil que detecte o sequestro de prefixos. O seu mecanismo de detecção faz uso de informações tanto do plano de controle quanto do plano de dados, através de um mecanismo de correlação. A proposta considera que a junção dessas informações sobre diferentes ASes funciona como uma espécie de impressão digital, e assim é possível descobrir se uma mudança de rota foi causada por um ataque.

[1]Disponível em `https://bitbucket.org/marcioviniciussantos/as-defender`

O projeto ARTEMIS [Sermpezis et al. 2016, Sermpezis et al. 2018], proposta que serviu de base para este trabalho, propõe uma forma para detecção e mitigação de ataques de sequestro de prefixo voltados para a proteção do próprio AS que o emprega. A descoberta do ataque é realizada da forma tradicional, através do monitoramento das mensagens *UPDATE* do BGP a partir de fontes públicas de roteamento, como o *Route Views* [University of Oregon 2018] e o *RIPE NCC* [RIPE NCC 2018]. O processo de mitigação envolve outro conceito tradicional: anunciar, automaticamente, os subprefixos da faixa sequestrada para atrair o respectivo tráfego de volta ao AS. Essa estratégia leva em conta o mecanismo de *longest prefix match* usado por padrão nos roteadores da Internet, através do qual rotas para faixas de rede menores - ou seja, contendo prefixos (endereços de rede) maiores - que levem a um mesmo destino são sempre preferidas. O *AS-Defender* usa essas mesmas estratégias do ARTEMIS, porém os processos de detecção e mitigação automática contra ataques de sequestro de prefixo são voltados para a proteção dos ASes vizinhos, ou clientes, do que podemos chamar de "AS defensor". E para tal, o *AS-Defender* se vale do paradigma SDN implementado via *OpenFlow*.

Em relação a outras propostas, o *AS-Defender* diferencia-se por proteger, de forma automática e simultânea, ASes clientes contra ataques de sequestro de prefixo de forma totalmente customizável, graças à programabilidade do paradigma SDN. Além disso, este trabalho mostra como esse mecanismo de defesa pode ser potencializado através do uso de filtros de prefixo.

3. Arquitetura e funcionamento do *AS-Defender*

Por ser um mecanismo de defesa para os sistemas autônomos vizinhos, o *AS-Defender* opera internamente em um AS de trânsito e sua arquitetura possui alguns componentes adicionais para que todo o processo de detecção e de mitigação ocorra. Por simplicidade, denominaremos o AS de trânsito que roda a solução proposta de *AS-Defender*. Na Figura 1 estão dispostos os componentes que envolvem a solução, bem como a sua interação com os demais ASes em um ambiente BGP tradicional. O *AS-Defender* interage com a rede via protocolo OpenFlow e foi implementado na linguagem *python*, fazendo uso do controlador Ryu[2].

A detecção do ataque é feita a partir da recepção dos anúncios BGP que são enviados pelos demais ASes. Isso é possível por meio de um coletor de rotas (usou-se o *ExaBGP*[3]) que encaminha as mensagens *UPDATE* recebidas para o módulo de detecção. Paralelamente, ao se detectar um ataque, instruções são enviadas ao controlador *Ryu* de maneira a alterar as tabelas de fluxo dos *switches* e, assim, mudar o encaminhamento dos pacotes de dados, mitigando o ataque. O *AS-Defender* realiza todo o processo entre a detecção do evento de sequestro de IP e a comunicação com o controlador, de forma que todo o ciclo de detecção/mitigação ocorra adequadamente.

O coletor de rotas é construído a partir da implementação de um servidor *WebSocket* que entrega as mensagens através de uma interface de comunicação *full-duplex* com a *thread* de monitoramento existente no *AS-Defender*. Para coletar as mensagens trocadas entre os ASes, um processo BGP é instanciado onde cada mensagem recebida é formatada em uma estrutura *JSON* por um *parser* e entregue à *thread* de monitoramento.

[2]Disponível em `https://osrg.github.io/ryu/`
[3]Disponível em `https://github.com/Exa-Networks/exabgp`

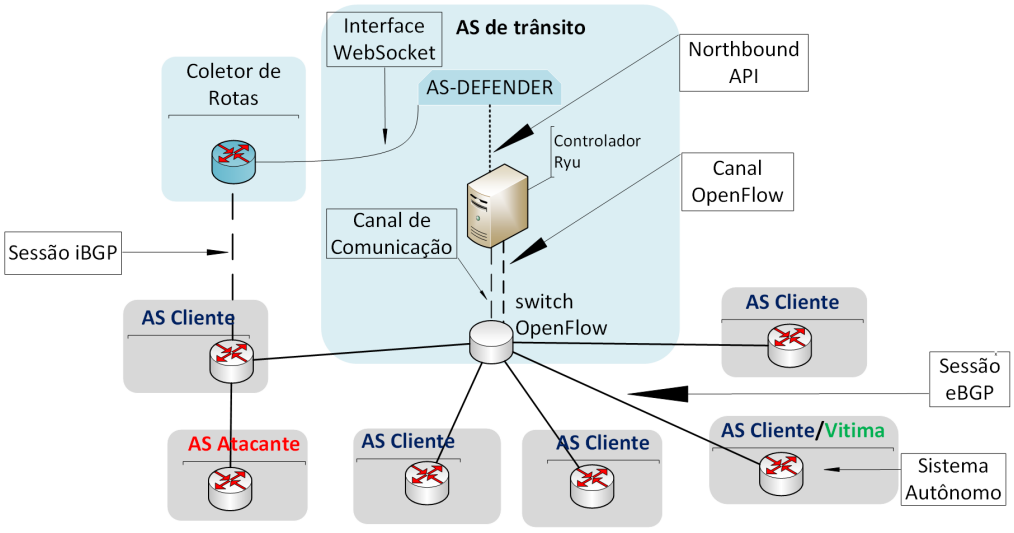

Figura 1. Componentes da arquitetura do *AS-Defender*

O coletor de rotas possui unicamente o papel de representar um mecanismo semelhante ao existente nos serviços de disponibilização de dados de roteamento na Internet, como o *Route Views* ou o disponibilizado pelo *RIPE NCC*. Numa implementação em cenário real, essas informações devem ser obtidas através desses serviços.

