Redes Definidas por Software para a Detecção de Anomalias e Contra Medidas em Redes *Smart Grid*

R. C. C. Ferrari*, A. A. Shinoda*, C. M. Schweitzer**.

*Departamento de Engenharia Elétrica, Unesp, Ilha Solteira-SP, Brasil
**Departamento de Matemática, Unesp, Ilha Solteira-SP, Brasil

e-mail: rccferrari@hotmail.com

Abstract – *The work addresses the merging of Software Defined Networking (SDN) and Smart Grid technologies, enabling smart grids to take advantage of the SDN architecture, which centralizes every control plane in a controller, opening up opportunities for Smart Grid security applications. The work proposes an application with the use of standard deviation to define limits of maximum and minimum for detecting anomalies in the flows, besides detecting intrusive machines and blocking attacks originated from both of machines Intruding as slaves. In this way, two DDoS attacks were carried out, the first starting from intrusive machines and the second from slaves, allowing to analyze and monitor the dataflow and the port block of an OpenVSwitch (OVS).*

Palavras-chave: *SDN, Smart Grid, anomalias, DDoS, segurança.*

Introdução

Com o objetivo de aumentar a integração e um uso de energia mais inteligente, com recursos energéticos distribuídos, a *Smart Grid* (detalhada em [8]) foi proposta para substituir o sistema de energia convencional, possibilitando um sistema inteligente de energia e comunicação de dados. Uma diversidade de dispositivos e eletrodomésticos serão interligados e se comunicarão em uma rede inteligente para medição, monitoramento e controle.

Pesquisas recentes com *Smart Grid*, como apresentadas em [9, 11, 12], avaliam a utilização de um novo paradigma para auxiliar em seu gerenciamento, chamado de SDN (*Software Defined Network*). SDN vem despertando muita atenção devido a um novo conceito de arquitetura de rede que abstrai as funcionalidades de controle do hardware para um controlador de *software* externo (plano de controle). Isto insere a SDN como uma opção extremamente conveniente, possibilitando centralizar o plano de controle da rede em uma única máquina. Pelo fato do controlador ser implementado como *software*, permite o controle do fluxo nos comutadores de rede, proporcionando qualquer tipo de configuração ou autoconfiguração mais ágil. Desta forma, uma SDN é vista como uma opção interessante de utilização na comunicação de uma rede *Smart Grid*.

Com base em uma revisão bibliográfica realizada, várias pesquisas abordam a utilização de SDN para fornecer recursos para redes *Smart Grid*, porém não foi encontrado nenhum estudo que trate do monitoramento no fluxo de uma *Smart Grid* para detecção de anomalias para possíveis tentativas de ataques ou mau funcionamento da rede no fluxo de dados de um *switch* com suporte SDN ou em escravos individuais, levando em consideração o cálculo de desvio padrão em diferentes intervalos.

O trabalho apresentado por [8] ilustra 8 arquiteturas, onde são consideradas como candidatas para soluções SDN e *Smart Grid*, sendo que 3 apresentam soluções de segurança, descritas em [7, 1, 10]. A primeira arquitetura trata de sistemas baseados em IEC 61850 (*International Electrotechnical Commission*), capaz de realizar reencaminhamento de tráfego de dados, QoS (*Quality of Service*) e distribuição da carga de tráfego. Utiliza também o coletor de dados *sFlow* (solução pronta) para detecção de limite excedido e o comunica ao controlador. A segunda arquitetura descreve apenas um protótipo e não mostra nenhuma implementação ou resultados. Por fim, em [10], a solução proposta apresenta recursos como, estabelecer rotas dinâmicas, restabelecer um roteamento comprometido e troca de canais com cifragem, em possíveis ataques. O IDS (*Intrusion detection system*) está implementado fora do controlador, isso pode ocasionar maior custo de manutenção em uma atualização, pois são 2 *hosts* diferentes. Também não apresenta uma forma de identificar algum “estranho” na rede ou “eliminar” o atacante. A simulação não apresenta nenhuma forma de aprendizado para que o controlador possa monitorar a rede de forma mais precisa, levando em conta que cada rede pode se comportar de uma maneira diferente.

Motivado pelas oportunidades que uma SDN pode proporcionar para uma *Smart Grid* em

relação a sistemas de segurança e aplicação de outras técnicas para detecção de intrusos nesta rede, é proposta uma arquitetura capaz de calcular desvios padrões para diferentes momentos e analisar possíveis anomalias de fluxos em diferentes intervalos do monitoramento, além de detectar e eliminar nós que não fazem parte da rede pré-definida.

O artigo está organizado da seguinte forma: a segunda seção detalha a SDN e a *Smart Grid*. A terceira seção expõe os materiais e métodos. A quarta seção apresenta os resultados obtidos. Por fim, a conclusão do artigo.

