ANÁLISE DO PROTOCOLO IPSEC A PARTIR DA API OPENDATAPLANE EM AMBIENTE SOC PARA APLICAÇÃO EM SDN

Thales de Társis Cezare[1], João Gabriel Rangel Gonçalves [2]

[1]Universidade São Francisco, Fatec de Mogi Mirim, SPIN TECNOLOGIA, Itatiba, Mogi Mirim, Engenheiro Coelho, Brasil (thales.cezare@fatec.sp.gov.br)
[2] Fatec de Mogi Mirim, SPIN TECNOLOGIA, Mogi Mirim, Engenheiro Coelho, Brasil

Resumo: Em uma era com hardware de alta performance capaz de substituir vários servidores de baixo desempenho e softwares cada vez melhores, há também a necessidade de melhorar a segurança de rede, computadores como a Raspberry mostra-se satisfatório para o custo-benefício em que ela se encontra, sendo possível a utilização de uma API de Plano de Dados para melhorar a segurança de tráfego de dados. Este projeto visa realizar estudos nos mecanismos de segurança da informação, entender suas funcionalidades, realizar testes de desempenho na API OpenDataPlane (ODP), visa ainda, avaliar vários modelos de criptografia implementados na API ODP em um Raspberry PI 4, e analisar a viabilidade de uma possível implementação de novas funcionalidades nos serviços de criptografia no ODP.

Palavras-chave: Plano de Dados. Redes de Computadores. Raspberry. Criptografia, SDN, OpenDataPlane.

INTRODUÇÃO

O OpenDataPlane é uma API que permite que implementações forneçam independência da plataforma, aceleração automática de hardware e dimensionamento de CPU para aplicativos de rede, uma AP que descreve um modelo funcional para aplicativos de plano de dados. Uma das especificações interessantes do ODP é a capacidade de receber, manipular e transmitir dados de pacotes, as APIs do ODP não possuem modalidade preferida, no qual permite inovar a forma como essas funções podem ser realizadas em várias plataformas que oferecem implementações do ODP.

O ODP também fornece APIs de criptografia exigidos pelos aplicativos, contendo vários modelos de criptografia, é possível fazer testes de desempenho desses modelos e mensurar a viabilidade de implementação.

O Plano de Dados é componente de software de um roteador ou switch

relacionado ao roteamento, encaminhamento de dados, ou tráfego de usuário entre duas ou mais interfaces de rede. Os pacotes que passam entre roteadores e switchs para trafegar dados, usam o plano de dados.

O Plano de Dados precisa fornecer caminhos de alta velocidade, e, muita implementação está em hardware, isso não significa que redes virtualizadas não possuam um plano de dados, existem várias técnicas de software para fornecer esse caminho.

Exemplo de Plano de Dados: Envio de Tráfego HTTP para outro dispositivo na rede.

A motivação dessa pesquisa é a viabilidade de implementação de novos modelos criptográficos em ambientes de sistemas de rede, contribuindo assim, para a comunidade de desenvolvimento de código aberto.

O trabalho atual torna-se importante para servidores de rede que buscam melhora no desempenho e maior confiabilidade na segurança, inovação e fácil implementação de novos ambientes tanto em hardware quanto em software.

SERVIÇOS DE CRIPTOGRAFIA

O ODP oferece APIs para executar operações de criptografia exigidos pelos aplicativos, essas APIs são baseadas em sessões que oferecem serviços de descarregamento de algoritmos de criptografia, é oferecido também serviços de transferência de protocolos para IPSec usando um conjunto diferente de APIs (OPENDATAPLANE, 2020).

AH

O IP Authentication Header (AH) é usado para fornecer integridade sem conexão e autenticação de origem de dados para datagramas IP e para fornecer proteção contra replays. Este último serviço opcional

pode ser selecionado, pelo receptor, quando uma Associação de Segurança (SA) é estabelecida.

