SOLUÇÕES DE SEGURANÇA EM REDES UTILIZANDO A LINGUAGEM P4

Pili Andwele Rodrigues Antunes – piliandwele1@gmail.com – (12) 981-792-172

FACULDADE DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO

Linguagem P4 - Segurança - Criptografia - Roteadores

Introdução

Pessoas e empresas estão cada vez mais dependentes da troca de informações através de redes de computadores e muitas vezes é de grande interesse manter esta informação restrita ao destinatário real, sendo este o propósito, a criptografia da informação trafegada é fundamental para evitar que dispositivos intermediários da rede possam ter acesso a esta informação.

Como a linguagem P4 é independente de protocolos de comunicação e pode ser compilada para execução em múltiplas plataformas é bastante conveniente a possibilidade de criar rotinas de encaminhamento, aprimoramento de desempenho e segurança nesta linguagem.

Se tratando de uma linguagem relativamente nova, ela possui um número limitado de funções primitivas que podem ser utilizadas pelo programador diretamente, para isto foram criadas duas funções, uma para cifrar os dados e outra para traduzir ao estado original.

Neste estudo procurou-se resolver o problema de segurança no trafego de informações entre os dispositivos encaminhadores, sendo que a segurança do trajeto envolvendo o emissor e primeiro dispositivo encaminhador e o trajeto do último dispositivo até o destino não são tratados por esta solução.

A topologia de rede utilizada para esta solução de segurança envolve a utilização de um dispositivo encaminhador P4 conectado a cada grupo de dispositivos que se deseja proteger, funcionando como um gateway para a rede externa.

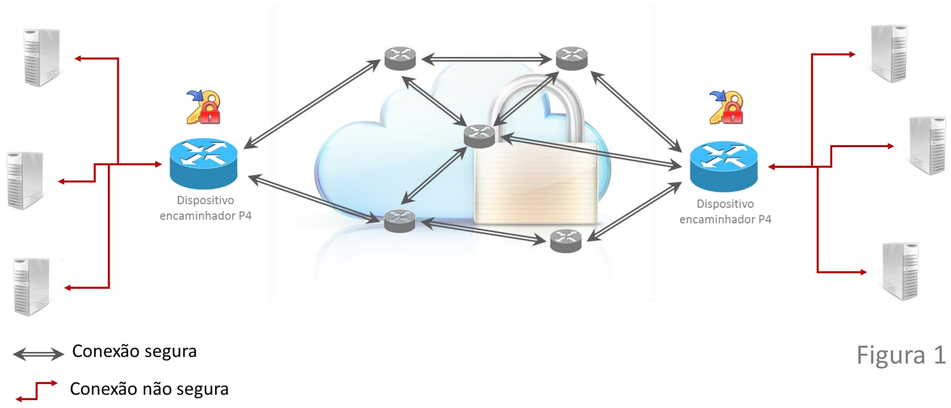


Figura 1

Metodologia

Topologia de rede:

Para os testes foi montada uma topologia de rede equivalente, representada por dois hosts e dois roteadores P4, como descrito na figura [2].

Roteador P4:

Como dispositivo de encaminhamento de pacotes foi utilizado um roteador codificado em linguagem P4. Este roteador é capaz de criptografar pacotes transmitidos usando os protocolos Ethernet na camada de enlace, Ipv4 na camada de rede e TCP na camada de transporte, sendo que pacotes transmitis utilizando outros protocolos são apenas roteados sem criptografia. A figura 3 representa o diagrama de alto nível para o processamento de pacotes utilizado pelo roteador.

Algoritmo de criptografia:

Para a implementação das funções primitivas de criptografia foi utilizado o algoritmo RC4 (ou ARC4), que por efetuar basicamente permutações e soma de inteiros possui um alto desempenho e com isso não introduz atrasos significativos na comunicação. Outro fato que levou a escolha deste algoritmo foi a necessidade de se obter uma saída com o mesmo tamanho da entrada devido a uma limitação da linguagem P4.

Este algoritmo trabalha basicamente com transformações lineares, portanto não são necessários cálculos complexos, já que o sistema funciona basicamente por permutações e somas de valores inteiros, o que torna este algoritmo relativamente simples e rápido.

Funções P4:

Foram introduzidas nas funções primitivas da linguagem P4 as funções de criptografia (encryption_routine) e decriptografia (decryption_routine). As funções foram codificadas em C++ e incluídas na biblioteca de funções da linguagem. Uma referência para estas foi criada no Behavior Model V2 (compilador para Behavior Model targets), de modo que o compilador consegue interpretar as funções.

