Mitigando a Ameaça dos Ataques Slow DDoS a Redes SDN usando Consolidação de Regras

Francisco de A. C. de Albuquerque[1] **Jr., Iguatemi E. Fonseca**[1]

Universidade Federal da Paraíba (UFPB)
Programa de Pós-Graduação em Informática (PPGI)

facajx@gmail.com, iguatemi@ci.ufpb.br

Abstract. *This paper presents an approach to mitigate denial of service attacks that exploit the architecture of SDN networks, focusing on low-flow attacks on TCAM memory. The TCAM attack can result in the unavailability of SDN switches, since their TCAM memory is completely occupied by malicious rules coming from a botnet. To face this type of attack, we propose the use of rule consolidation in SDN networks. This approach consists of reducing TCAM memory usage by consolidating similar rules, which avoids excessive resource consumption and minimizes the impact caused by the Slow-TCAM attack.*

Resumo. *Este artigo apresenta uma abordagem para mitigar ataques de negação de serviço que exploram a arquitetura das redes SDN, com foco em ataques de baixo fluxo a memória TCAM. O ataque a TCAM pode resultar na indisponibilidade de switches SDN, uma vez que sua memória TCAM é preenchida completamente por regras maliciosas provenientes de uma botnet. Para enfrentar esse tipo de ataque, propomos a utilização da consolidação de regras em redes SDN. Essa abordagem consiste em reduzir o uso da memória TCAM por meio da consolidação de regras semelhantes, o que evita o consumo excessivo de recursos e minimiza o impacto causado pelo ataque Slow-TCAM.*

1. Introdução

Nos últimos anos, os ataques de negação de serviço distribuídos (DDoS - *Distributed Denial of Service*) têm se tornado uma ameaça crescente para organizações e usuários em todo o mundo. Entre as várias formas de ataques DDoS, um tipo particularmente insidioso é o de baixo tráfego de dados, conhecido como Slow DDoS, que visa sobrecarregar recursos de rede e servidores ao longo de um período prolongado, dificultando sua detecção e mitigação. Esse tipo de ataque se caracteriza por enviar tráfego malicioso em baixa intensidade, visando esgotar recursos lentamente, resultando em interrupções no serviço e perda de disponibilidade.

Os ataques DDoS de baixa taxa de tráfego, são difíceis de mitigar com as ferramentas de defesa existentes. Adicionalmente, a maioria dos estudos se concentra na solução de ataques DDoS de alta taxa. Por enquanto, as recentes ameaças DDoS de taxa lenta são difíceis de detectar e mitigar [Yungaicela-Naula et al. 2022]. Ataques de camada de aplicação de baixo volume são caracterizados pela pequena quantidade de tráfego necessária para derrotar uma vítima. Essas ameaças podem ser classificadas em três categorias: i) Ataques de baixa taxa que enviam tráfego em pulsos periódicos de curto prazo, ii) ataques de taxa lenta que exploram parâmetros de tempo no lado do servidor

enviando ou recebendo tráfego mais lento do que o esperado e iii) ataques one-shot que danificam as vítimas em um único pedido [Punitha et al. 2020].

Redes Definidas por Software (SDN - *Software Defined Networking*) recentemente atraíram muita atenção, pois se mostram promissoras como um conceito de gerenciamento de rede que pode oferecer defesa DDoS baseada em rede eficaz contra várias formas de ataque DDoS. Na arquitetura SDN, o controlador SDN centralizado pode aproveitar o conhecimento de sua própria rede para detectar ataques DDoS por meio de técnicas como análise de padrão de tráfego ou aprendizado de máquina. Uma vez que os ataques DDoS são detectados, o controlador SDN pode usar estratégias de mitigação, como bloquear o fluxo do invasor ou redirecionar o tráfego legítimo para um sistema seguro, implantando uma política de segurança atualizada nos switches de rede [Hong et al. 2018]. Para maior eficiência, os switches SDN usam um tipo de memória TCAM (*Ternary Content-Addressable*) de alto desempenho para instalar regras. No entanto, devido ao alto custo e consumo de energia do TCAM, os switches possuem uma quantidade limitada de memória TCAM. Consequentemente, um número limitado de regras pode ser instalado. Essa limitação foi explorada para realizar ataques DDoS, como ataques de saturação, que geram grandes quantidades de tráfego e com ataques de baixo tráfego, o que torna o ataque difícil de ser detectado [Pascoal et al. 2020], [Pascoal et al. 2017], [Jiahao Cao 2022].

Um dos recentes ataques direcionados às redes SDN é o ataque Slow-TCAM. Esse ataque tem como objetivo explorar a memória TCAM dos switches SDN, sobrecarregando-a com um grande número de regras maliciosas. Quando a memória TCAM é completamente preenchida, o switch se torna incapaz de processar novas regras, resultando em indisponibilidade da rede e interrupção dos serviços [Pascoal et al. 2017], [Pascoal et al. 2020]. Recentemente, uma variação do ataque SlowTCAM foi publicada na literatura [Jiahao Cao 2022]. Neste contexto, o presente trabalho tem como objetivo demonstrar uma abordagem eficaz para mitigar o ataque Slow-TCAM por meio da consolidação de regras em redes SDN. A consolidação de regras é uma técnica que visa reduzir o uso da memória TCAM, eliminando regras redundantes ou sobrepostas e combinando várias regras em uma única. Ao reduzir a quantidade de regras armazenadas na memória TCAM, é possível minimizar o impacto do ataque Slow-TCAM e garantir a disponibilidade contínua da rede.