O AH fornece autenticação para o máximo possível do cabeçalho IP, bem como para dados de protocolo de próximo nível. No entanto, alguns campos de cabeçalho de IP podem mudar em trânsito e o valor desses campos, quando o pacote chega ao receptor, pode não ser previsível pelo remetente. Os valores de tais campos não podem ser protegidos por AH. Assim, a proteção fornecida ao cabeçalho IP pelo AH é fragmentada. O AH pode ser aplicado sozinho, em combinação com o IP Encapsulating Security Payload (ESP) ou de forma aninhada. Os serviços de segurança podem ser fornecidos entre um par de hosts em comunicação, entre um par de gateways de segurança em comunicação ou entre um gateway de segurança e um host. O ESP pode ser usado para fornecer os mesmos serviços de anti-reprodução e de integridade semelhantes, além de fornecer um serviço de confidencialidade (criptografia). (S.KENT,2005)

ESP

O cabeçalho Encapsulating Security Payload (ESP) foi projetado para fornecem uma combinação de serviços de segurança em IPv4 e IPv6. ESP pode ser aplicado sozinho, em combinação com AH, ou em aninhado. Os serviços de segurança podem ser fornecidos entre um par de comunicações de hosts, entre um par de gateways de segurança de comunicação ou entre um gateway de segurança e um host.

O cabeçalho ESP é inserido após o cabeçalho IP e antes do próximo cabeçalho do protocolo de camada (modo de transporte), ou antes, de um IP encapsulado cabeçalho (modo de túnel).

ESP pode ser usado para fornecer confidencialidade, origem de dados e autenticação, integridade sem conexão, um serviço anti-replay (uma forma de integridade de sequência parcial) e confidencialidade de fluxo de tráfego (limitado). O conjunto de serviços prestados depende das opções selecionado no momento do estabelecimento da Associação de Segurança (SA) e na localização da implementação em uma topologia de rede. (S.KENT,2005)

IPSEC

IPsec (Internet Protocol Security) é um conjunto de protocolos que fornece segurança para comunicações de Internet na camada IP. O uso atual mais comum de IPsec é fornecer um Virtual Private Network (VPN), entre dois locais (gateway-to-gateway) ou entre um usuário remoto e uma rede corporativa (host para gateway); isto também pode fornecer segurança ponta a ponta ou host a host. (IETF,2011)

NETWORK FUNCTION VIRTUALIZATION (NFV)

Com o aumento de redes de comunicação, aumentou-se também os custos de hardware e manutenção desses dispositivos, tornando-se algumas vezes caro demais para ser mantido, o Network Function Virtualization (NFV) pode ser considerada uma das soluções para esses problemas, sugerindo criação de softwares capazes de substituir equipamentos caros e de alto custo de funções de rede via virtualização, estes hardwares virtualizados podem ser encontrados em nuvem e em máquinas virtuais, deixando o custo muito mais barato e de fácil manuseio, permitindo também escalabilidade mais simples em qualquer cenário (UFRJ, 2020).

A implantação do NFV é possível graças aos equipamentos de alta qualidade que vieram com a computação em nuvem, dentre eles temos switches Ethernet que possibilitam o tráfego de informação entre máquinas virtuais e interfaces físicas. Além disso, a computação em nuvem contribui com métodos de otimização de recursos computacionais, permitindo associar aplicações virtuais diretamente ao núcleo da CPU, memórias e interfaces corretas, além da reinicialização de máquinas virtuais defeituosas (UFRJ, 2020 / CIENA 2020).

O ODP pode ser utilizado em conjunto com o NFV, pois uma rede virtual pode ser criada com o NFV para rodar uma API do OpenDataPlane com o objetivo de reduzir custos de hardware e manutenção e garantir que toda a infraestrutura funcione com eficiência e segurança.