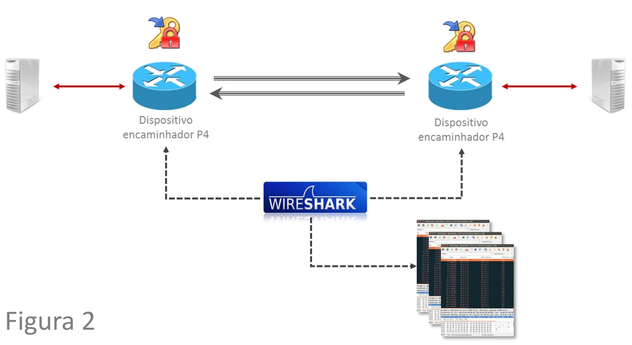


Figura 2

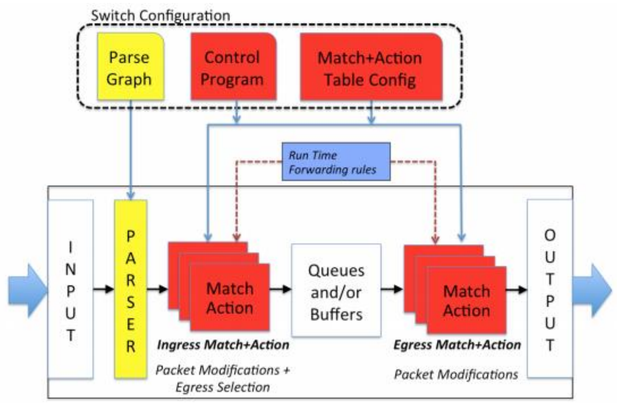


Figura 3

Resultados

Testes de integridade do pacote:

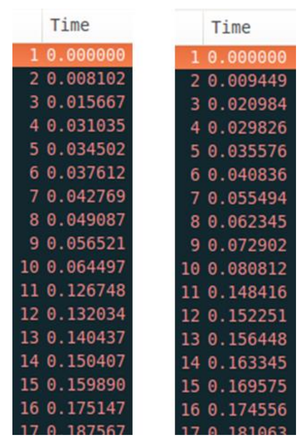
Foram utilizados nos testes uma série de redes virtuais as quais foram associadas o roteador P4 e monitoradas utilizando a aplicação WireShark, esta aplicação monitora as redes virtuais e exibe o que está sendo trafegado por elas, sendo assim, é possível identificar os pacotes e seu conteúdo, criptografados e decriptografados.

Para os testes foi construído um pacote com headers Ethernet, Ipv4 e TCP com um payload de tamanho fixo, devido a uma limitação da linguagem P4. Como os headers do pacote são preservados, a utilização deste método de criptografia não limita a circulação dos pacotes apenas por roteadores ou protocolos específicos.

Os dados coletados via WireShark dos pacotes que saem do primeiro roteador mostraram que apenas o seu payload foi alterado pela função de criptografia como o esperado. Os dados coletados na saída do segundo roteador mostraram que o payload do pacote foi restaurado ao que era antes da criptografia, completando o processo de envio de informação ao host de destino.

Testes de desempenho:

Nos testes de desempenho foram enviados uma série de pacotes na rede utilizando apenas um roteador, como mostrado na figura [4], o delay adicionado com a função de criptografia não alterou de maneira significativa o desempenho do sistema para pacotes de até 1Kbyte de payload, sendo a maior variação de desempenho percebida causada por outros fatores.



	Time
1	0.000000
2	0.008102
3	0.015667
4	0.031035
5	0.034502
6	0.037612
7	0.042769
8	0.049087
9	0.056521
10	0.064497
11	0.126748
12	0.132034
13	0.140437
14	0.150407
15	0.159890
16	0.175147
17	0.187567

Não Criptografado

	Time
1	0.000000
2	0.009449
3	0.020984
4	0.029826
5	0.035576
6	0.040836
7	0.055494
8	0.062345
9	0.072902
10	0.080812
11	0.148416
12	0.152251
13	0.156448
14	0.163345
15	0.169575
16	0.174556

Criptografado

Figura 4

Conclusão

As funções primitivas de criptografia implementadas podem ser utilizadas nas funções do tipo "*Action*" como qualquer outra primitiva, sendo possível assim associá-las as tabelas de *"Match+Action"* para selecionar a quais pacotes e em que condições a função deve ser aplicada em tempo de execução. Devido a limitação na declaração de headers de tamanho dinâmico na linguagem P4 e a impossibilidade de utilizar a estrutura *field_list* como entrada das funções primitivas foi necessário definir um tamanho mínimo para o pacote, de modo que o *parser* fosse capaz de interpretar o payload do pacote.

As funções de criptografia apresentaram um bom desempenho, de modo que não houve atraso significativo no encaminhamento dos pacotes.

Referências bibliográficas

The P4 Language Consortium "The P4 Language Specification" Version 1.0.2 March 3, 2015 Disponível em: http://p4.org/wp-content/uploads/2015/04/p4-latest.pdf acessado em 10 de Junho de 2016

Whitley B., "Implementing the RC4 Algorithm" Computer Science 3 - CSI 3050 Term Paper (2004)