Para realizar a demonstração, foi utilizado um ambiente de teste em laboratório, no qual um switch SDN foi submetido a um ataque Slow-TCAM, com regras maliciosas enviadas por uma botnet. Em seguida, a técnica de consolidação de regras foi aplicada para reduzir o uso da memória TCAM do switch afetado. Foram realizadas diversas análises e medições para avaliar a eficácia da abordagem proposta em mitigar o impacto do ataque Slow-TCAM. Este trabalho contribui para o avanço da segurança nas redes SDN, oferecendo uma solução prática e eficiente para mitigar um dos ataques mais prejudiciais direcionados a essa arquitetura. A consolidação de regras proposta pode ser aplicada em switches SDN existentes, sem a necessidade de modificações significativas na infraestrutura de rede, o que a torna uma abordagem viável para a proteção contra ataques Slow-TCAM.

O restante deste artigo está organizado da seguinte maneira. Na Seção 2, a memória TCAM é brevemente descrita. Na Seção 3, a técnica de consolidação utilizada neste artigo é apresentada. Os resultados experimentais, impactos no desempenho

da rede e na mitigação do ataque Slow-TCAM são debatidos na Seção 4. As conclções e comentários finais são mostrados na Seção 5.

2. A Memória TCAM e o seu Papel em Redes SDN

Em redes tradicionais, o controle e o encaminhamento dos pacotes são realizados por dispositivos de rede individuais, como roteadores e switches. O plano de controle é distribuído e cada dispositivo toma decisões de roteamento independentemente. As configurações e políticas de rede são implementadas localmente em cada dispositivo. Nas redes SDN, o controle é centralizado em um controlador de rede, que toma decisões globais de roteamento e envia instruções para os dispositivos de rede. Os dispositivos de rede atuam como encaminhadores de pacotes simples, seguindo as instruções do controlador. O plano de controle é separado do plano de dados, permitindo maior flexibilidade e programabilidade. A Fig. 1 ilustra a comparação entre redes tradicionais e redes SDN.

A memória TCAM é um componente importante em redes SDN. Ela desempenha um papel fundamental no encaminhamento de pacotes em tempo real. Em uma rede SDN, a TCAM é usada principalmente em dispositivos de encaminhamento de pacotes, como switches e roteadores. Esses dispositivos geralmente contêm tabelas de fluxos (*flow tables*) que são usadas para tomar decisões de encaminhamento com base nas informações contidas nos pacotes, como endereços IP de origem e destino. A memória TCAM oferece correspondência rápida de pacotes, suporte a fluxos complexos, eficiência de espaço, flexibilidade e programabilidade. Ela é essencial para lidar com políticas de roteamento e filtragem de pacotes sofisticadas, além de permitir implementações avançadas de encaminhamento.

A memória TCAM é o hardware mais amplamente utilizado para pesquisar regras de classificação de pacotes devido ao seu desempenho. Cada célula TCAM pode armazenar três estados, 0, 1 ou "não importa" [Lin and Wang 2023]. Essa flexibilidade é útil para implementar recursos como listas de controle de acesso (ACL - *Access Control Lists*) ou políticas de roteamento complexas. No entanto, a TCAM também tem algumas limitações. Ela é relativamente cara em termos de custo e consumo de energia, o que pode restringir seu uso em dispositivos de baixo custo ou com restrições de energia. Além disso, a capacidade de armazenamento da TCAM pode ser limitada, o que pode impor limites à escala das redes SDN.

3. A Consolidação de Regras

Uma técnica promissora na redução das demandas de TCAMs é usar a consolidação de regras na tabela de fluxo do switch SDN, essa técnica tem como foco mesclar várias entradas de fluxo em uma sem modificar a semântica de encaminhamento. A consolidação de regras é uma solução a nível de software e não requer nenhuma alteração no protocolo OpenFlow, nem nos equipamentos de switch OpenFlow. É possível implementar a consolidação diretamente nos controladores [Luo et al. 2015].

Para melhorar a eficiência da memória TCAM de forma a atender o maior número de requisições, a consolidação de regras serve como um bom método para sobrepor um conjunto de regras de fluxo em uma regra comum, de modo que esses fluxos possam ser tratados simultaneamente com o mínimo de regras possíveis. Diversos mecanismos

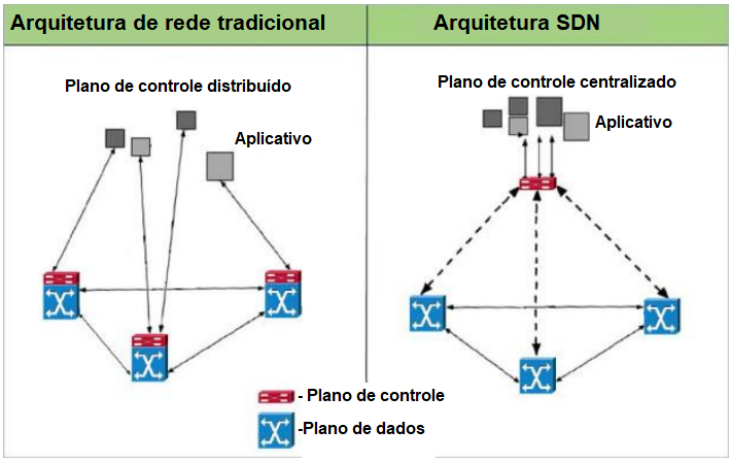

Figura 1. Rede tradicional e rede SDN [Altangerel et al. 2019].

de consolidação de regras foram introduzidos para reduzir os custos de comunicação e memória em redes SDN.

z	*m*	*a*
1	0111	Fwd 1
2	1111	Fwd 1
3	*101	Fwd 2
4	*011	Fwd 1
5	1*0*	Fwd 3
6	1*1*	Fwd 3
7	****	Drop

(a) a tabela original

z	*m*	*a*
1	*101	Fwd 2
2	**11	Fwd 1
3	1***	Fwd 3
4	****	Drop

(b) a tabela consolidada

Figura 2. Exemplo da redução de regras causada por consolidação. Fonte: [Luo et al. 2015].

