CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

SISTEMA DE SEGURANÇA PARA DETECTAR E IMPEDIR ATAQUES DE NEGAÇÃO DE SERVIÇO EM SERVIDORES SIP

MARCUS AURÉLIO ARAÚJO ANDRADE

Orientador: Luciano Nascimento Moreira
CEFET-MG

TIMÓTEO
DEZEMBRO DE 2019

MARCUS AURÉLIO ARAÚJO ANDRADE

SISTEMA DE SEGURANÇA PARA DETECTAR E IMPEDIR ATAQUES DE NEGAÇÃO DE SERVIÇO EM SERVIDORES SIP

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Luciano Nascimento Moreira
CEFET-MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
TIMÓTEO
DEZEMBRO DE 2019

Marcus Aurélio Araújo Andrade

SISTEMA DE SEGURANÇA PARA DETECTAR E IMPEDIR ATAQUES DE NEGAÇÃO DE SERVIÇO EM SERVIDORES SIP

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 09 de dezembro de 2019:

Prof. Me. Luciano Nascimento Moreira
Orientador

Prof. Me. Adilson Mendes Ricardo
Professor Convidado

Prof. Dr. Maurílio Alves Martins da Costa
Professor Convidado

Timóteo
2019

Resumo

Em conjunto com o avanço das tecnologias surgiram diversas aplicações com o propósito de melhorar os serviços já conhecidos, como é o caso da tecnologia VoIP que permite iniciar a comunicação entre dispositivos por meio da Internet. As aplicações VoIP realizam chamadas entre si utilizando o protocolo SIP, sendo ele o responsável por intermediar a troca de mensagens entre os dispositivos envolvidos em uma chamada. Portanto, se um servidor SIP ficar indisponível todo o serviço VoIP será comprometido, podendo causar prejuízos para as empresas e usuários que utilizam essa tecnologia. Baseando no aumento do número de ameaças direcionadas a servidores, este trabalho apresenta um sistema para impedir que os ataques DoS cheguem ao servidor que utiliza o protocolo SIP. De forma a atingir esse objetivo, a aplicação desenvolvida trabalhou em conjunto com o Firewall FreeBSD pfSense e a ferramenta de monitoramento de rede Snort. Durante a simulação do ataque com o ambiente desprotegido, o consumo médio de CPU do servidor foi de 37,44%, aumentando 32,78% em relação ao período onde não ocorria um ataque. Utilizando o sistema deste trabalho o consumo de CPU atingiu o valor médio de 5,36% durante o ataque DoS, implicando em uma diminuição de 32,08% em comparação com os resultados obtidos no experimento com o ambiente desprotegido.

Palavras-chave: SIP, VoIP, Firewall, DoS, Snort, Segurança.

Abstract

Along with the advancement of technologies, several applications have emerged to improve popular services, such as VoIP technology, which allows communication between devices over the Internet. VoIP applications make calls to each other using the SIP protocol, which is responsible for intermediating the exchange of messages between the devices involved in a call. Therefore, if a SIP server becomes unavailable, the entire VoIP service will be compromised, which could cause losses for companies and users using this technology. Based on the increase in the number of threats directed at the servers, this work presents a system to prevent DoS attacks from reaching the server using the SIP protocol. To achieve this goal, the developed application worked together with the Firewall FreeBSD pfSense and the network monitoring tool Snort. During the simulation of the attack with the unprotected environment, the average CPU consumption of the server was 37,44%, increasing 32,78% in relation to the period when an attack did not occur. Using the system of this work, the CPU consumption reached an average value of 5,36 % during the DoS attack, implying a decrease of 32,08% when compared with the results obtained in the experiment with the unprotected environment.

Keywords: SIP, VoIP, Firewall, DoS, Snort, Security.

Lista de Figuras

Lista de Tabelas

Lista de Quadros

Lista de Abreviaturas e Siglas

VoIP	*Voz sobre IP*
SIP	*Protocolo de Inicialização de Sessão*
DoS	*Ataque de Negação de Serviço*
DDoS	*Ataque Distribuído de Negação de Serviço*
IP	*Internet Protocol*
IPS	*Sistema de Prevenção de Intrusão*
IDS	*Sistema de Detecção de Intrusão*
rps	*Requisições por segundo*
bps	*Bits por segundo*

Sumário

1 Introdução

A transmissão de Voz sobre IP (VoIP) é um serviço que utiliza a Internet para a transmissão de voz, sendo muito usado por ter menores custos de implantação e uso em relação à telefonia tradicional. Não é necessário instalar novos cabos de fibra e usuários não precisam contratar um serviço próprio de telefonia devido ao uso da Internet para iniciar as chamadas telefônicas (TANENBAUM, 2003).

O Protocolo de Inicialização de Sessão (SIP) utilizado em aplicações VoIP é responsável por iniciar, alterar e encerrar as sessões multimídia entre os canais de comunicação. Semelhante ao Protocolo de Transferência de Hipertexto (HTTP), o SIP é baseado em um modelo de solicitações que envia mensagens para um servidor e espera as devidas respostas (FOROUZAN, 2008). Este cenário possibilita a um intruso enviar múltiplas requisições de mensagens para um ou vários servidores SIP em curtos períodos de tempo, causando sobrecarga de recursos do servidor e da rede. Este alto número de solicitações recebe o nome de ataque de negação de serviço (DoS). A variante deste problema é o ataque distribuído de negação de serviço (DDoS), que ocorre quando são feitos múltiplos acessos por muitos dispositivos em pequenos intervalos de tempo. Esses ataques têm o objetivo de interromper e diminuir a qualidade do serviço por meio dos altos consumos de banda da rede, processamento e memória do servidor (TANENBAUM, 2003).