A *thread* de monitoramento é criada pelo processo do *AS-Defender* para realizar o monitoramento dos anúncios BGP paralelamente à outras atividades. Ao receber um anúncio, a *thread* verifica se este é legítimo e, caso seja um anúncio falso, informa ao processo principal para que as ações de mitigação sejam executadas.

No controlador, é instanciado um processo BGP que troca mensagens com os demais ASes por meio de um *switch* OpenFlow (ou uma rede de *switches*). As demais instâncias de roteamento na Figura 1 representam ASes vizinhos ao *AS-Defender*, aqui denominados como "clientes". Esses ASes clientes são os consumidores do serviço de proteção contra o sequestro de prefixo.

Entre o controlador e o *switch* existem dois canais de comunicação. Um deles é o canal OpenFlow, usado para executar as operações na tabela de fluxos do *switch*. O outro é um canal de comunicação padrão, por onde trafegam pacotes de outros tipos (por exemplo, ICMP ou BGP) com destino ao próprio *switch* ou aos ASes vizinhos.

3.1. As fases de funcionamento

Para cumprir todo o ciclo de detecção e mitigação do ataque, o funcionamento do *AS-Defender* pode ser compreendido em quatro fases: instanciação, operação, monitoramento e mitigação. A separação dessas fases é em grande parte conceitual, dado que em diversos momentos o protótipo executa atividades de mais de uma fase simultaneamente.

3.1.1. Instanciação

O objetivo da instanciação é tornar o *AS-Defender* preparado para iniciar a comunicação e a troca de mensagens BGP com os seus ASes vizinhos.

Primeiramente, um arquivo *YAML* e outro *JSON* são carregados em memória para

que o protótipo obtenha as informações do *switch OpenFlow* e dos ASes a serem protegidos. Ambos os arquivos são configurados previamente pelo operador. Em seguida, são populadas algumas estruturas de dados referentes às rotas para os ASes vizinhos e são carregadas, no plano de controle, informações necessárias à formação da tabela de encaminhamento dos pacotes na rede. Além disso, são instaladas algumas regras de fluxo proativamente no *switch* com o objetivo de possibilitar a comunicação entre o controlador e os demais ASes. Por fim, o processo BGP é executado no controlador, iniciando-se a troca de mensagens com os vizinhos. As regras de fluxo iniciais instaladas no *switch* têm como objetivo possibilitar a comunicação entre os demais ASes e o *AS-Defender*. É importante ressaltar que, após a instalação das regras de fluxo iniciais, grande parte dos pacotes não são mais encaminhados ao controlador já que o *switch* passa a ter um conjunto de regras capaz de adequadamente encaminhar esses pacotes.

3.1.2. Operação

A etapa de operação é determinada pelo recebimento de mensagens *PacketIn* vindas do *switch*. Ao receber essas mensagens, o *AS-Defender* insere no *switch* regras de encaminhamento conforme as informações armazenadas no plano de controle, para que os próximos pacotes referentes ao respectivo fluxo não sejam enviados novamente ao controlador. Diferentemente do ocorrido na fase de instanciação, tais fluxos possuem tempo de expiração de 10 segundos, o que evita que sejam criadas rotas estáticas para a comunicação entre os ASes e, assim, se contrarie a dinamicidade do funcionamento do BGP. O tempo de expiração foi definido empiricamente após alguns testes, de modo a não produzir um número excessivo de mensagens *PacketIn* por parte do *switch* e, ao mesmo tempo, evitar regras de fluxo com tempo de expiração muito longo. A inserção de fluxos no *switch* é feita de forma reativa, ou seja, apenas na presença de um *PacketIn*.

A etapa de operação permeia todo o ciclo de funcionamento do *AS-Defender* e serve como suporte a qualquer processo executado pelo protótipo, visto que, sem a devida conectividade com os demais ASes, não seria possível formar as sessões eBGP.

3.1.3. Monitoramento

A fase de monitoramento inicia desde o momento em que a aplicação é instanciada e a *thread* de monitoramento é iniciada. Para monitorar o ataque, a *thread* realiza uma conexão *WebSocket* na porta 5000 com o processo servidor executando no coletor de rotas. Cada anúncio na rede BGP recebido pelo coletor é formatado e entregue ao *AS-Defender*, que inicia o seu algoritmo de verificação do ataque. A sequência de funcionamento do monitoramento é ilustrada na Figura 2.