A exemplo, o DEVOFLOW [Curtis et al. 2011] que permite a consolidação aproveitando as regras curinga para mitigar as interações entre os switches e o controlador, bem como para reduzir o número de entradas de fluxo armazenadas no TCAM. A dificuldade essencial aqui é que para todos os fluxos com a mesma regra curinga, o controlador não pode obter estatísticas de fluxo individuais para aplicar um tratamento específico a grandes fluxos. Além disso, o DEVOFLOW requer modificações significativas nos dispositivos de hardware e no protocolo OpenFlow [Minh et al. 2019].

3.1. Problemas da Consolidação

Embora o mecanismo de consolidação de entradas de fluxo possa otimizar eficientemente o tamanho das tabelas de fluxo para aceitar mais entradas de fluxo, o processo de compressão-descompressão requer altos recursos computacionais que adicionam mais sobrecarga no plano de dados, especialmente na condição de uma grande quantidade de tráfego agregado [Alsaeedi et al. 2019]. A forma como é implementada a consolidação da tabela de fluxo em SDN é crítica para a eficiência. Isso ocorre porque as regras de encaminhamento são voláteis e o algoritmo de consolidação da tabela de fluxo retardará as atualizações da tabela e aumentará a duração da atualização. No contexto das redes SDN,

o plano de dados é responsável por encaminhar o tráfego de rede e o plano de controle é responsável por gerenciar e programar o encaminhamento do tráfego. Durante o processo de atualização de regras no plano de controle, pode ocorrer uma inconsistência temporária entre os planos de dados e controle. Isso significa que as regras de encaminhamento de tráfego ainda não foram completamente atualizadas em todos os dispositivos de rede, e, portanto, os dispositivos de rede podem encaminhar o tráfego de acordo com as regras antigas ou desatualizadas.

Essa inconsistência pode levar a erros de encaminhamento, como falhas de acessibilidade, loops de encaminhamento, isolamento de tráfego e vazamento, que podem afetar o desempenho e a segurança da rede. Portanto, é importante minimizar o tempo de inconsistência e implementar procedimentos de segurança para mitigar quaisquer riscos durante esse período.

A consolidação da tabela de fluxo é um problema difícil porque as regras na tabela não são prefixadas. O trabalho de [Applegate et al. 2007] provou que encontrar a expressão mínima para uma tabela sem prefixo é NP-hard, mesmo se houver apenas dois tipos de ação. Segundo [Luo et al. 2015], a melhor prática, inclusive seguida por várias referências, é a ideia de tecelagem de bits. A tecelagem de bits é baseada em uma observação importante de que, ao silenciar algumas das posições de bits, um grupo de regras não prefixadas poderia ser transformado em formato de prefixo simultaneamente.

O processo de consolidação não deve alterar a parte da ação de nenhuma entrada de fluxo. Como resultado, motivado pela consolidação de TCAMs, alguns autores propõem um esquema de consolidação *offline* FFTA (*Fast Flow Table Aggregation*) sem prefixo que pode agregar 100 partições de regras em apenas alguns milissegundos. Ele corta os campos correspondentes sem prefixo para prefixar partições permutáveis e, em seguida, agrega cada partição respectivamente. O FFTA constrói uma árvore de pesquisa binária (BST - *Binary Search Tree*) de partições de prefixo e, em seguida, aplica o algoritmo ORTC (*Optimal Routing Table Constructor*) para omitir as permutações e simplificar o processo de consolidação. No esquema de consolidação de entradas de fluxo, é possível que um novo fluxo desconhecido corresponda a uma entrada de fluxo agregado resultando em possíveis erros de roteamento.

3.2. Algoritmo utilizado avaliação da consolidação de regras

O problema de consolidação de regras em memória TCAM surge quando há um conjunto de regras de fluxo, cada uma definindo uma correspondência de padrão específica e uma ação a ser tomada. O objetivo é encontrar uma maneira eficiente de armazenar essas regras na TCAM, minimizando o número de entradas ocupadas e evitando que a memória fique cheia sem quebrar a lógica da rede.

A consolidação de regras é uma técnica utilizada em redes de computadores para agregar múltiplos endereços IP em um único endereço, com o objetivo de reduzir a complexidade e melhorar a eficiência da rede. Em redes SDN, essa técnica pode ser aplicada de forma dinâmica, utilizando o controlador de rede para gerenciar o processo de agregação. Para permitir que essa técnica seja aplicada de forma automatizada, foi criado um algoritmo específico para ser utilizado no controlador Ryu, o qual é mostrado na Figura 3 . Este algoritmo é responsável em realizar o processo de consolidação de IPs e é acionado sempre que a quantidade de regras presentes no controlador alcança um

limiar. Existem distintos modelos de consolidação de regras, este mais simplificado foi desenvolvido para este laboratório, no intuito de testar a eficiência da consolidação em um ambiente vulnerável.

Neste caso, é aplicado um método de detecção baseado em limiar que geralmente monitora os comportamentos da rede em tempo real. Uma vez que um determinado indicador excede o limite predefinido, sugere-se que a rede está passando por um momento de crescimento de tráfego de clientes legítimos ou que o ataque ocorreu e nesse momento pode-se aplicar uma técnica de consolidação de regras. Visto que o Slow-TCAM não consome CPU e que qualquer técnica de consolidação, mesmo que consuma recursos de processamento, só precisa ser ativada após a memória chegar a um limiar e, pela característica do ataque Slow-TCAM ser de baixo tráfego, permite que a técnica de consolidação possua tempo suficiente para agir, reduzindo o consumo de memória e assim mitigando o ataque.