SDN e Smart Grid

Apresentado por [8], uma *Smart Grid* é uma combinação entre redes de energia, redes de comunicação e sistemas de gerenciamento de informações, o que favorece a geração de energia verde e econômica. Geralmente, uma *Smart Grid* possui uma infraestrutura de comunicação de dados integrada com uma rede elétrica que coleta e analisa dados capturados em tempo de execução sobre a transmissão, distribuição e consumo de energia.

O *Distributed Network Protocol* (DNP3) é adequado para aplicação em ambiente *Supervisory Control and Data Acquisition* (SCADA), incluindo comunicações para *Intelligent Electronic Device* (IED), tendo a flexibilidade para suportar vários modos de operação, como resposta de leituras e respostas não solicitadas, além de permitir vários mestres e escravos [6].

Já uma SDN, como discutida em [8] é uma arquitetura de rede da próxima geração com serviços de gerenciamento, como filtragem de pacotes, comutação e monitoramento. Em uma SDN, um controlador de rede centralizado gerencia o plano de controle da rede em vez de *switches* de rede individuais. O fluxo de cada *switch* é reportado ao controlador SDN quando essa comunicação não está incluída em sua tabela de fluxo. O controlador SDN fornece a nova regra de fluxo ao *switch* através de uma interface (como o *OpenFlow)* para o plano de encaminhamento.

Foram desenvolvidos diversos controladores para o paradigma SDN. Porém, o controlador utilizado nesse trabalho foi o POX, sendo uma plataforma para o desenvolvimento, além de prototipagem de aplicações para SDNs usando *Python*.

Materiais e Métodos

O objetivo principal do trabalho é analisar o funcionamento do componente *flow_stats* modificado para a detecção e contra medidas de ataques DDoS, tanto para ataques originados de máquinas invasoras, como de escravos da rede *Smart Grid*, possibilitando identificar a eficiência da aplicação como um sistema de detecção de intrusos.

A Figura 1 apresenta o ambiente utilizado para testes. Para a realização foi utilizado o ambiente GENI (*Global Environment for Network Innovations*), uma infraestrutura aberta afim de proporcionar a pesquisa em redes em escala e sistemas distribuídos, sendo 24 máquinas escravas, 1 OVS, 1 sensor, 1 mestre e 1 controlador POX.

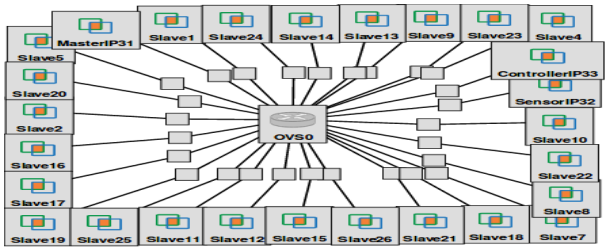

Figura 1. Ambiente de teste no GENI.

Foram utilizados os *softwares*, OVS 2.3.1 [2], GENI [5], *OpenDNP3* 1.1.0-RC1 [3] e o controlador POX 0.1.0-betta [4].

Para realizar os experimentos, foram utilizadas 27 máquinas interligadas através do OVS. Com isso, o mestre gera fluxo DNP3 contínuo para 16 escravos de aproximadamente 62 pacotes/min e 4830 *bytes*/min e as máquinas escravas respondem as solicitações com 31 pacotes/min e 2738 *bytes*/min, com a ferramenta *OpenDNP3*, possibilitando com isso o cálculo dos desvios em 12 intervalos de 2 horas (24 horas). O controlador da rede SDN está preparado para monitorar os fluxos necessários para as comunicações entre as máquinas escravas e o mestre da *Smart Grid* para os cálculos dos desvios padrões. Uma máquina sensor vasculha a rede durante o período de monitoramento, afim de auxiliar a detecção de novas máquinas na rede. No controlador foi empregado o POX para monitorar o tráfego entre as máquinas, para que fosse identificada máquinas intrusas na rede, possíveis ataques DDoS e anomalias nos fluxos individuais com o dobro do desvio.

Foram realizados dois ataques DDoS no mestre da rede *Smart Grid*, sendo o primeiro partindo de 8

máquinas intrusas e o segundo a partir de 8 escravos.

Após as finalizações dos cálculos dos desvios D1-D12, no período de 24 horas, com início às 10 horas, o controlador é capaz de iniciar o monitoramento do fluxo no *switch*, identificando possíveis máquinas invasoras, ataques e anomalias nos fluxos.

Para os cálculos dos desvios, foi definido um tempo de leitura da tabela de fluxo do OVS de 10 seg., com 720 leituras por intervalo, para determinar os conjuntos de 720 valores de pacotes e *bytes* utilizados para os cálculos dos desvios. Os valores definidos, como tempo de leitura, quantidade de leituras, IP do mestre, número de desvios, tempo de bloqueio, IP do sensor e número de desvios padrões para os fluxos e *switch* podem ser alterados em um arquivo de configuração.