O intruso pode usar diferentes estratégias para diminuir o desempenho do serviço VoIP, uma delas são ataques DoS direcionados ao servidor de registro SIP, em que são enviadas centenas de solicitações de Registro por segundo. Este problema também ocorre com solicitações de Convite, que podem ser direcionadas ao servidor proxy do serviço VoIP. Muitas vezes estes servidores não possuem um sistema de segurança para bloquear tais solicitações (PARK, 2009).

Em uma pesquisa sobre DDoS realizada pelo Instituto SANS, foram entrevistados 378 gerentes de rede e segurança, onde foi identificado que estes ataques estão mais sofisticados e ocorrendo com maior frequência. Cerca de 40% das empresas que participaram da entrevista estão vulneráveis e 50% nunca fizeram testes da capacidade de suportar DDoS na rede. Em média, os entrevistados presenciaram 4,5 ataques DDoS por ano (PERCATORE, 2014). Essas informações são reforçadas pela Kaspersky (2017), que identificou em um estudo que muitas empresas não estão preparadas para se protegerem de ataques DDoS.

As ameaças mencionadas têm a capacidade de atingir grandes números de pessoas e empresas por meio da Internet, a qual permite armazenar segredos comerciais e estratégias de marketing que podem causar prejuízos financeiros se forem descobertas por pessoas mal intencionadas. Uma ferramenta utilizada para proteger a rede de eventuais ameaças é o Firewall, pois possibilita ao administrador da rede configurar regras de bloqueio visando filtrar pacotes e descartar aqueles que falharem nos testes (TANENBAUM; WETHERALL, 2011).

Para descobrir os prejuízos causados pelos ataques DDoS, a Kaspersky e B2B (2016) realizaram um estudo entre setembro de 2015 e setembro de 2016. Foram entrevistados mais de 4.000 representantes de pequenas, médias e grandes empresas distribuídas em 25 países. O resultado mostrou que o custo médio de um ataque DDoS é de aproximadamente $106.000, podendo chegar a mais $1.6 milhões para grandes empresas. Além disso, foi identificado que os custos podem ser reduzidos pela metade se o ataque for descoberto nas primeiras 24 horas.

Neste contexto, alguns mecanismos de defesa podem atuar em conjunto com o Firewall na segurança da rede, os quais são conhecidos como Sistema de Detecção de Intrusão (IDS) e Sistema de Prevenção de Intrusão (IPS). Esses sistemas usam regras para identificar anomalias e produzir alertas aos administradores da rede responsáveis pela análise dos alarmes. Esta é uma área que ainda está em desenvolvimento, embora existam diversas ferramentas disponíveis no mercado (PETERSON; DAVIE, 2013).

Em virtude dos fatos mencionados, observou-se que os mecanismos de defesa aplicam regras de bloqueio estáticas para ambientes dinâmicos. Portanto, torna-se necessário desenvolver um sistema para gerenciar essas regras com a finalidade de criar um ambiente adaptável a mudanças no padrão de comportamento da rede.

Visto que a proposta deste trabalho será capaz de gerenciar o Firewall de forma automática, espera-se que os resultados sejam melhores quando comparados com os dados obtidos antes da implantação da aplicação. Em suma, reduzir a carga de processamento do servidor SIP e diminuir o consumo de banda da rede são resultados pretendidos.

1.1 Objetivos

Esta seção demonstra os objetivos deste trabalho divididos nas categorias Geral e Específicos.

1.1.1 Geral

O objetivo geral é desenvolver um sistema de segurança para servidores SIP que permita bloquear ataques de negação de serviço. Como critérios de avaliação, serão levados em consideração a carga de processamento do servidor e a largura de banda da rede.

1.1.2 Específicos

- Definir o ambiente de rede a ser utilizado nos experimentos.

- Analisar a eficiência através da simulação de ataques de negação de serviço, comparando os resultados de antes e após a implantação da proposta.

2 Fundamentação Teórica

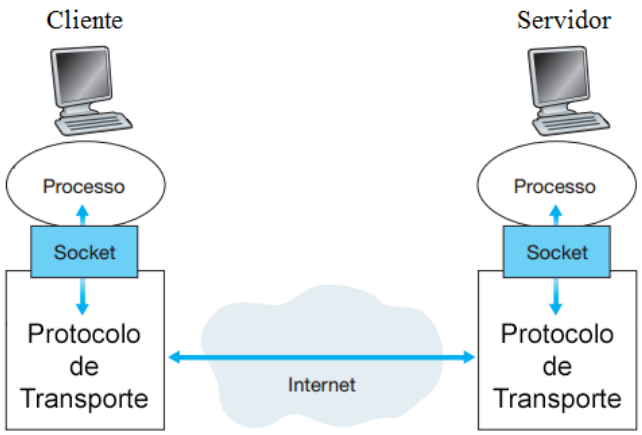
Este capítulo descreve os conceitos, protocolos e ferramentas utilizadas no desenvolvimento deste trabalho.

2.1 Comunicação entre processos

Os processos podem ser vistos como programas dentro das aplicações, sendo responsáveis pela troca de informações entre dispositivos. Por exemplo, clientes e servidores se comunicam por meio de processos de aplicações. Uma vez que existem diversas aplicações em um sistema operacional, a interface entre o processo da aplicação e o protocolo de transporte denominada *socket* é utilizada para entregar os dados para o destinatário correto (KUROSE; ROSS, 2014).

A Figura 1 mostra como é feita a transmissão de dados entre processos utilizando o *socket*, que contém a porta da aplicação e o endereço IP do destinatário.

Figura 1 – Comunicação entre processos



Fonte: Adaptado de (KUROSE; ROSS, 2014).