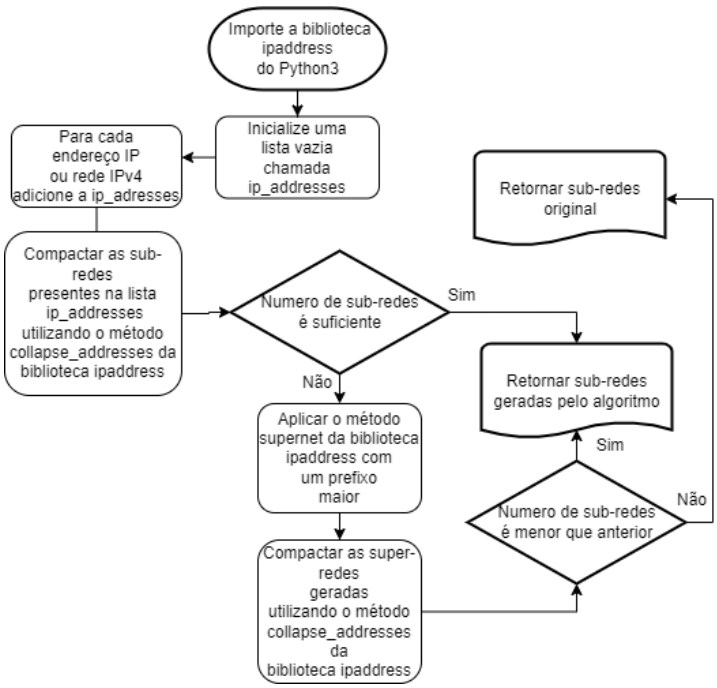

Figura 3. Algoritmo para consolidação de regras em SDN.

A complexidade do algoritmo demonstrado na Figura 3 depende do número de endereços IP fornecidos como entrada, bem como do valor dos parâmetros para o número de sub-redes geradas. No entanto, podemos fazer algumas observações. A primeira parte do algoritmo, que cria objetos IP a partir dos endereços fornecidos, tem complexidade linear $O(n)$, em que n é o número de endereços IP.

Em seguida, o algoritmo chama a função "ipaddress.collapse_addresses()", que tem complexidade $O(nlogn)$. Depois, o algoritmo itera sobre as subredes resultantes e cria subredes "mais compactas" (com prefixos menores) usando o método "subnet.supernet()", que tem complexidade $O(1)$. No entanto, a quantidade de iterações pode ser no máximo igual ao número de subredes resultantes da chamada anterior a "ipaddress.collapse_addresses()".

Por fim, o algoritmo chama novamente "ipaddress.collapse_addresses()" na lista de subredes resultantes do passo anterior, o que novamente tem complexidade $O(nlogn)$.

Portanto, a complexidade geral do algoritmo é de $O(nlogn)$.

3.3. Consolidação da Tabela de Regras como Mitigação ao Slow-TCAM

A ocorrência de ataques de estouro de tabela de fluxo em redes SDN é possível, pois tais ataques resultam da necessidade de armazenar as regras de fluxo em um ambiente físico finito. Mas, pela característica de ataques como o Slow-TCAM serem considerados de baixo tráfego e, neste caso, poder ser aplicado por meio de uma botnet simulando interações válidas, sua detecção é difícil para técnicas comuns de monitoramento de rede. Gerenciar a quantidade de regras inseridas no switch de forma a minimizar o uso de memória passa a ser uma forma de mitigar o ataque.

Um invasor é capaz de fazer inferências sobre a tabela de switch estar perto de ser completamente preenchida ou cheia de fato por meio de de diversas técnicas. A técnica mais simples é por meio do endereço MAC do switch. Os três primeiros bytes de um endereço MAC são destinados a identificação do fabricante, o que permite a pesquisa na documentação do switch após a descoberta do fabricante. Tal informação pode ser somada a análise do comportamento do switch por meio de um ataque preliminar mais simples, como o MAC *Flooding* ou CAM *Table Overflow*, na intenção de analisar o comportamento do switch. Dada a situação em que o invasor inferiu sobre tais informações, ele é capaz de lançar um ataque de inundação da tabela de fluxo forçando o switch a constantemente solicitar novas regras, tal comportamento pode ter efeitos negativos ao switch e ao controlador [Yoon et al. 2017].

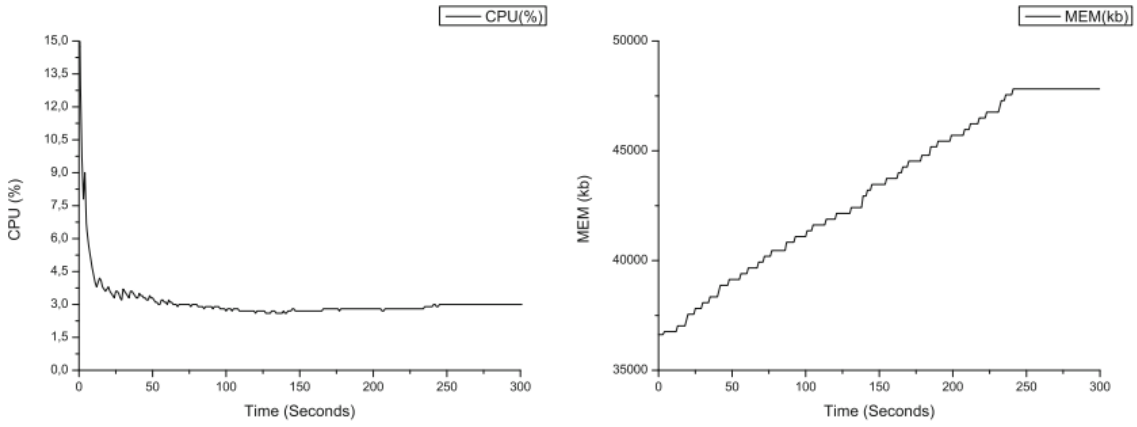

Figura 4. Uso de memória durante um ataque Slow-TCAM com intensidade de 5,8 pacotes únicos por segundo. Fonte: [Pascoal et al. 2020].