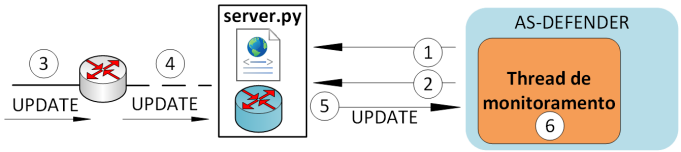

Figura 2. A fase de monitoramento

Primeiramente (1), a *thread* de monitoramento é iniciada e realiza uma conexão com o coletor de rotas. Além disso, (2) são passados parâmetros que especificam qual

tipo de mensagem deve ser entregue à *thread* e sobre quais prefixos de rede. Como o monitoramento é feito com base em anúncios de prefixos, a mensagem especificada é a *UPDATE*, enquanto que o valor referente ao prefixo é passado como 0.0.0.0, significando que o coletor de rotas deve transmitir todas as mensagens *UPDATE* recebidas. A filtragem dos prefixos dos ASes vizinhos fica a cargo do algoritmo de detecção do ataque que é executado a cada anúncio recebido. Em seguida, (3) o anúncio é recebido através de uma sessão eBGP na topologia, (4) encaminhado ao coletor de rotas através de uma sessão iBGP e (5) formatado para ser entregue à *thread*, onde (6) a mensagem é analisada para se detectar um sequestro de prefixo. Maiores detalhes sobre o algoritmo de detecção usado são passados na Seção 3.2.

3.1.4. Mitigação

A fase de mitigação é iniciada quando o monitoramento detecta um ataque direcionado a um prefixo de propriedade de um dos vizinhos. Imediatamente são coletados alguns dados e o mecanismo de mitigação do ataque é acionado conforme as configurações relativas ao vizinho afetado. A mitigação do ataque envolve uma área compartilhada entre os módulos do sistema que serve para a passagem de alguns dados relevantes sobre o ataque.

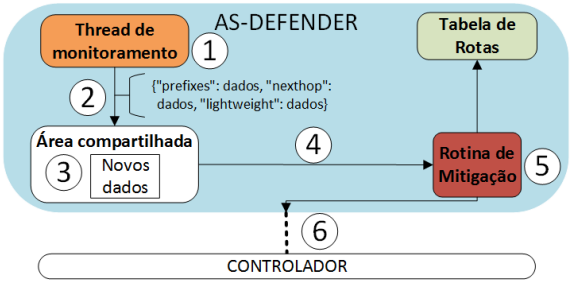

Figura 3. A fase de mitigação

A Figura 3 mostra as etapas da mitigação. Após (1) a detecção de um ataque, (2) os dados sobre os prefixos afetados, o endereço IP do vizinho afetado e o respectivo tipo de mitigação para ele configurado são inseridos na área compartilhada. Através de uma verificação contínua, (3) a rotina de mitigação identifica a presença de novos dados na área compartilhada e (4) os coleta. Em seguida, (5) o processo de mitigação é iniciado e, conforme o tipo de mitigação configurada no cliente, (6) a *Northbound API* do controlador é utilizada. O objetivo principal da fase de mitigação é, na presença de um ataque direcionado a um prefixo de pelo menos um dos vizinhos do *AS-Defender*, influenciar o máximo número possível de ASes a utilizarem um novo caminho para a vítima.

3.2. O mecanismo de detecção

Para detectar o sequestro de prefixos, foi desenvolvido um algoritmo baseado no método de detecção descrito no projeto ARTEMIS [Sermpezis et al. 2018]. Essa detecção é baseada no monitoramento dos anúncios BGP que chegam até o coletor de rotas. O principal componente da mensagem *UPDATE* utilizado nessa detecção é o atributo *AS PATH*. De um modo geral, através do *AS PATH* é verificado se o prefixo de algum dos vizinhos foi sequestrado, qual prefixo foi sequestrado e o AS que sequestrou o prefixo. Essa detecção

não possui falsos positivos, dado que utiliza informações do vizinho inseridas em um arquivo de configuração, comparando-as com o anúncio recebido para identificar a presença ou não de um ataque.

As informações sobre cada vizinho que deve ser protegido são armazenadas em um arquivo *JSON* (*asdefender.json*), formando uma lista de dicionários onde cada item é um vizinho. Esses itens possuem as configurações definidas sobre o prefixo a ser protegido, o tipo de mitigação e outros aspectos sobre os mecanismos de defesa. A Tabela 1 mostra os itens que devem ser configurados para cada vizinho a ser protegido. Para detectar um ataque ao se receber um anúncio, o algoritmo percorre as informações de cada cliente para saber qual deles é o alvo do ataque e realizar as respectivas ações de mitigação.

Campo	**Descrição**
ASN	Número do sistema autônomo protegido
PREFIX	Prefixo de rede que deve ser protegido
NEIGHBORS	Lista com o ASN de todos os vizinhos do cliente
SUBPREFIXES	Lista com os subprefixos utilizados pela rotina de mitigação
MOAS	Contém o ASN do AS-Defender
MOASNEIGHBORS	Contém a lista de ASNs dos vizinhos do AS-Defender
NEXTHOP	IP que o cliente utiliza na sessão eBGP com o AS-Defender
ANNOUNCE	É uma flag utilizada para o mecanismo de mitigação.

Tabela 1. Campos do arquivo *as-defender.json*

O algoritmo de detecção analisa o primeiro e o segundo valor do campo *AS PATH* (os dois mais "à direita" da lista) para identificar se, em alguma das duas posições, há um *ASN* que não possa estar nessas posições. Nesse caso, isso se configura como um anúncio falso direcionado ao prefixo de um dos vizinhos. O *AS-Defender*, em seu arquivo de configuração, possui a informação de todos os ASes que fazem *peering* com os seus vizinhos e, dessa forma, é possível saber quais *ASNs* poderiam estar nas duas primeiras posições do *AS PATH*. Diante dessas informações, a detecção é eficaz em qualquer tipo de topologia e, no caso de uma mudança física nas conexões entre ASes, basta que o AS cliente informe sua nova lista de vizinhos ao administrador do *AS-Defender* para que essas informações sejam atualizadas.