Os protocolos de transporte são utilizados para realizar a comunicação entre os dispositivos a fim de estabelecer como os dados irão trafegar entre as redes (TANENBAUM; WETHERALL, 2011). Dois destes protocolos de transporte são definidos nos itens abaixo.

- **Protocolo de Controle de Transporte (TCP):** Orientado a conexão, estabelece um caminho virtual antes de iniciar a transmissão de dados. Garante a entrega dos pacotes em ordem e sem erros, retransmitindo caso seja necessário (FOROUZAN,

2008). Utiliza um controle de fluxo, impedindo que o emissor sobrecarregue o receptor enquanto envia os pacotes (TANENBAUM; WETHERALL, 2011).

- **Protocolo de Datagrama de Usuário (UDP):** Não garante que os pacotes enviados serão recebidos, pois não é orientado a conexão. Os dados não são retransmitidos se forem perdidos ou corrompidos. Por ser mais rápido que o TCP, é mais utilizado quando a velocidade é um requisito superior a precisão da entrega, como ocorre na transmissão de voz em tempo real (TANENBAUM; WETHERALL, 2011).

A quantidade de pacotes que podem ser enviados e recebidos depende da largura de banda, a qual define a capacidade máxima de transmissão de bits em certo período de tempo que geralmente é medido em segundos (PETERSON; DAVIE, 2013). Considerando uma rede com largura de banda equivalente a 100 bps (bits por segundo), se for gerado um trafego com uma taxa de 400 bps, a rede será sobrecarregada e haverá perda de desempenho. Portanto, a largura de banda é proporcional à quantidade de bits que poderão ser transmitidos em intervalos de tempo.

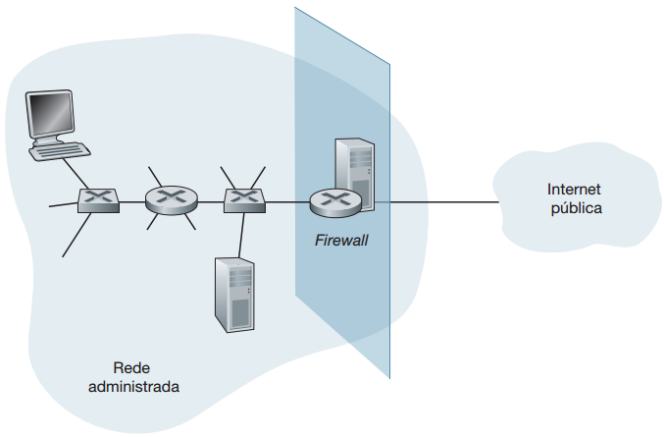
2.2 Segurança

Uma parte importante do sistema de segurança é a autorização. Este termo refere-se aos recursos que poderão ser utilizados ou acessados por algum processo. A partir do momento que os serviços da aplicação estão acessíveis através da Internet, torna-se necessário controlar o acesso dos usuários por meio de ferramentas, as quais são conceituadas nos identificadores destacados abaixo (TANENBAUM; STEEN, 2008).

Firewall: Isola os componentes da rede dos usuários externos, impedindo que informações não autorizadas sejam enviadas para usuários e servidores internos. Os pacotes de entrada e saída são inspecionados e repassados se não restringirem as regras de segurança configuradas (TANENBAUM; STEEN, 2008). Alem disso, "permite a um administrador de rede controlar o acesso entre o mundo externo e os recursos da rede que ele administra, gerenciando o fluxo de tráfego de e para esses recursos."(KUROSE; ROSS, 2014, p. 538). Portanto, o Firewall funciona como um filtro de pacotes, descartando aqueles que não cumprem os critérios definidos pelo administrador da rede (TANENBAUM; WETHERALL, 2011).

Como mostra a Figura 2, todos os pacotes que trafegam entre a Internet pública e a rede administrada passam pelo filtro de pacotes do Firewall, que analisa os campos de cabeçalho IP, UDP e TCP. Por meio dessa inspeção, descarta ou encaminha os pacotes conforme a política de segurança da rede, a qual descreve como os recursos devem ser manipulados

Figura 2 – Firewall entre as redes pública e privada



Fonte: (KUROSE; ROSS, 2014).

pelos usuários.

De acordo com (KUROSE; ROSS, 2014), os critérios de filtragem do firewall sao definidos conforme a política de segurança da rede e podem ser configurados da seguinte forma:

- Filtro de protocolo: Impõe restrições aos protocolos UDP e TCP por meio da análise de pacotes.
- Filtro de porta: Adiciona restrições baseadas nas portas de origem e destino dos processos.
- Largura de banda: Limita a utilização da largura de banda por usuário.
- Filtro de IP: Define os endereços IP de origem e destino a serem bloqueados.

Firewall pfSense: É um sistema operacional de código aberto que permite transformar computadores em um Firewall, podendo atuar no roteamento e balanceamento de carga se estiver configurado para essa finalidade (WILLIAMSON, 2011). Um arquivo XML chamado config.xml é usado para armazenar todas as configurações dos serviços disponíveis no pfSense, isso possibilita a transferência e réplica de um ambiente em funcionamento com facilidade por meio de uma cópia desse arquivo. Dessa maneira a máquina pode ser restaurada ou configurada apenas copiando o arquivo config.xml (RIBEIRO; PEREIRA, 2009).

Sistema de Detecção de Intrusão (IDS): Analisa os pacotes da rede e emite alertas para o administrador da organização quando alguma atividade suspeita é detectada, como por exemplo, ataques DoS e DDoS.

Sistema de Prevenção de Intrusão (IPS): Por meio da análise dos pacotes, descarta aqueles que são classificados como suspeitos.