Conforme [Pascoal et al. 2020], a Fig. 4 ilustra o uso da memória TCAM pelo ataque Slow-TCAM com intensidade de $5,8\ \ regras/segundo$. Esse ataque pela sua característica de gerar baixo tráfego de rede, demora um pouco mais de 4 minutos para ocupar toda a capacidade de regras instalando 1500 regras. Os demais cenários de simulação do ataque com diferentes intensidades de ataque tiveram o mesmo comportamento. Para o ataque mais lento com intensidade de $3,2\ \ pacotes/segundo$, o invasor pode negar o serviço ainda mais silenciosamente em cerca de 8 minutos, praticamente sem impacto no uso da CPU do controlador.

A Fig. 5 mostra o número de entradas de fluxo monitoradas pelo switch durante simulação realizada por [Minh et al. 2019], em que se compara um switch controlado pelo

controlador de consolidação de regras e um controlador normal. É claro ver a eficácia do controlador de consolidação de regras, pois seu switch correspondente lida com menos de 100 entradas de fluxo, enquanto sua contraparte gerenciada por um controlador normal deve lidar com um número maior de entradas de fluxo (em média, 1.500 a 2.000 entradas).

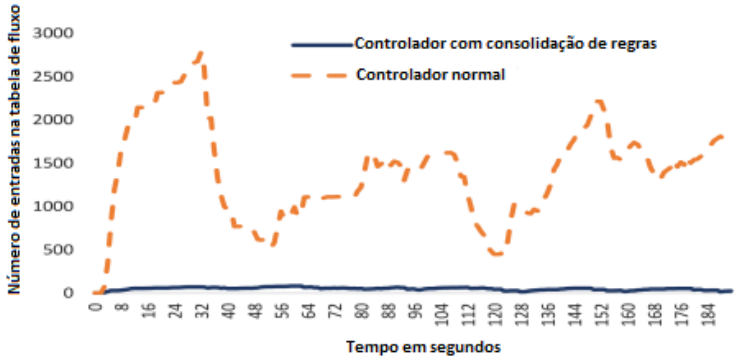

Figura 5. Eficácia do controlador de consolidação proposto em termos de redução do número de entradas de fluxo. Fonte: [Minh et al. 2019].

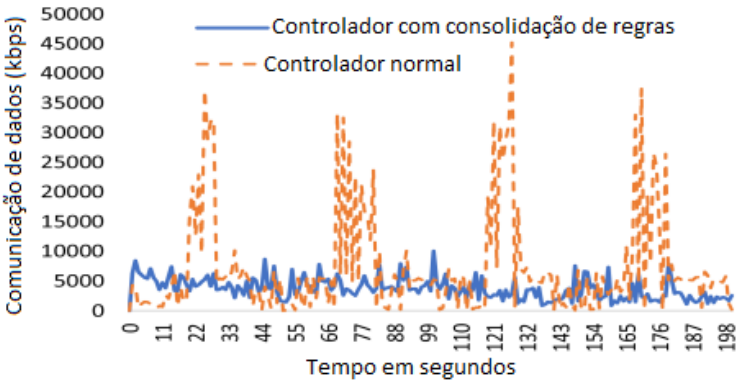

Figura 6. Redução de custos de comunicação entre o switch e o controlador. Fonte: [Minh et al. 2019].

A Fig. 6 apresenta a quantidade de dados que chega ao controlador. Conforme mostrado, ao instalar menos regras de fluxo, o controlador de consolidação pode mitigar significativamente os custos de comunicação (kbps) em comparação com o controlador normal. Além disso, as comunicações no controlador normal flutuam dramaticamente e parecem ter periodicidade de tempo, pois após um período de tempo, os hosts, durante a simulação alteram seus endereços IP e geram um grande número de novos fluxos. Em alguns pontos, os custos de comunicação no controlador de consolidação são iguais ou até superiores ao do controlador normal. Esse fenômeno, que é comum, ocorre quando os hosts encerram o tráfego atual, alteram seus endereços IP para nova geração de tráfego e quando o controlador de consolidação solicita estatísticas de rede [Minh et al. 2019].

É visível nos gráficos acima como a consolidação de regras é capaz de reduzir drasticamente o uso da TCAM, estabilizando o seu consumo e assim se tornando uma alternativa contra o ataque Slow-TCAM.

4. Resultados ao Aplicar a Consolidação de Regras

4.1. Métricas Utilizadas

Neste trabalho foram utilizadas duas métricas para a avaliação de desempenho do algoritmos de mitigação e ataques executados, a saber: o *Time to Service* (TTS) e a disponibilidade. A ferramenta JMeter foi utilizada para a geração e medição de tráfego na rede [JMeter 2023].

O TTS é uma métrica que mede o tempo que leva para que um serviço seja fornecido aos usuários após uma solicitação ser feita. No contexto de um controlador SDN que está sofrendo um ataque de negação de serviço distribuído (DDoS), o TTS pode ser uma boa métrica para avaliar a capacidade do controlador de lidar com o ataque. Segundo Tulio [Pascoal et al. 2020] uma rede SDN possui um TTS maior devido ao tempo de instalação que a abordagem precisa para instalar as regras (envio do PACKET_IN por parte do switch, recepção pelo controlador, envio do FLOW_MOD por parte do controlador, recepção e instalação da regra por parte do switch e somente a partir daí proceder com os encaminhamentos dos pacores).

Quando um controlador SDN é alvo de um ataque DDoS, o tráfego malicioso pode sobrecarregar o sistema e causar atrasos na resposta às solicitações de serviço. Isso pode levar a um aumento no TTS, o que significa que os usuários terão que esperar mais tempo para obter os serviços solicitados. Ao medir o TTS durante um ataque DDoS, é possível determinar se o controlador está lidando de forma eficaz com a carga de tráfego e mantendo os tempos de resposta dentro de um nível aceitável. Se o TTS aumentar drasticamente durante um ataque, isso pode indicar que o controlador está sobrecarregado e não é capaz de fornecer serviços de forma eficiente.