Resultados

Após os testes, foram obtidos os resultados das análises realizadas pelo controlador nos intervalos 1-12, monitorados por 48 horas com início às 10:00 horas. O fluxo total no OVS, entre o mestre e 16 escravos, resultaram no mínimo de 28 e máximos de 489 falsos positivos no período 5 e 12 respectivamente, usando os respectivos valores de mínimo e máximo de pacotes, definidos a partir dos desvios padrões calculados pelo dobro, tanto para mais como para menos.

Observou-se que no intervalo 12 o número de anomalias foi alto, graças ao segundo ataque DDoS, provocando seguidas execuções do componente *host_tracker,* sendo realizado por escravos.

Contudo, foram gerados os gráficos das 48 horas de monitoramento, apresentados nas Figuras 2 e 3, apenas para os pacotes, já que o comportamento do fluxo de *bytes* é muito semelhante ao fluxo de pacotes.

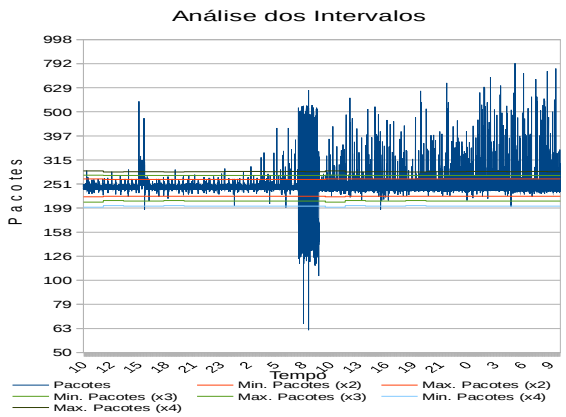

Figura 2. Monitoramento do total de pacotes no OVS.

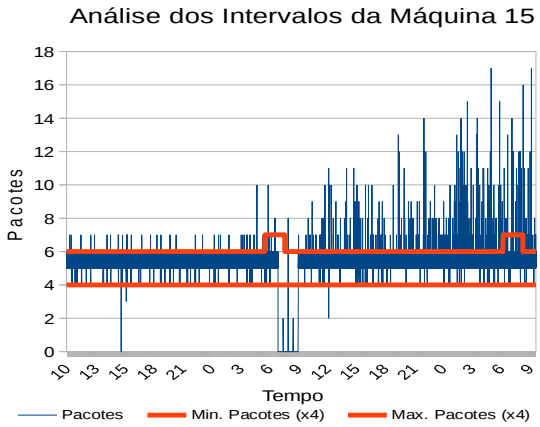

Figura 3. Monitoramento de pacotes máquina 15.

Assim, pode-se verificar o comportamento do fluxo em todo período de monitoramento e o momento que ocorreram as anomalias, tanto no OVS como na máquina 15 (escravo atacante). Os gráficos apresentam o número mínimo e máximo de pacotes, com o dobro, triplo e quádruplo do desvio padrão na Figura 2 e apenas o quádruplo na Figura 3 para uma melhor visualização.

A Figura 2 apresenta o comportamento do fluxo no decorrer do tempo, mostrando que próximo as 15 horas e 8 horas das primeiras 24 horas ocorreram duas anomalias no fluxo por consequência de dois ataques DDoS com duração de aproximadamente 35 minutos e 112 minutos respectivamente. Nas últimas 24 horas o fluxo apresentou um comportamento anômalo nas Figuras 2 e 3, isso se deve ao aumento do processamento do componente *host_tracker* que foi modificado para auxiliar na detecção de máquinas intrusas e ocasionou atrasos na leitura da tabela de fluxo, apresentando valores mais altos para o número de pacotes, contribuindo para um número maior de falsos positivos. Esse comportamento foi identificado em todos os fluxos.

Também podemos observar, que a Figura 2 apresenta anomalias para valores altos no primeiro ataque e altos e baixos no segundo ataque. Isso ocorreu pelo fato do primeiro ataque ser realizado por máquinas invasoras que apenas acrescentaram pacotes na rede durante o ataque. Já no segundo ataque os atacantes eram escravos que já estavam se comunicando, desta forma, o mestre continuou tentando conectar-se com os escravos bloqueados, o que acarretou em um fluxo aleatório no *switch* devido aos bloqueios das portas dos escravos. Ainda na Figura 2, os picos no início e no final do primeiro ataque é resultado da continuação do ataque DDoS pelas máquinas invasoras, sendo que o tempo de bloqueio de portas foi definido em 30 minutos.