2.3 Protocolo de Inicialização de Sessão

Ao acessar algum site da Internet, várias mensagens são trocadas entre o processo da aplicação do cliente e o processo da aplicação do servidor. Essa comunicação é realizada por meio do Protocolo de Transferência de Hipertexto (HTTP) que é responsável pelas regras da troca de mensagens (KUROSE; ROSS, 2014).

De forma análoga, o Protocolo de Inicialização de Sessão (SIP) transmite mensagens de sinalização para se comunicar com servidores e aplicações de outros usuários. Sendo responsável por iniciar, controlar, finalizar as sessões multimídia e, junto com servidores auxiliares, descobrir a localização dos usuários através da troca de mensagens (PETERSON; DAVIE, 2013). O comportamento desses servidores é explicado nos itens destacados abaixo.

Servidor de localização: Sempre que o usuário inicia a aplicação SIP, uma mensagem é enviada ao servidor de registro para informar o endereço de IP atual associado com uma identificação de usuário, como por exemplo, bob@domain.com. A Figura 3 mostra uma mensagem de registro enviada do usuário Bob para servidor de localização domain.com.

Figura 3 – Mensagem de registro SIP

```
REGISTER sip:domain.com SIP/2.0
Via: SIP/2.0/UDP 193.64.210.89
From: sip:bob@domain.com
To: sip:bob@domain.com
Expires: 3600
```

Fonte: (KUROSE; ROSS, 2014).

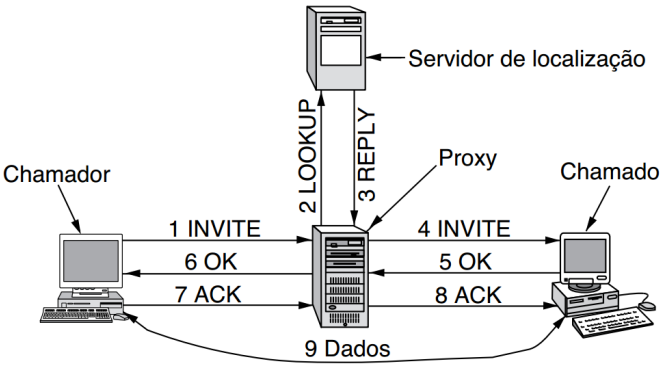
Nesse exemplo o usuário Bob enviou em um pacote UDP o endereço de IP atual 193.64.210.89, sendo transferido por meio do cabeçalho *Via*. Uma vez registrado no servidor, Bob pode ser contactado por outro usuário.

Supondo que Bob pode se conectar com a Internet por meio de roteadores em momentos distintos, como na escola e no trabalho, diferentes endereços IP são atribuídos à aplicação

SIP de forma a impossibilitar a localização. Por este motivo é enviada uma mensagem REGISTER para se registrar e permitir que Bob seja encontrado (KUROSE; ROSS, 2014).

Proxy SIP: Se o usuário Max quer conversar com Bob, é enviada uma mensagem de convite contendo o identificador bob@domain.com. Essa requisição é encaminhada para o destino através do servidor proxy, o qual realiza uma consulta do endereço de IP atual do destinatário no servidor de localização, enviando a mensagem ao local correto. Portanto, o servidor proxy é responsável pela troca de mensagens SIP entre os agentes (TANENBAUM; WETHERALL, 2011).

Figura 4 – Troca de mensagens entre os agentes que utilizam o protocolo SIP



Fonte: (KUROSE; ROSS, 2014).

A Figura 4 mostra a troca de mensagens entre dois usuários realizadas com o auxílio dos Servidores de Localização e Proxy. Esse processo é explicado no Quadro 1

2.4 Ameaças contra os recursos computacionais

Esta seção irá demonstrar como os ataques DoS e DDoS podem causar prejuízos financeiros provenientes das sobrecargas de processamento e largura de banda dos servidores.

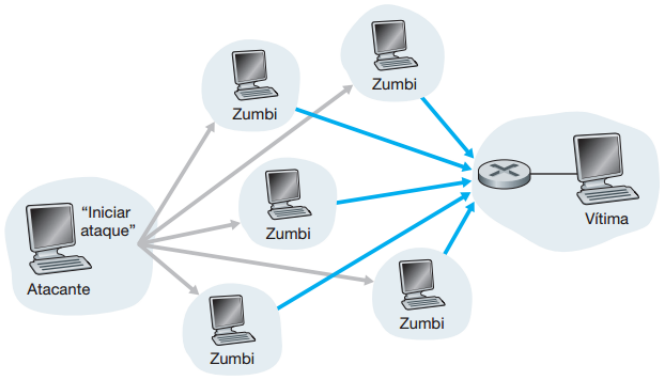
A Internet fornece diversos recursos importantes para o funcionamento das aplicações, como por exemplo, a transferência de arquivos entre usuários e mensagens de e-mail. No entanto, arquivos recebidos da Internet podem infectar o dispositivo do usuário com um conteúdo malicioso conhecido como *malware*. Um dispositivo contaminado pode executar ações sem o consentimento da vítima, sendo capaz de excluir fotos, acessar sites da Internet e realizar ataques de negação de serviço. Os usuários infectados por um *malware* podem fazer parte de uma *botnet*, que é uma grande rede de dispositivos controlados por pessoas mal intencionadas (KUROSE; ROSS, 2014). Na Figura 5 é demonstrado um ataque DDoS de uma *botnet*, onde o usuário infectado é nomeado de *zumbi* devido ao fato de participar de ataques sem ter consentimento.