Portanto, o TTS pode ser uma métrica útil para avaliar a capacidade de um controlador SDN de lidar com um ataque DDoS e garantir que os usuários recebam serviços de forma rápida e eficiente, mesmo em condições adversas.

A disponibilidade na ferramenta JMeter é calculada usando uma fórmula simples:

$Disponibilidade = [(Totaldeamostras - Amostrascomfalha)/Totaldeamostras] * 100\%$.

Ou seja, a disponibilidade é o percentual de amostras bem-sucedidas em relação ao total de amostras executadas. Para calcular a disponibilidade, o JMeter conta o número de amostras bem-sucedidas e o número de amostras com falha durante a execução dos testes de carga. Por exemplo, se tiver executado 100 amostras e 5 delas falharam, a disponibilidade seria calculada da seguinte maneira:

$Disponibilidade = [(100 - 5)/100] * 100\% = 95\%$.

Isso significa que a disponibilidade foi de 95% durante a execução dos testes de carga. Para o laboratório foram utilizadas 3000 amostras em um tempo de 300 segundos com timeout máximo de 10 segundos. Ao considerarmos que a disponibilidade foi de 38,23%, atráves do cálculo acima temos: $0,3823 = (3000 - Amostrascomfalha)/3000 * 100\%$, o que significa que das 3000 amostras executadas, 1853 delas falharam, enquanto 1147 foram bem-sucedidas dentro do tempo limite de 10 segundos, durante os 300 segundos de teste.

4.2. Experimentos com um Controlador

No primeiro ambiente de teste, foram utilizadas quatro máquinas virtuais. Uma máquina virtual foi configurada como controlador, outra como um banco de dados MySQL para registrar as requisições, uma terceira como um ambiente Mininet com o switch S1 e três hosts representando um servidor HTTP, um cliente e um atacante. Uma quarta máquina virtual foi configurada com o switch S2, tendo um host como servidor HTTP e outro como cliente, conforme mostrado na Figura 7. Durante os testes, monitoramos o uso da CPU do controlador e o tempo de resposta do servidor HTTP em redes sem ataques e com a execução do ataque. A máquina virtual de aplicação foi responsável por simular uma aplicação simples de registro de acesso, sem interferir nas decisões tomadas pelo controlador.

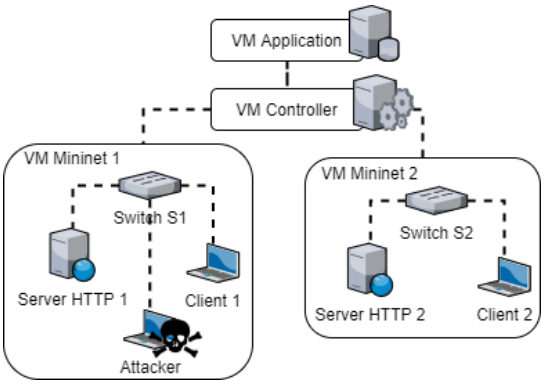

Figura 7. Experimento com um controlador.

Nesta série de experimentos, foi utilizado o controlador Ryu. Inicialmente, realizamos o ataque Slow-TCAM em um cenário com o controlador Ryu, que não implementa a consolidação de regras. Após a comunicação entre os clientes, conforme mostrado na Tabela 1, a máquina virtual Mininet VM 1 sofreu com o preenchimento completo de sua tabela TCAM, resultando em indisponibilidade total. Além disso, o ataque Slow-TCAM executado entre o atacante e o servidor HTTP 1 foi suficiente para encher a fila de requisições do controlador, impedindo-o de atender às requisições do Switch S2 e causando uma indisponibilidade de 38,2% na máquina virtual Mininet 2. Dos 3.000 usuários, apenas 1.147 foram atendidos. Finalmente, a Tabela 1 demonstra que a rede SDN apresenta 100% de disponibilidade quando a consolidação de regras está aplicada.

A Figura 8 ilustra visualmente o processo de consolidação de regras e como ele evita que a memória da TCAM atinja sua capacidade máxima, resultando no TableFull. Na linha superior, em vermelho, que representa o uso da TCAM sem a consolidação de regras, a memória se enche rapidamente à medida que as regras de fluxo são adicionadas. Conforme mais regras são inseridas, o espaço disponível diminui até que a capacidade máxima seja alcançada, resultando no TableFull. No entanto, na linha intermediária, em amarelo, que representa o uso da TCAM com a consolidação de regras, é possível notar uma ocupação mais eficiente da memória, chegando próximo ao gráfico da rede sem ataque.

Inicialmente, o uso de memória é baixo em ambos os casos. No entanto, quando ocorre o ataque Slow-Saturation, a utilização de memória aumenta rapidamente e atinge a capacidade máxima do switch, resultando em um DoS. Quando a consolidação está ativada, o algoritmo é programado para iniciar a consolidação quando o switch atinge 50%

da capacidade. Isso é representado pela linha amarela no gráfico da Figura 8. Como resultado, o uso de memória é reduzido significativamente e o DoS é evitado, inclusive após 200 segundos de simulação a quantidade de memória usada no switch sofrendo ataque, mas com consolidação de regras, se compara com o switch sem ataque. Ao longo do tempo, o uso de memória com a consolidação ativa é mais estável e controlado em comparação com a ausência de consolidação. O gráfico mostra claramente que a consolidação de regras é eficaz para reduzir o uso de memória e evitar a exaustão da memória TCAM durante um ataque de Slow-Saturation.