Na Figura 3 é mais evidente os bloqueios de portas, justamente pela interrupção do fluxo decorrente do bloqueio.

No primeiro ataque a ausência de fluxo para a máquina 15 é sequela da limpeza da tabela de fluxo para liberar as portas que estavam bloqueadas. Porém, no segundo ataque a ausência de fluxo é exclusivamente pelo bloqueio de sua porta pelo tempo total de 2 horas, os 3 picos no período de bloqueio é pelo desbloqueio de sua porta após 30 minutos, bloqueando imediatamente depois de identificar a continuidade do ataque. Para os dois ataques o controlador evitou 99% dos pacotes enviados com os bloqueios das portas dos atacantes.

Conclusões

Foram obtidos diferentes características em cada intervalo no momento dos ataques, como momento da limpeza da tabela de fluxo e o fluxo no OVS quando atacado por máquinas invasoras e escravos da rede *Smart Gird.* Outro comportamento que chamou atenção foi o atraso nas leituras da tabela de fluxo com o processamento do componente *host_tracker.* Esse comportamento foi consequência do intervalo de 10 segundos na leitura da tabela de fluxo, pois com um tempo maior esse atraso seria evitado. Outro ponto importante foi o desvio padrão encontrado, com valor pequeno por consequência do fluxo contínuo, dando um intervalo de mínimo e máximo muito curto no experimento. Porém o número de desvios padrões é um parâmetro ajustável no arquivos de configurações.

Os resultados para os dois ataques no mestre da rede *Smart Grid* foram satisfatórios, com até 99% dos pacotes bloqueados pelo controlador e quase nenhuma anomalia no mestre.

Sabendo da importância em preservar a segurança em redes *Smart Grid*, o controlador POX modificado apresenta uma alternativa para complementar a integridade da rede, sendo capaz de monitorar não apenas o fluxo total do *switch*, mas de cada fluxo individualmente, além da arquitetura possuir um sensor para identificação de intrusos na rede e bloqueio de portas com novos fluxos no período de monitoramento.

Os próximos passos serão monitorar um número maior de escravos, afim de analisar o comportamento de fluxo em vários *switches*, além de evitar o uso do componente host_tracker, adicionando sua função no componente *flow_stats*.

Desta forma, podemos concluir que além do monitoramento detalhado dos fluxos, uma contramedida para impedir um ataque é fundamental, pois não foi encontrada uma arquitetura projetada com tal flexibilidade.

Referências

[1] A. Cahn; Hoyos, J.; Hulse, M.; Keller, E. Software-defined energy communication networks: From substation automation to future Smart Grids. In Proceedings of the 2013 IEEE International Conference on Smart Grid Communications (Smart GridComm), Vancouver, BC, Canada, 21–24 October 2013; pp. 558–563.

[2] Disponível em: http://openvswitch.org/. Acesso em: 10 maio 2017.

[3] Disponível em: https://github.com/gec/dnp3/. Acessado em: 5 julho 2017.

[4] Disponível em: https://github.com/noxrepo/pox. Acesso em: 10 maio 2017.

[5] Disponível em: https://portal.geni.net/. Acessado em: 5 julho 2017.

[6] Disponível em: https://www.dnp.org/Pages/AboutFeatures.aspx. Acessado em: 6 de julho 2017.

[7] E. Molina; Jacob, E.; Matias, J.; Moreira, N.; Astarloa, A. Using software defined networking to manage and control IEC 61850-based systems. Comput. Electr. Eng. 2015, 43, 142–154.

[8] J. Kim; Filali, F.; Ko, Y.-B. Trends and Potentials of the Smart Grid Infrastructure: From ICT Sub-System to SDN-Enabled Smart Grid Architecture. *Appl. Sci.* **2015**, *5*, 706-727.

[9] J. Zhang, B. C. Seet, T. T. Lie, C. H. Foh. Opportunities for software-defined networking in Smart Grid. In Proceedings of the 2013 9th International Conference on Information, Communications and Signal Processing (ICICS), Tainan, Tanwan, 10–13 December 2013; pp. 1–5.

[10] X. Dong, H. L., R. Tan, R. K. Iyer, Z. Kalbarczyk. Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges. In Proceedings of The 1st Cyber-Physical System Security Workshop (CPSS), April 14 -17, 2015, Singapore.

[11] X. Dong; Lin, H.; Tan, R.; Iyer, R.K.; Kalbarczyk, Z. Software-defined networking for Smart Grid resilience: Opportunities and challenges. In Proceedings of the 1st ACM Workshop on Cyber-Physical System Security, Singapore, 14–17 April 2015; ACM: New York, NY, USA; pp. 61–68.

[12] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, N. Venkatasubramanian. A Software Defined Networking Architecture for the Internet-of-Things, 2014 IEEE network operations and management symposium (NOMS), 1-9.