Quadro 1 – Redirecionamento SIP

Nome	Função
1 INVITE	O *Chamador* envia uma mensagem de convite ao servidor proxy com a intenção de iniciar uma sessão.
2 LOOKUP	O servidor proxy pesquisa o endereço de IP do usuário de destino no servidor de localização.
3 REPLY	O endereço de IP atual é enviado para o servidor proxy.
4 INVITE	O servidor proxy encaminha ao agente *Chamado*, que é o usuário de destino, uma solicitação de início de sessão enviada previamente pelo usuário *Chamador*.
5 OK	O agente *Chamado* aceita o convite e responde o servidor proxy.
6 OK	O servidor proxy encaminha a resposta ao usuário *Chamador*.
7 ACK	O agente *Chamador* envia uma mensagem ao servidor proxy confirmando o início da sessão.
8 ACK	Encaminha a mensagem que confirma o início da sessão ao agente *Chamado*.
9 Dados	A sessão é iniciada e os usuários podem trocar informações diretamente.

Fonte: Elaborado pelo próprio autor.

Figura 5 – Ataque distribuído de negação de serviço (DDoS)



Fonte: (KUROSE; ROSS, 2014).

Os ataques de DoS e DDoS são muito comuns na Internet, onde milhares ocorrem anualmente. Eles podem utilizar toda a largura de banda da rede e elevar a carga de processamento do servidor de forma a tornar o serviço inutilizável por usuários legítimos (KUROSE; ROSS, 2014). Nesse contexto, o ataque é classificado como uma ameaça de interrupção (TANENBAUM; STEEN, 2008).

3 Trabalhos Relacionados

Este capítulo irá abordar trabalhos que analisam as vulnerabilidades do protocolo SIP, desenvolvem métodos para diminuir os impactos causados por ataques DoS e DDoS e aplicam técnicas a fim de descobrir usuários mal intencionados em um serviço SIP.

O trabalho de Stanek e Kencl (2012) propôs uma arquitetura composta de três estágios responsáveis por filtrar as requisições de mensagens.

- No primeiro, o servidor de redirecionamento utilizou uma função para atribuir o endereço de IP e porta do próximo estágio à mensagem, enviando-a para o agente que fez a requisição.

- O segundo estágio possui um Firewall que verifica se o número de requisições do agente em certo intervalo de tempo é maior que o limite definido ε_f, em caso positivo a mensagem é descartada.

- O terceiro estágio usou uma função para verificar se o endereço de IP de origem foi encaminhado para o servidor SIP correto, descartando a mensagem se divergir. Logo é verificado o limite ε_n de requisições permitida por intervalo de tempo ao servidor SIP, onde a mensagem é ignorada se for ultrapassado.

O servidor SIP foi implementado em um quadcore Xeon 2,5 GigaHertz (GHz) com 8 GigaByte (GB) de memória RAM e o limite ε_f foi configurado para permitir no máximo 50 requisições em 5 segundos.

O tráfego legítimo foi simulado com 150 requisições por segundo divididas entre 50 dispositivos virtuais. Dessa maneira, a carga de processamento do servidor aumentou de 50% para aproximadamente 85% no instante que o ataque foi iniciado, o qual utilizou 100 origens para enviar 1000 requisições por segundo em um intervalo de 100 segundos. No decorrer de 5 segundos após o início do ataque, o Firewall do segundo estágio bloqueou as requisições dos endereços IP que ultrapassaram o limite ε_f.

Devido ao SIP tentar reenviar as mensagens referentes às requisições onde a resposta não foi obtida, o processamento do servidor começou a diminuir após 10 segundos, voltando para os valores iniciais cerca de 90 segundos depois do início do ataque.

No início do segundo teste o número de origens subiu para 125 e as outras configura-

ções foram mantidas. À vista disso o ataque chegou ao terceiro estágio, pois o número de requisições foi limitado em 8 para cada origem, implicando que o limite ε_f de 50 requisições não fosse rompido, uma vez que o número de requisições foi igual a 40 no tempo de 5 segundos. Por consequência, a carga de processamento do servidor aumentou em média para 85% nos primeiros 20 segundos, atingindo valores próximos de 100% no tempo restante do ataque que durou 80 segundos. Isso foi suficiente para prejudicar o tráfego legítimo da rede.

O estudo de Seo, Lee e Nuwere (2013) apresentou um sistema que utiliza regras de estado com hierarquia. Com o objetivo detectar ataques DoS, foi criado um conjunto de mensagens para cada estado do sistema, podendo caracterizar irregularidades quando a resposta esperada pelo servidor SIP não é recebida. O limite de pps (pacotes por segundo) para cada estado foi definido de acordo com a transmissão legítima com acréscimo de 2 pps para não prejudicar o funcionamento normal do serviço. Como resultado dos testes, o sistema detectou todos os ataques que utilizaram 35 pps, 10 pps, 5 pps e 3 pps.

No trabalho desenvolvido por Tas, Ugurdogan e Baktir (2016), foi implementado em laboratório um simulador de ataques DoS e DDoS de forma a explorar as vulnerabilidades do mecanismo de retransmissão SIP. Foram formulados ataques para burlar listas de bloqueio e limitações de taxas baseadas em IP, pacotes e consumo de banda. Essas regras foram ignoradas com a técnica conhecida como IP spoofing, a qual modifica o endereço IP real dos pacotes enviados. Adicionalmente, a lista de usuários legítimos do sistema foi salva em arquivo para modificar os valores dos campos To, User-Agent e From do cabeçalho das mensagens SIP. De forma complementar, os campos Contact e Via foram gerados através do IP spoofing.