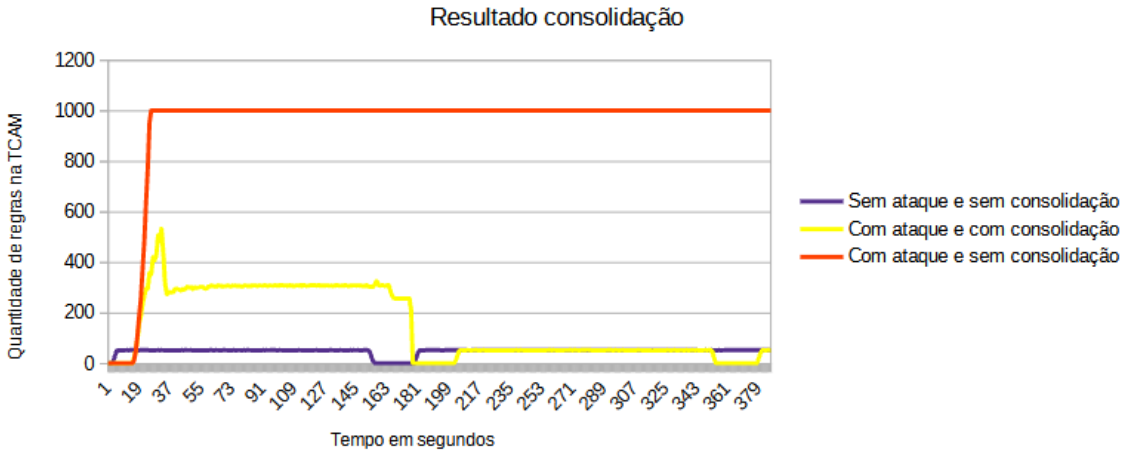

Figura 8. Resultado do algorítimo de consolidação para o experimento com um controlador e um switch.

Tabela 1. Experimentos com Ryu para topologia de um switch com consolidação.

Com ataque e sem consolidação			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	0%	9071 ms	10078 ms
VM Mininet 2	38,23%	7547 ms	18665 ms
	Sem ataque		
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	100%	15 ms	135 ms
VM Mininet 2	100%	36 ms	140 ms
Com ataque e com consolidação			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	100%	55 ms	3180 ms
VM Mininet 2	100%	54 ms	328 ms

4.3. Experimentos com dois controladores

No experimento com dois controladores, conforme mostrado na Figura 9, analisamos se a topologia era capaz de fornecer disponibilidade suficiente para conter o ataque Slow-Saturation durante um teste com duração total de 5 minutos. Realizamos os seguintes passos, sem realizar o ataque, apenas para comprovar que era possível alcançar disponibilidade mesmo ao derrubar um dos controladores:

1. durante o primeiro minuto os dois controladores estavam online;

2. no segundo minuto o Controlador 1 foi parado;
3. no terceiro minuto o Controlador 1 foi iniciado, estando novamente os dois controladores online;
4. no quarto minuto o Controlador 2 foi parado; e
5. no quinto minuto os dois controladores ficaram online.

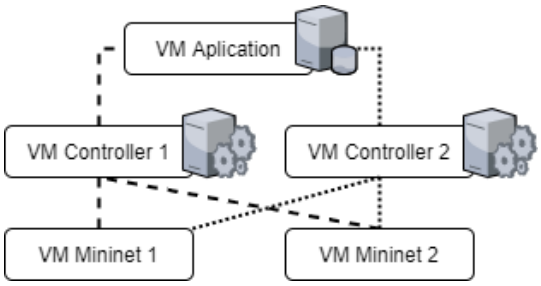

Figura 9. Experimento com dois controladores.

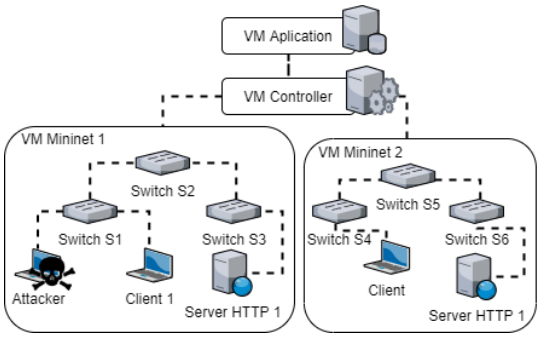

Figura 10. Experimento com topologia em árvore.

Tabela 2. Experimentos com Ryu para topologia de dois controladores.

Com ataque e sem consolidação			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	0%	8801 ms	1254 ms
VM Mininet 2	57,50%	5992 ms	12014 ms
Sem ataque			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	100%	59 ms	1140 ms
VM Mininet 2	100%	55 ms	1090 ms
Com ataque e com consolidação			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	100%	45 ms	1202 ms
VM Mininet 2	100%	32 ms	890 ms

A Tabela 2 sugere que o ataque Slow-TCAM só é efetivo quando a consolidação de regras não é implementada. No entanto, com a consolidação de regras, o uso da memória TCAM é controlado, garantindo que as requisições não sobrecarreguem o switch. É importante ressaltar que essa topologia proporciona disponibilidade mesmo em casos de perda total de comunicação entre o switch e o controlador, como observado no experimento em que um dos controladores foi desligado. No entanto, ter a fila de processamento do controlador saturada não resulta em perda de comunicação. Com o ataque Slow-TCAM, a fila de processamento de ambos os controladores é comprometida, uma vez que o *Packet-In* enviado pelo switch é sempre encaminhado para ambos os controladores pois o ataque irá consumir toda a memória do switch.

4.4. Experimento com Topologia em Árvore

No experimento, foi realizado um teste com múltiplos switches, conforme ilustrado na Figura 10. O laboratório foi configurado de forma simples, em que cada switch envia uma mensagem *Packet-In* para o controlador. No teste com 3 switches em cada máquina

virtual, observamos que o resultado se agravou. Conforme visto na Tabela 3, o ataque realizado na máquina virtual que continha os switches S1, S2 e S3 foi suficiente para derrubar a conexão com os switches S4, S5 e S6 na máquina Virtual 2, de maneira mais rápida do que nos experimentos anteriores.