SOFTWARE DEFINED NETWORK (SDN)

> Redes Definidas por Software (SDNs) se baseiam na separação dos planos de dados e de controle da rede, sendo que este se refere ao conjunto de funções, logicamente centralizado em controladores de rede, que influencia em como os pacotes são encaminhados a destinos na rede por elementos que aquele define para realizar tal tarefa por meio de uma interface de comunicação bem definida. Dessa forma, a inteligência da rede se concentra em sua maior parte no plano de controle, o qual pode potencialmente abrigar qualquer aplicação de rede que possibilite a implementação de melhores estratégias para encaminhamento de tráfego por inúmeros atuadores em diferentes granularidades. Consequentemente, SDN pode estabelecer algoritmos eficientes no plano de dados para atuar no balanceamento de carga em enlaces, tendo como critérios políticas que contenham quaisquer critérios que sejam úteis a esta tarefa. (Rosa e colaboradores, 2014).

MATERIAL E MÉTODOS

Foi coletado informações sobre desempenho de criptografias implementadas na API ODP, tanto em um computador com arquitetura x86, quanto em um Raspberry PI 4 com arquitetura ARM.

As informações coletadas foram filtradas, e seus dados foram usados para criação de gráficos para melhor análise.

Características de hardware e software dos ambientes preparados para medir desempenho da API pode ser observado na Tabela 1.

Tabela 1 - Características de hardware e software.

Computador x86	Raspberry PI 4
• Processador em arquitetura x86. Intel Core 2 Duo com 2 Nucleos a 2.2Ghz; • 4GB RAM; • Sistema Operacional: Ubuntu 18.04 (Baseado em Debian) e Windows.	• Processador em arquitetura ARM. Broadcom BCM2711 com 4 Nucleos a 1.5Ghz • 4GB RAM; • Sistema Operacional: Raspavam (Baseado em Debian)

Softwares Utilizados:

- OpenDataPlane e suas dependências;
- OriginLab (para análise de dados e gráficos);
- SSH (para envio de comandos para a Raspberry em remoto, o mesmo pode ser feito diretamente pelo Raspberry, eliminando a necessidade do SSH);
- Git (para baixar o ODP no GitHub de forma fácil e rápido).

Para a instalação da API, deve-se baixar a API localizado no GitHub do ODP:

- Entre em modo root em seu terminal;
- Instalar as dependências que o programa necessita para ser executado;
- Entre na pasta odp/, rode o comando ./bootstrap;
- Logo após, rode o comando ./configure, caso esteja em um Raspberry PI 4, rode o comando ./configure CFLAGS="-mcpu=cortex-a72 -mfloat-abi=hard -mfpu=neon-fp-armv8 -mneon-for-64bits"
- Configurado sem erros, utilize os comandos make && make install.

A API ODP já está instalada em sua máquina.

Para melhores práticas, é recomendado utilizar o passo a passo disponível no site OpenDataPlane.

Após a instalação correta da API, para obter os dados que serão apresentados, o aplicativo necessário pode ser encontrado em /odp/test/performance/odp_ipsec.

Rode o programa odp_ipsec, se preferir salvar os dados em um documento de texto, insira em seu terminal o seguinte programa: odp_ipsec >> (título de sua preferência.txt).

Com as informações salvas, será filtrado os dados das colunas payload (bytes) e elapsed (us) para serem analisadas em um software de análise de dados OriginLab, pode utilizar um software alternativo de sua preferência.

RESULTADOS E DISCUSSÃO

Dados os métodos de obter as informações que serão apresentadas, usaremos todos os tipos de criptografia presentes no odp_ipsec, vamos separar duas colunas específicas (payload e elapsed), e logo após gerar um gráfico para sua visualização, é esperado um gráfico semelhante ao gráfico da Figura 1 (lembrando que cada máquina terá um desempenho diferente).

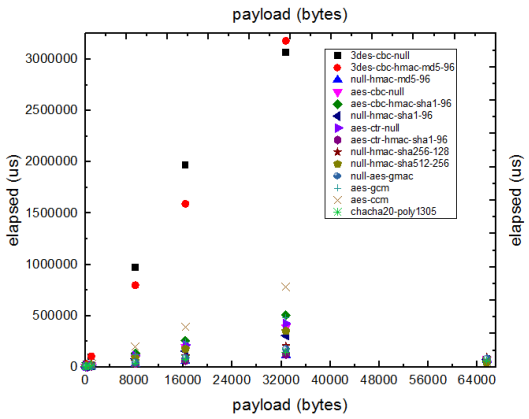

Figura 1: Benchmark de criptografia com Raspberry PI 4.