Visando detectar requisições que infrinjam o padrão de comportamento da rede, os limites Atual, Normal, Suspeito e Ataque foram calculados baseando-se na utilização habitual do servidor SIP. Se o endereço IP do suposto invasor for bloqueado, uma mensagem é enviada para o IP em questão com a finalidade de receber uma resposta, se for recebida o endereço IP é desbloqueado. Durante os testes, o mecanismo de defesa reduziu a carga de processamento do servidor SIP para 13,6%, que sob ataque era 87%.

Em Semerci, Cemgil e Sankur (2018), foi proposto um sistema composto por um módulo Monitor com o objetivo de detectar os ataques DDoS e um módulo Identificador a fim de encontrar usuários utilizadores do sistema que contenham intenções maliciosas. Para aplicar essas ideias, foram realizados cálculos estatísticos com o intuito de analisar e comparar padrões de comportamento, buscando encontrar similaridades entre uma amostra conhecida com um dado desconhecido. Nesse contexto, foram feitas avaliações do tempo entre a transmissão das mensagens SIP, duração das chamadas e a probabilidade de

realizar chamadas em determinado instante.

Nos experimentos realizados, foram usadas máquinas virtuais para representarem o servidor SIP, usuários legítimos e usuários maliciosos. Considerando todo o tráfego malicioso encontrado em um caso de teste, foi detectado que em média 79% dos ataques foram capturados de forma correta, mostrando que 21% dos alarmes foram falso-positivos.

4 Materiais e Métodos

Este capítulo relata os materiais e processos utilizados durante o desenvolvimento deste trabalho. Durante os experimentos, foi aplicada a configuração da Tabela 1 contendo intervalos de tempo similares aos encontrados no trabalho de Stanek e Kencl (2012).

Tabela 1 – Cenário do teste

Configuração	Tráfego legítimo	DoS
Duração (segundos)	200	160
Início (segundos)	0	40
Finalizado (segundos)	200	200
Quantidade de dispositivos	1	1
rps por dispositivo	5	200

Fonte: Elaborado pelo próprio autor.

Todas as ferramentas utilizadas para o desenvolvimento da proposta, testes e análise dos resultados são descritas nos seguintes itens:

- **Firewall pfSense:** Possui recursos para proteger os dispositivos da rede de invasores, como por exemplo, filtro de pacotes e monitoramento da largura de banda pfSense (2017).

- **Snort:** É uma ferramenta IDS e IPS usada para detectar anomalias na rede baseando-se em assinaturas previamente definidas compostas por um conjunto de regras que identificam atividades de invasão. Cada pacote que passa pela rede é comparado com as assinaturas da base de dados, se o tráfego for classificado como suspeito, é emitido um alerta que pode ser registrado em um documento de texto para ser analisado pelo administrador da rede. Portanto, para detectar ataques DoS e DDoS, é necessário ter uma ou várias assinaturas em uma base de dados que caracterizem estes ataques (KUROSE; ROSS, 2014).

- **SIPp:** Pode simular milhares de usuários enviando mensagens para o servidor SIP. Por meio dessas requisições, são gerados dados estatísticos para medir o desempenho de uma rede que utiliza o protocolo SIP, como por exemplo, informações do tempo de resposta do servidor GAYRAUD et al. (2017).

- **Kamailio:** É um Servidor SIP que implementa serviços de proxy e registro. Logo pode adicionar novos usuários ao sistema, obter o endereço IP atual do agente e iniciar sessões através da troca de mensagens SIP Kamailio (2017).

- **DoSMonitor:** Aplicação desenvolvida nesse trabalho, sendo responsável por analisar os logs emitidos pelo Snort e adicionar regras de bloqueio ao Firewall pfSense Andrade (2019a).

- **SystemMonitor:** Aplicação desenvolvida em linguagem Python para monitorar e salvar em arquivo de log informações do consumo da largura de banda, memória e CPU em tempo real Andrade (2019b).

- **sngrep:** Ferramenta para mostrar em tempo real o fluxo de mensagens SIP de um terminal (QUEROL, 2016).

O Snort foi responsável por monitorar o tráfego da rede, escrevendo os endereços IP de origem e destino junto com a descrição do alerta em um arquivo de texto quando um invasor é detectado. Esse arquivo contém informações que foram interpretadas pelo DoSMonitor que gerenciou as regras de segurança do pfSense por meio de modificações no arquivo config.xml do Firewall. A quantidade de mensagens trocadas entre o servidor Kamailio e o emissor dos ataques foi monitorada pelo sngrep, o qual mostrou que a quantidade de mensagens processadas implica diretamente no consumo da largura de banda e processamento com o auxilio da ferramenta SystemMonitor.

As políticas de segurança de um sistema devem ser descritas para especificar as ações que os usuários e serviços podem exercer (TANENBAUM; STEEN, 2008). Com o intuito de atender esses propósitos, a seguinte regra foi adicionada nas configurações do Snort para emitir um alerta quando algum usuário enviar um número maior ou igual a 150 mensagens no intervalo de 1 segundo:

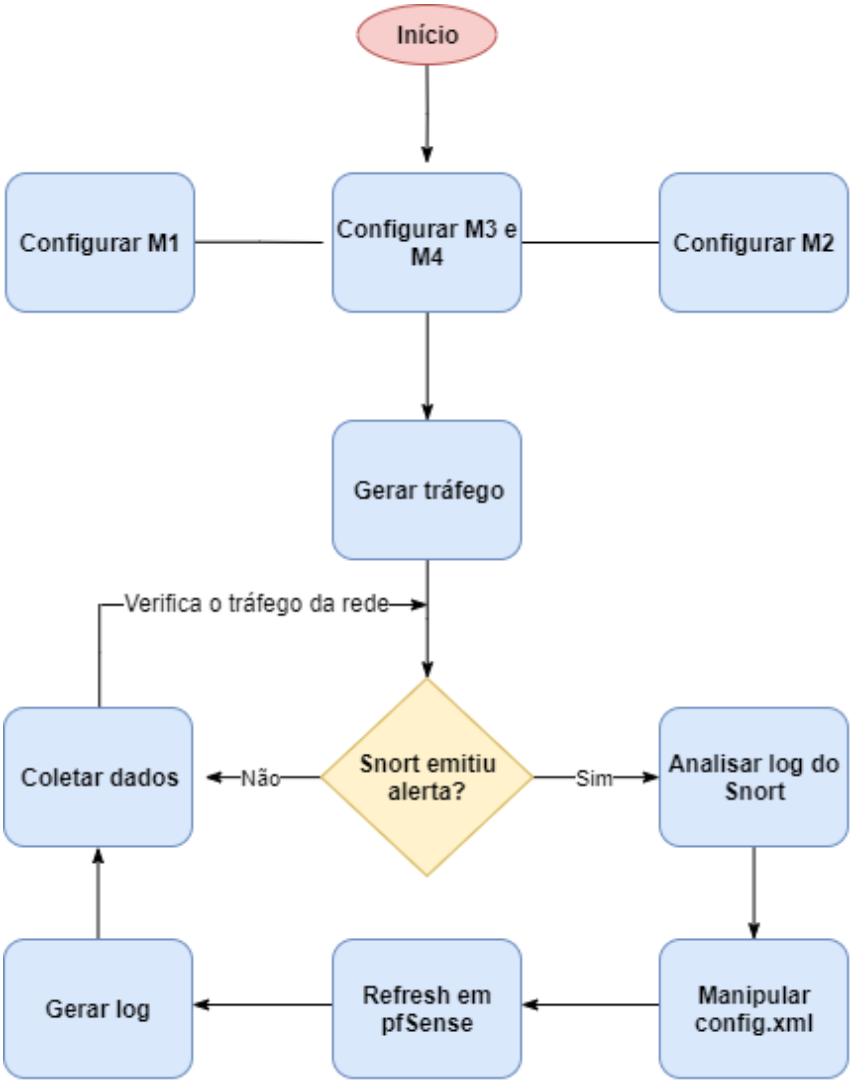
alert udp any any -> 192.168.100.254 5060 (msg:"DoS"; threshold: type both, track by_dst, count 150, seconds 1, sid:321; rev:3;)

Quando houve consumo excessivo de requisições na rede, o Snort emitiu um alerta contendo os IPs de origem e destino do ataque, salvando essa informação no arquivo /var/log/snort/alert.log conforme o trecho em destaque abaixo.

05/27-02:30:42.070886 [**] [1:321:3] DoS [**] [Priority: 0] UDP 192.168.100.103:5060 -> 192.168.100.254:5060

No instante em que o DoSMonitor identificou a alteração em alert.log, o arquivo /cf/conf/config.xml do pfSense foi atualizado com uma nova regra para bloquear o endereço de IP do emissor da mensagem. A atualização foi concluída após apagar o arquivo /tmp/config.cache e executar os comandos */etc/rc.filter_configure_sync* e *pfctl -F state* na máquina virtual do pfSense. A Figura 6 mostra como esse processo foi executado.

Figura 6 – Fluxo de comportamento do sistema



Fonte: Elaborado pelo próprio autor.

Os blocos encontrados na Figura 6 representam um algoritmo de fluxo das ações executadas durante os testes, onde cada item exerce funções conforme mostra o Quadro 2.

Para poupar recursos o modo gráfico das máquinas M1, M2, M3 e M4 foi desativado, ao contrário de M5 que foi usada para acessar a interface de configuração web do pfSense e executar comandos via ssh. O dispositivo físico utilizado foi um Acer E1-572-6_BR648, possuindo 6GB de memória RAM DDR3L SDRAM com frequência 1600Mhz, SSD Plus SanDisk modelo G26 de 480GB, processador Intel core i5-4200U com frequência de 1.6GHz

Quadro 2 – Definição dos blocos do algoritmo de fluxo de dados

Nome do bloco	**Representação**
Início	Início do algoritmo
Configurar M1	Instalar em Maquina 1: Sistema operacional Linux, SystemMonitor e Kamailio
Configurar M2	Instalar em Máquina 2: FreeBSD pfSense, DoSMonitor e Snort.
Configurar M3 e M4	Instalar em Máquina 3 e Máquina 4: Sistema operacional Linux e SIPp.
Gerar tráfego	Simula o tráfego de rede com a ferramenta SIPp.
Snort emitiu alerta?	Verifica se o Snort emitiu algum alerta.
Coletar dados	Obtém informações da largura de banda, processamento e memória de M1 .
Ler log do Snort	Analisa as informações contidas no log do Snort para gerenciar as regras do firewall pfSense
Manipular config.xml	Atualiza as regras do firewall pfSense através do arquivo config.xml
Refresh em pfSense	Sincroniza o pfSense com as novas regras do arquivo config.xml
Gerar log	Exibe no console as alterações feitas nas regras do pfSense

Fonte: Elaborado pelo próprio autor.

e capacidade máxima 2.6GHz, placa de vídeo Intel HD Graphics 4400. Os recursos usados nos experimentos são expostos no Quadro 3.

As máquinas virtuais foram configuradas em um ambiente de rede semelhante ao ilustrado na Figura 7. A decisão de deixar todos os dispositivos na rede gerenciada pelo pfSense foi aderida para permitir controlar todos os pacotes que circulam na rede, facilitando o bloqueio de endereços IP de usuários mal intencionados e o redirecionamento de pacotes.