Como o valor mais à direita representa o *AS* que primeiramente anunciou um determinado prefixo, esse valor só pode ser preenchido com o *ASN* do proprietário do prefixo. Seguindo a mesma lógica, o segundo valor deve conter o *ASN* de um dos vizinhos do proprietário. Se alguma dessas condições não for satisfeita para um anúncio relacionado ao prefixo de um dos clientes (conforme são chamados os vizinhos do *AS-Defender*), o mecanismo de detecção alertará um sequestro de prefixo. Para isso, o prefixo de cada cliente é armazenado em uma lista e, para cada anúncio recebido, toda a lista é percorrida para verificar se o anúncio está relacionado a algum dos prefixos protegidos. Caso o anúncio seja referente a um prefixo protegido, o algoritmo de detecção é executado a fim de verificar se o anúncio se trata de um ataque.

Conforme a mitigação configurada para cada cliente, pode ser necessária a realização de um anúncio do prefixo em nome do cliente e, para isso, o algoritmo de

detecção identifica que o anúncio foi emitido pelo próprio mecanismo de mitigação, evitando assim emitir um alerta de ataque decorrente do próprio mecanismo. Essa identificação se dá pela inspeção das mensagens BGP recebidas. Caso na primeira posição do campo *AS-PATH* esteja o ASN do *AS-Defender*, o algoritmo entende ser um anúncio emitido pelo mecanismo de mitigação.

O algoritmo de detecção possui uma sequência de atuação. Inicialmente verifica o número do AS na primeira e, depois, na segunda posição no *AS PATH*. Mesmo que num anúncio exista sequestros de prefixo ligados à primeira e à segunda posição, o gatilho para o mecanismo de mitigação será gerado a partir da primeira inconsistência encontrada, nesse caso, o ataque na primeira posição.

3.3. O mecanismo de mitigação

O objetivo desta mitigação é reduzir ao máximo possível os danos causados pelo ataque. Essa redução deve ser entendida como o ato de proporcionar, ao maior número de ASes possível, o acesso à vítima mesmo após o prefixo dela ter sido sequestrado. De uma forma geral, o mecanismo de mitigação busca redirecionar o tráfego alvo, ou seja, destinado ao prefixo sequestrado, para o AS vítima com o objetivo de manter a disponibilidade de comunicação e acesso aos seus serviços. Após a detecção do ataque, os dados referentes aos campos *SUBPREFIXES*, *NEXTHOP* e *ANNOUNCE*, referentes ao cliente vitimado, são encaminhados para a área compartilhada. Para que o *AS-Defender* saiba qual é o tipo de mitigação a ser realizado, o campo *ANNOUNCE* é verificado. Este campo se comporta como uma *flag*, recebendo os valores "1" ou "0" que habilitam ou desabilitam, respectivamente, a chamada mitigação com anúncio. As duas formas de mitigação, com e sem anúncio, serão descritas a seguir.

3.3.1. Mitigação sem anúncio

A mitigação sem anúncio afeta apenas o tráfego com destino ao prefixo sequestrado que passa pelo *AS-Defender*, redirecionando-o ao AS cliente vitimado. Nesse caso, apenas as informações internas de roteamento do *AS-Defender* são ajustadas para que os dados cheguem corretamente ao destino.

Quando o anúncio falso é visualizado pelo *AS-Defender*, este identifica o ataque e aplica uma correção na sua tabela, retirando a rota falsa e adicionando uma nova rota que direciona o tráfego sequestrado novamente para a vítima. Essa inserção de rota é possível através do uso do campo *NEXTHOP*, recebido pelo mecanismo de mitigação através da área compartilhada. Nenhuma regra de fluxo é inserida no plano de dados, porém, após as regras do *switch* expirarem, este contatará o controlador via *PacketIn*, que responderá com a nova rota correta por meio de um *PacketOut*.

Conforme ilustrado na Figura 4, o AS3 (cliente/vítima) anuncia o prefixo 10.2.0.0/23, de sua propriedade, enquanto o AS1 (atacante) realiza um ataque anunciando o subprefixo 10.2.1.0/24, de propriedade do AS3. Assim que a *thread* de monitoramento detecta o ataque, o mecanismo de mitigação atualiza a sua tabela de rotas, retirando a entrada que tinha o AS2 como caminho para a rede 10.2.1.0/24 e adicionando uma rota para a mesma rede, porém com o AS3 definido como próximo salto. Na prática, o *AS-Defender* corrige a informação da sua própria tabela de encaminhamento e essa correção é propagada até o plano de dados a partir da chegada do próximo *PacketIn* relacionado a