Usando o Raspberry PI 4 com arquitetura ARM, pode-se notar uma desvantagem considerável em criptografias com 3des, tanto em modo criptografia, quanto em criptografia + hash, sendo muito mais lentas do que as demais.

Separando o gráfico em 3 partes, autenticação (Figura 2), criptografia (Figura 3) e criptografia + autenticação (Figura 4).

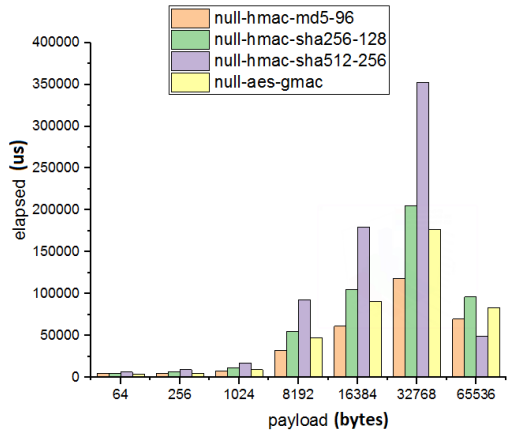

Figura 2: Comparativo RPI4 Autenticação.

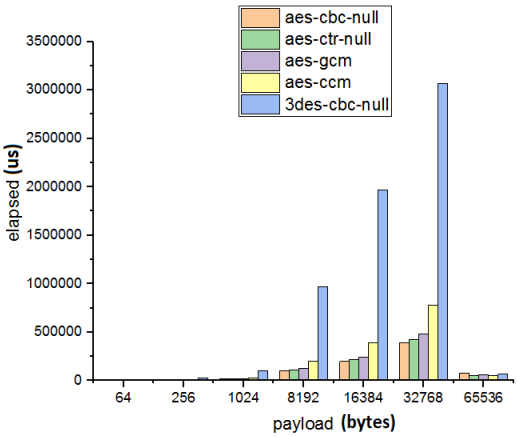

Figura 3: Comparativo RPI4 Criptografia.

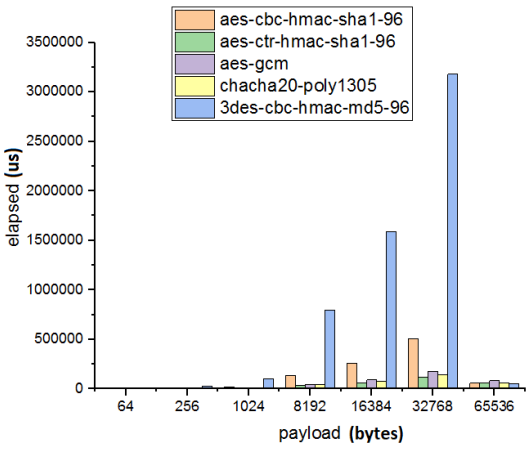

Figura 4: Comparativo RPI4 Criptografia + Autenticação.

O comparativo de autenticação apresentou os melhores resultados, especificamente os hashes null-hmac-md5-96 e null-hmac-sha1-96, são códigos de autenticação de mensagens, necessitam de um hash (neste caso sha1 e md5) e uma chave secreta, porém neste caso, não se tem uma cifra específica.

A seguir, são apresentados os dados em tabelas.

Tabela 1. Comparativo Autenticação.

Payload (bytes)	null-hmac-md5-96 (us)	null-hmac-sha256-128 (us)	null-hmac-sha512-256 (us)	null-aes-gmac (us)
64	4623	5419	6721	4166
256	5301	6621	9485	5176
1024	7973	11340	17638	9281
8192	32858	55115	93221	47217
16384	61318	105219	179533	90483
32768	118389	205267	352327	177175
65536	69876	96184	49458	83706

Tabela 2. Comparativo Criptografias.