Esse agravamento ocorreu porque o controlador não estava programado para verificar a rota do pacote e implementar a regra nos três switches de forma proativa. Consequentemente, quando um pacote chegava em qualquer um dos switches, cada switch enviava uma mensagem Packet-In ao controlador, multiplicando a intensidade do ataque pela quantidade de switches. Tornar o controlador mais inteligente, por exemplo, implementando a verificação de rotas de pacotes, reduz o tempo de resposta do ataque. No entanto, não garante proteção contra o Slow-TCAM, que já se mostrou eficaz mesmo com apenas um switch, conforme demonstrado anteriormente.

Por fim, note na Tabela 3 que a subrede Mininet 2 também sofre com o ataque, embora o alvo do ataque tenha sido a subrede Mininet 1. Isso ocorre porque o controlador da rede também é atingido pelo ataque devido a uma problema conhecido como efeito de acoplamento da arquitetura SDN . Umas das consequências do efeito de acoplamento é atingir outras redes que não estão ligadas ao atacante, com isso, no cenário em pauta, a disponibilidade da subrede Mininet 2 é prejudicada. Na Tabela 3, é possível ver que a consolidação de regras também atua de forma positiva para contornar o problema de acoplamento e da indisponibilidade da subrede Mininet 1 causada pelo congestionamento da memória TCAM do switch SDN. Resultados semelhantes foram obtidos nos experimentos apresentados na Subseções 4.2 e 4.3.

Tabela 3. Experimentos com Ryu para topologia em árvore.

Com ataque e sem consolidação			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	0%	4915 ms	10096 ms
VM Mininet 2	2,47%	6416 ms	20718 ms
Sem ataque			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	100%	123 ms	1744 ms
VM Mininet 2	100%	120 ms	2144 ms
Con ataque e com consolidação			
Máquina virtual	Disponibilidade	TTS Médio	TTS Max.
VM Mininet 1	100%	380 ms	3948 ms
VM Mininet 2	100%	372 ms	2768 ms

5. Conclusão

Este artigo demonstrou que a aplicação da técnica de consolidação de regras pode reduzir significativamente o consumo de memória TCAM do switch SDN afetado pelo ataque. Isso resultou em uma capacidade maior de processamento de regras, tornando o switch mais resistente a ataques Slow-TCAM e garantindo a disponibilidade da rede. É importante ressaltar que a consolidação de regras não é uma solução única para todos os ataques de negação de serviço em redes SDN. Ela é especialmente eficaz contra o ataques DDoS de baixa taxa (*low rate*), como o Slow-TCAM. Outras medidas de segurança devem ser

adotadas em conjunto para proteger completamente as redes SDN contra outras ameaças. Como trabalhos futuros, novas estratégias de consolidação de regras serão testadas.

Referências

Alsaeedi, M., Mohamad, M. M., and Al-Roubaiey, A. A. (2019). Toward adaptive and scalable openflow-sdn flow control: A survey. *IEEE Access*, 7:107346–107379.

Altangerel, G., Chuluuntsetseg, T., and Yamkhin, D. (2019). Performance analysis of sdn controllers: Pox, floodlight and opendaylight.

Applegate, D. A., Calinescu, G., Johnson, D. S., Karloff, H., Ligett, K., and Wang, J. (2007). Compressing rectilinear pictures and minimizing access control lists. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1066–1075, USA. Society for Industrial and Applied Mathematics.

Curtis, A. R., Mogul, J. C., Tourrilhes, J., Yalagandula, P., Sharma, P., and Banerjee, S. (2011). Devoflow: Scaling flow management for high-performance networks. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, page 254–265, New York, NY, USA. Association for Computing Machinery.

Hong, K., Kim, Y., Choi, H., and Park, J. (2018). Sdn-assisted slow http ddos attack defense method. *IEEE communications letters*, 22(4):688–691.

Jiahao Cao, Mingwei Xu, Q. L. K. S. Y. Y. (2022). The loft attack: Overflowing sdn flow tables at a low rate. *IEEE/ACM TRANSACTIONS ON NETWORKING*, pages 1–16.

JMeter (2023). Jmeter. https://jmeter.apache.org/.

Lin, H.-T. and Wang, P.-C. (2023). Tcam-based packet classification for many-field rules of sdns. *Computer Communications*, 203:89–98.

Luo, S., Yu, H., and Li, L. (2015). Practical flow table aggregation in sdn. *Computer Networks*, 92:72–88.

Minh, Q. T., Van Le, A., Dang, T. K., Nam, T., and Kitahara, T. (2019). Flow aggregation for sdn-based delay-insensitive traffic control in mobile core networks. *IET Communications*, 13(8):1051–1060.

Pascoal, T. A., Dantas, Y. G., Fonseca, I. E., and Nigam, V. (2017). Slow tcam exhaustion ddos attack. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 17–31. Springer.

Pascoal, T. A., Fonseca, I. E., and Nigam, V. (2020). Slow denial-of-service attacks on software defined networks. *Comput. Networks*, 173:107223.

Punitha, V., Mala, C., and Rajagopalan, N. (2020). A novel deep learning model for detection of denial of service attacks in http traffic over internet. *International Journal of Ad Hoc and Ubiquitous Computing*, 33(4):240–256.

Yoon, C., Lee, S., Kang, H., Park, T., Shin, S., Yegneswaran, V., Porras, P., and Gu, G. (2017). Flow wars: Systemizing the attack surface and defenses in software-defined networks. *IEEE/ACM Trans. Netw.*, 25(6):3514–3530.

Yungaicela-Naula, N. M., Vargas-Rosales, C., Perez-Diaz, J. A., and Carrera, D. F. (2022). A flexible sdn-based framework for slow-rate ddos attack mitigation by using deep reinforcement learning. *Journal of network and computer applications*, 205:103444.