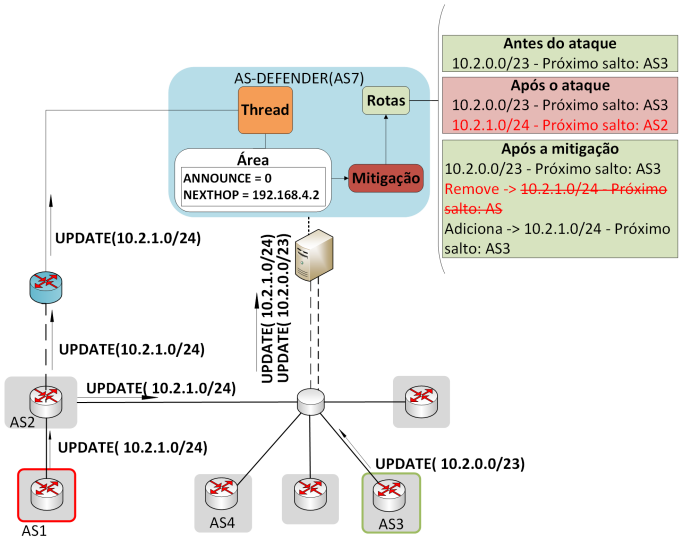

Figura 4. Mitigação sem anúncio

um pacote com destino à vítima.

3.3.2. Mitigação com anúncio

A mitigação com anúncio visa interferir na tabela de encaminhamento dos demais ASes, atraindo o tráfego destinado ao prefixo atacado por meio de um anúncio mais específico que o realizado pelo atacante, beneficiando-se assim do mecanismo de *longest prefix match* adotado pelos roteadores na escolha da melhor rota. Em seguida, ao receber o fluxo de dados correspondente, o *AS-Defender* os entrega diretamente à vítima.

Logo que o ataque é detectado, o mecanismo de mitigação anuncia cada um dos subprefixos (obtidos através do campo *SUBPREFIXES* da Tabela 1) referentes ao cliente vitimado. O anúncio é realizado para todas as sessões BGP mantidas pelo *AS-Defender*, propagando-o em todas as direções possíveis para alterar as tabelas de rotas no maior número possível de ASes.

No exemplo da Figura 5, o AS5 (atacante) anuncia o prefixo 10.2.1.0/24, um subprefixo de propriedade do AS3 (vítima) e obtém parcialmente o tráfego destinado ao AS3. Ao detectar o ataque, o *AS-Defender* anuncia o mesmo prefixo, visto que prefixos mais específicos que /24 são filtrados por roteadores na Internet e, portanto, nesse caso, não seria possível anunciar um prefixo mais específico que o anunciado pelo atacante. Por esse mesmo motivo, ataques a subprefixos maiores que /24 não são uma preocupação, já que o próprio ataque será normalmente filtrado.

O comportamento esperado é mostrado na tabela de rotas do AS2 que, logo após o ataque, passa a ter uma entrada tendo o AS1 como próximo salto para o prefixo sequestrado. Após o recebimento do anúncio de mitigação, uma nova entrada surge nessa tabela apontando o AS7 como próximo salto. Esse fato não garante que o caminho para o AS atacado seja restabelecido em todos os ASes que receberem o anúncio de mitigação, já que isso é determinado pelo processo de seleção do BGP junto com as políticas implementadas em cada AS. Entretanto, isso possibilita que alguns ASes possam, ainda que sob uma condição de sequestro, efetivamente rotear pacotes à rede vitimada.

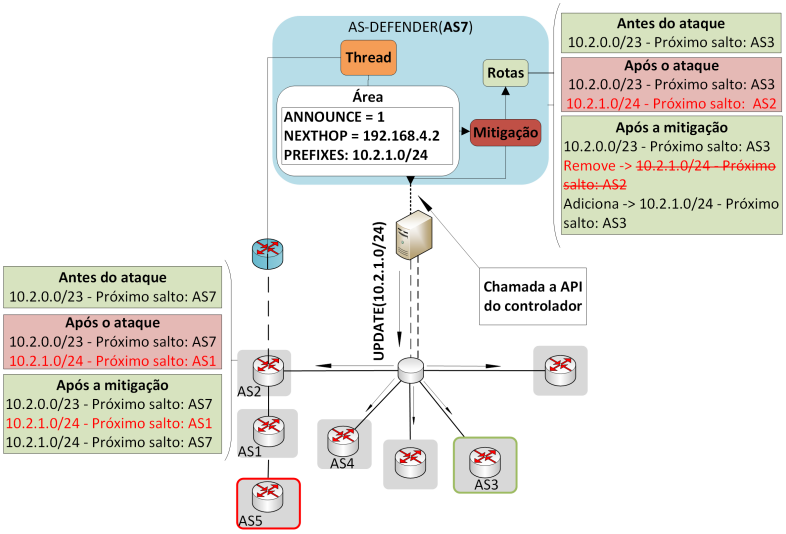

Figura 5. Mitigação com anúncio

4. Validação experimental

Como prova de conceito, foram definidos dois cenários de comunicação BGP onde foram executados tanto ataques de prefixo como de subprefixo, para avaliar possíveis diferenças de comportamento do protótipo. No primeiro cenário, avalia-se o funcionamento da detecção e mitigação em si, enquanto o segundo expôs o *AS-Defender* a diversas situações representativas de cenários reais, tendo seu comportamento avaliado conforme algumas variáveis do ambiente interdomínio eram modificadas. Essas variáveis foram a posição do atacante e a da vítima, a quantidade de vítimas, o tipo de ataque (a prefixo ou a subprefixo), o tipo de defesa (com ou sem anúncio) e a adição de filtros de prefixos como mecanismo extra de defesa. As validações foram feitas a partir do uso do comando *traceroute* em máquinas previamente configuradas na topologia (*hosts* H1, H2, H3 e H4), onde foram executadas tentativas de acesso aos IPs dos ASes vítimas antes, durante e após os ataques, coletando os caminhos referentes a cada um desses momentos.