Payload (bytes)	aes-cbc-null (us)	aes-ctr-null (us)	aes-gcm (us)	aes-ccm (us)	3des-cbc-null (us)
64	5155	6444	5162	5543	9151
256	7437	9884	8825	10055	27047
1024	16472	19907	20029	28294	98661
8192	100620	111405	124128	197989	970115
16384	196742	215631	243387	391758	1968840
32768	389804	424231	481503	780242	3065881
65536	76926	55076	64596	51966	66161

Tabela 3. Comparativo Criptografias com autenticação.

Payload (bytes)	aes-cbc-hmac-sha1-96 (us)	aes-ctr-hmac-sha1-96 (us)	aes-gcm (us)	chacha20-poly1305 (us)	3des-cbc-hmac-md5-96 (us)
64	7523	4598	4166	5400	11677
256	10478	5306	5176	6729	30258
1024	22198	8031	9281	9842	104459
8192	131384	33605	47217	39830	797281
16384	256116	62826	90483	74225	1591741
32768	506440	122000	177175	144488	3179794
65536	57604	56068	83706	61028	55423

Nestes modelos apresentados, os mais rápidos para a API ODP são novamente as de autenticação, notamos que em payloads de 65536 bytes, o Raspberry reduz consideravelmente o tempo em comparação a cargas de 32768 bytes. A implementação de novas criptografias pode ajudar a API a crescer sua comunidade de pessoas que o fazem uso, melhorando também toda a infraestrutura de comunicação de dados.

CONCLUSÃO

As vantagens dessa API de plano de dados estão relacionadas a facilidade de implementação para seu uso em redes definidas por software (SDN), como OpenFlow, OpenvSwitch entre outras, que permitem uma escalabilidade, custo, segurança e manutenção melhores do que as presente em redes por hardware.

A análise feita pode ser entendida como um ponto de partida para uma possível implementação de criptografia para esta API, é esperado novos estudos nesse mesmo assunto para melhores interpretações.

Nesta pesquisa foram abordados os conceitos e uma breve análise de criptografia em uma arquitetura de comunicação de rede, especificamente o plano de dados, os resultados coletados foram analisados e filtrados, gerados gráficos e revisados considerando a função de cada tipo de criptografia, é de se esperar que a segurança desse tipo de comunicação cresça consideravelmente nos próximos anos.

AGRADECIMENTOS

Agradecemos as entidades que nos recebe profissionalmente como pesquisadores.

REFERÊNCIAS

CIENA, O que é NFV? – Ciena. Disponível em: <https://www.ciena.com.br/insights/what-is/What-is-Network-Functions-Virtualization_pt_BR.html/>. Acesso em Outubro de 2020.

Krawczyk, H.; Bellare, M.; Canetti R.; HMAC: Keyed-Hashing for Message Authentication. Disponível em: https://tools.ietf.org/html/rfc2104 Acesso em: Outubro de 2020.

OPENDATAPLANE, OpenDataPlane (ODP) Users-Guide. Disponível em: <https://opendataplane.github.io/odp/users-guide//>. Acesso em: Outubro de 2020

Pellegrini, J.; Introdução à Criptografia e seus Fundamentos notas de aula - versão: 2019.11.28.20.34. Disponível em: http://aleph0.info/cursos/ic/notas/cripto.pdf Acesso em: Outubro de 2020.

Rosa, R.; Siqueira, M.; Barea, E.; Marcondes, C.; Rothenberg, C.; Network Function Virtualization: Perspectivas, Realidades e Desafios. Disponível em: http://www.dca.fee.unicamp.br/~chesteve/pubs/MC-SBRC14-NFV.pdf Acesso em: Outubro de 2020.

S.KENT, IP Authentication Header. Disponível em: <https://tools.ietf.org/html/rfc4302/>. Acesso em: Outubro de 2020.

S.KENT, IP Encapsulating Security Payload (ESP). Disponível em: <https://tools.ietf.org/html/rfc4303/>. Acesso em: Outubro de 2020.

UFRJ, NFV. Disponivel em: https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-v1/nfv//. Acesso em: Outubro de 2020.