As Figuras 6 e 7 contêm as topologias de cada cenário experimental, que foram implementadas utilizando a ferramenta *Knet*[4]. O *Knet* permitiu o isolamento de cada instância de sistema autônomo através do uso da tecnologia *Docker*[5] para a criação de contêineres. O *software* utilizado para a execução do processo BGP em cada contêiner foi o *BIRD*[6]. Todo esse ecossistema experimental foi executado em uma VM no serviço de computação em nuvem *Google Cloud Platform*[7]. A topologia do segundo cenário foi inspirada em cenários reais de conectividade BGP obtidos com a ferramenta *BGPlay*[8], onde o *AS-Defender* estaria vinculado a um AS de trânsito que agiria de forma similar a um PTT, porém mantendo seu número de AS no *AS PATH* dos anúncios que dele partem.

No total foram executados 10 experimentos, sendo um referente ao primeiro cenário, para analisar a eficácia da proposta, e outros nove utilizando o segundo cenário,

[4]Disponível em `https://github.com/knetsolutions/KNet`
[5]Disponível em `https://www.docker.com/`
[6]Disponível em `https://bird.network.cz/`
[7]Disponível em `https://cloud.google.com/`
[8]Disponível em `https://stat.ripe.net/widget/bgplay`

com vistas à avaliação de sua eficiência em situações diversas.

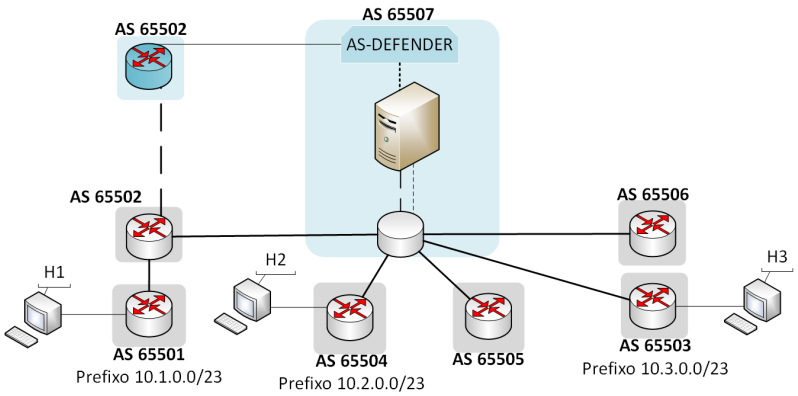

Figura 6. Primeiro cenário experimental

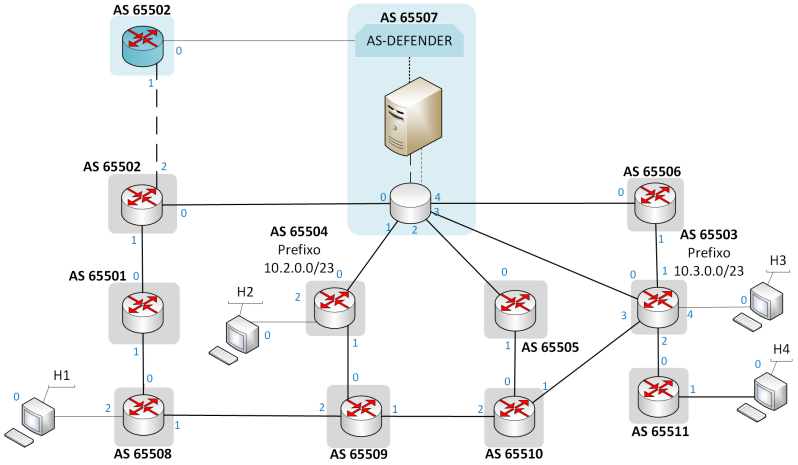

Figura 7. Segundo cenário experimental

No caso do primeiro experimento, executado no primeiro cenário, foi realizado um ataque de prefixo a partir do AS 65501, por meio do anúncio do prefixo 10.3.0.0/23, de propriedade do AS 65503. O mecanismo de mitigação com anúncio foi escolhido e os subprefixos anunciados pelo *AS-Defender* foram o 10.3.0.0/24 e o 10.3.1.0/24. A Tabela 2 mostra que, após a mitigação do ataque, todos os ASes voltaram a ter conectividade com o AS vítima, mostrando a eficácia do mecanismo em um cenário favorável. As colunas da Tabela 2 mostram os resultados do primeiro experimento, onde CAA (Caminho Antes do Ataque), CDA (Caminho Depois do Ataque) e CDM (Caminho Depois da Mitigação) indicam, respectivamente, os caminhos percorridos a partir de cada AS antes do ataque, após o ataque e após a mitigação. As colunas "Poluído" e "Válido" informam, respectivamente, se um AS passou a preferir a rota falsa após o ataque e se faz uso de uma rota verdadeira após a mitigação. Verifica-se que apenas dois ASes tiveram suas tabelas de rota poluídas, o AS 65501, que é o próprio atacante, e o AS 65502. No caso do AS 65501, o fato deste receber subprefixos anunciados pelo *AS-Defender* faz com que o próprio AS atacante prefira o novo caminho até a vítima devido à funcionalidade de *longest prefix match*, não mais efetivando o sequestro desse prefixo. ASes não poluídos

foram contabilizados como válidos de maneira a indicar todos os ASes que, ao final do processo de mitigação, acessavam a vítima adequadamente.

AS	CAA	CDA	CDM	Poluído	Válido
65501	65502 - 65507 - 65503	interno	65502 - 65507 - 65503	Sim	Sim
65502	65507 - 65503	65501	65507 - 65503	Sim	Sim
65503	interno	interno	interno	Não	Sim
65504	65507 - 65503	65507 - 65503	65507 - 65503	Não	Sim
65505	65507 - 65503	65507 - 65503	65507 - 65503	Não	Sim
65506	65507 - 65503	65507 - 65503	65507 - 65503	Não	Sim
65507	65503	65503	65503	Não	Sim

Tabela 2. Resultados do experimento 1

Analogamente ao primeiro cenário, os demais experimentos usando o segundo cenário mostraram que, conforme as variáveis relacionadas ao ataque foram modificadas, a eficiência do mecanismo do *AS-Defender* variou bastante. A Tabela 3 resume as variações e os resultados para esses nove experimentos.

As colunas "Atacante" e "Vítima" mostram as variações em termos de posicionamento de ASes atacantes e vítimas, respectivamente. Nos experimentos 5 e 6, por exemplo, é possível notar que os ataques continham duas vítimas em cada. A coluna "Ataque" informa se o ataque é direcionado ao prefixo pertencente ao AS vítima ou a um subprefixo deste. Verifica-se que apenas o experimento 1 contém um caso de ataque a prefixo, todos os demais são direcionados a subprefixos da vítima. As duas colunas seguintes mostram, respectivamente, os prefixos anunciados pela vítima e os prefixos, ou subprefixos, sequestrados. A coluna "Mitigação" informa o tipo de mitigação usado no experimento e a coluna seguinte informa os subprefixos anunciados na mitigação, caso seu tipo tenha sido com anúncio. As colunas "Poluídos" e "Mitigados" informam, respectivamente, o número de ASes poluídos (ou seja, que tiveram suas tabelas de rota afetadas pelos prefixos sequestrados) e mitigados. A coluna "Válidos" mostra o número de ASes que, ao final, seguem uma rota válida até a vítima para acessar o prefixo mitigado. Vale notar que esse valor é sempre igual ou superior ao número de ASes mitigados, uma vez que contabiliza-se também nessa coluna os ASes que não foram poluídos e que, portanto, já possuiam uma rota válida. Por fim, a última coluna indica quais experimentos usaram filtros de prefixos. Esse mecanismo extra envolve a aplicação de filtros nos processos BGP dos ASes clientes (exceto no AS 65502, que faz a coleta das rotas), atravéś dos quais os anúncios de subprefixos relacionados com os prefixos dos demais ASes clientes só são aceitos se vierem do *AS-Defender*. Esse mecanismo parte do pressuposto que o *AS-Defender* funciona como "elo de ligação" entre os ASes clientes e apenas o experimento 9 fez uso dele. A Tabela 4 traz a lista de todos os ASes poluídos e mitigados em cada experimento do segundo cenário.

4.1. Discussão dos resultados

Analisando os resultados, foi possível perceber que, de uma forma geral, a mitigação sem anúncio foi consideravelmente menos eficiente, demonstrando que simplesmente redirecionar os dados passantes pelo *AS-Defender* não garante a mitigação do ataque. No entanto, a mitigação sem anúncio é menos "intrusiva", evitando que seu efeito se propague pela Internet, tal qual ocorre quando um prefixo é sequestrado. Como alternativa para

Exp	Atacante	Vítima	Ataque	Pr. Vítima	Pr. Sequest.	Mitigação	Pr. Mitig.	Poluídos	Mitigados	Válidos	Filtro
1	AS 65501	AS 65503	prefixo	10.3.0.0/23	10.3.0.0/23	Com anúncio	10.3.0.0/24 10.3.1.0/24	4	4	11	Não
2	AS 65501	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Sem anúncio	NA	9	4	6	Não
3	AS 65508	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Com anúncio	10.3.0.0/24 10.3.1.0/24	9	6	8	Não
4	AS 65508	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Sem anúncio	NA	9	2	4	Não
5a	AS 65509	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Com anúncio	10.3.1.0/24	9	6	8	Não
5b		AS 65504	subprefixo	10.2.0.0/23	10.2.1.0/24	Com anúncio	10.2.1.0/24	10	7	8	Não
6a	AS 65509	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Sem anúncio	NA	9	3	5	Não
6b		AS 65504	subprefixo	10.2.0.0/23	10.2.1.0/24	Sem anúncio	NA	10	1	2	Não
7	AS 65510	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Com anúncio	10.3.0.0/24 10.3.1.0/24	10	7	8	Não
8	AS 65503	AS 65504	subprefixo	10.3.0.0/23	10.3.1.0/24	Com anúncio	10.3.0.0/24 10.3.1.0/24	10	6	7	Não
9	AS 65508	AS 65503	subprefixo	10.3.0.0/23	10.3.1.0/24	Sem anúncio	NA	9	4	6	Sim

Tabela 3. Visão geral dos resultados dos experimentos realizados no segundo cenário

Exp	Atacante	Vítima	ASes poluídos	ASes mitigados
1	AS 65501	AS 65503	65501 / 65502 / 65508 / 65509	Todos
2	AS 65501	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510	65504 / 65505 / 65506 / 65507
3	AS 65508	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510	65502 / 65504 / 65505 / 65506 / 65507 / 65510
4	AS 65508	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510	65506 / 65507
5a	AS 65509	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510	65501 / 65502 / 65504 / 65505 / 65506 / 65507
5b		AS 65504	65501 / 65502 / 65503 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510 / 65511	65501 / 65502 / 65503 / 65505 / 65506 / 65507 / 65511
6a	AS 65509	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510	65502 / 65506 / 65507
6b		AS 65504	65501 / 65502 / 65503 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510 / 65511	65507
7	AS 65510	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510 / 65511	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65511
8	AS 65503	AS 65504	65501 / 65502 / 65503 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510 / 65511	65501 / 65502 / 65505 / 65506 / 65507 / 65508
9	AS 65508	AS 65503	65501 / 65502 / 65504 / 65505 / 65506 / 65507 / 65508 / 65509 / 65510	65504 / 65505 / 65506 / 65507

Tabela 4. Relação de ASes poluídos e mitigados nos experimentos do segundo cenário

melhorar a eficácia da mitigação sem anúncio, o experimento 9 repete o experimento 4, porém com os ASes clientes adotando filtros de prefixo, o que dobrou o número de ASes mitigados. Esse é um subterfúgio possível no mundo real, mas que demanda colaboração entre ASes vizinhos. Para os casos de mitigação com anúncio, verifica-se uma eficácia igual ou superior a 60% se considerarmos todos ASes poluídos e mitigados.

Um aspecto identificado foi que, em alguns casos, ASes que tiveram seu acesso à vítima reestabelecido passaram a fazê-lo por caminhos mais longos (em números de ASes). Trata-se de um efeito já esperado em caso de mitigação com anúncio. No entanto, a depender da topologia de interligação dos ASes e de suas políticas de importação e exportação de rotas, este efeito pode se propagar para ASes não poluídos. É desejável que tal efeito seja minimizado, porém é importante notar que, no experimento do segundo cenário, houve casos de ASes não-clientes poluídos que foram mitigados graças à propagação dos subprefixos anunciados pelo *AS-Defender* (experimentos 3, 5.b, 7 e 8).

5. Conclusão

Este trabalho apresenta o *AS-Defender*, uma solução SDN para auxiliar ASes de trânsito na detecção e mitigação de sequestros de prefixos IP de forma automatizada. Para tal, o *AS-Defender* utiliza um mecanismo de coleta de rotas BGP para detectar sequestros de prefixos, além de se apoiar nos benefícios inerentes ao paradigma SDN para programar, de forma rápida e flexível, o comportamento da rede e, assim, automaticamente mitigar o ataque detectado. De um modo geral, o *AS-Defender* roda no plano de controle de um AS de trânsito dotado de recursos de SDN para que, através dele, os ASes vizinhos (ou clientes) desse AS de trânsito passem a ser defendidos contra o sequestro de seus prefixos.

Os resultados que validam a proposta foram obtidos a partir de uma série de experimentos de ataques de sequestro de prefixo sobre topologias de rede emulando cenários realistas de interligação entre diferentes ASes, e tendo o *AS-Defender* rodando no AS de trânsito que os interliga. Foi possível verificar a eficácia do *AS-Defender* e como o alcance dessa solução pode variar em função de diferentes cenários possíveis de ataque.

Como trabalhos futuros, pretende-se incorporar à solução recursos de desagregação automática de prefixos e realizar um estudo sobre formas para minimizar o efeito da mitigação com anúncio pela Internet.

Referências

Balu, K., Pardal, M. L., and Correia, M. (2016). Darshana: Detecting route hijacking for communication confidentiality. In *Network Computing and Applications (NCA), 2016 IEEE 15th International Symposium on*, pages 52–59. IEEE.

Chi, Y.-J., Oliveira, R., and Zhang, L. (2008). Cyclops: the as-level connectivity observatory. *ACM SIGCOMM Computer Communication Review*, 38(5):5–16.

Horn, C. (2009). Understanding ip prefix hijacking and its detection. In *Seminar internet routing, intelligent networks (INET)*.

Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., and Zhang, L. (2006). Phas: A prefix hijack alert system. In *USENIX Security symposium*, volume 1, page 3.

ONF (2012). Software-defined networking: The new norm for networks. *ONF White Paper*, 2:2–6.

RIPE NCC (2018). RIPE Network Coordination Center. https://www.ripe.net/. [Online, accessed on 29-April-2018].

University of Oregon (2018). Route Views Project. http://www.routeviews.org/routeviews/. [Online, accessed on 29-April-2018].

Sermpezis, P., Chaviaras, G., Gigis, P., and Dimitropoulos, X. (2016). Monitor, detect, mitigate: Combating bgp prefix hijacking in real-time with artemis. *arXiv preprint arXiv:1609.05702*.

Sermpezis, P., Kotronis, V., Gigis, P., Dimitropoulos, X., Cicalese, D., King, A., and Dainotti, A. (2018). Artemis: Neutralizing bgp hijacking within a minute. *arXiv preprint arXiv:1801.01085*.

Shi, X., Xiang, Y., Wang, Z., Yin, X., and Wu, J. (2012). Detecting prefix hijackings in the internet with argus. In *Proceedings of the 2012 Internet Measurement Conference*, pages 15–28. ACM.

Traina, P. (1995). Bgp-4 protocol analysis. IETF RFC 1774.

Zhang, Z., Zhang, Y., Hu, Y. C., Mao, Z. M., and Bush, R. (2008). Ispy: detecting ip prefix hijacking on my own. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 327–338. ACM.