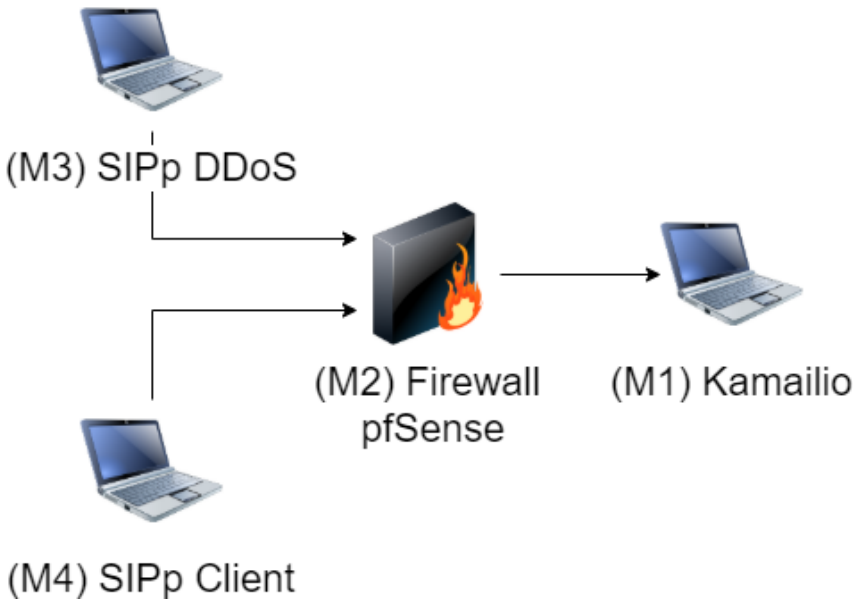
Uma regra de encaminhamento de pacotes foi adicionada ao pfSense com a finalidade de transmitir ao Kamailio todas as mensagens SIP recebidas. Dessa forma, as requisições enviadas das máquinas virtuais M3 e M4 foram direcionadas para M2, que redirecionou as mensagens para M1. O Quadro 4 mostra os endereços IPs das máquinas envolvidas no experimento e o Quadro 5 expõe o nome e a versão das tecnologias usadas neste trabalho.

Quadro 3 – Recursos utilizados nos experimentos

Máquina virtual	**Aplicações instaladas**	**Memória RAM (MB)**	**Armazenamento (GB)**
M1	Linux Mint, SystemMonitor, sngrep e Kamailio	922	17,96
M2	FreeBSD pfSense, Snort e DoSMonitor	512	15,00
M3	Linux Mint e SIPp	808	18,90
M4	Linux Mint e SIPp	808	18,90
M5	Linux Mint	808	18,90

Fonte: Elaborado pelo próprio autor.

Figura 7 – Representação do ambiente de rede do experimento



Fonte: Elaborado pelo próprio autor.

Quadro 4 – Endereços IPs usados pelas máquinas da rede

Máquina virtual	**Descrição**	**Endereço IP**
M1	Kamailio	192.168.100.100
M2	Firewall pfSense	192.168.100.254
M3	SIPp DoS	192.168.100.103
M4	SIPp Client	192.168.100.102
M5	Linux Mint	192.168.100.101

Fonte: Elaborado pelo próprio autor.

Quadro 5 – Aplicações utilizadas nos testes

Nome do recurso	**Versão**
Oracle VM VirtualBox	6.0.10
Linux Mint	19.1 (Tessa)
Kamailio	5.2.2
SIPp	3.5.2
FreeBSD pfSense	2.4.4
Snort	2.9.12
Python	2.7.16
SystemMonitor	1.0
sngrep	1.4.5

Fonte: Elaborado pelo próprio autor.

Os testes foram iniciados após os seguintes procedimentos:

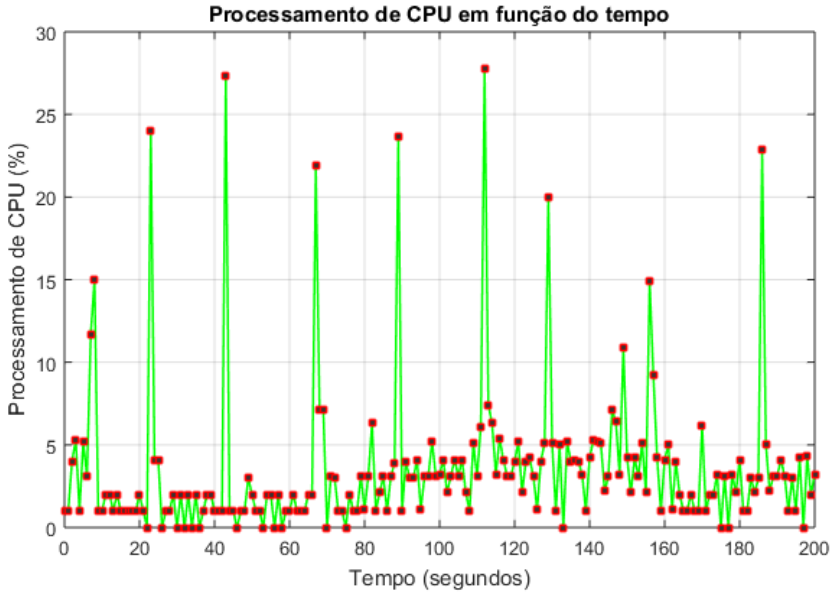
- Executar Snort em M2.
- Iniciar o DosMonitor em M2 executando o arquivo Main.py.
- Executar SIPp em M4 para simular o tráfego normal da rede direcionando as requisições para M2.
- Executar SystemMonitor em M1 para obter informações dos consumos de CPU, memória e largura de banda.
- Executar SIPp em M3 direcionando o ataque DoS para M2.

5 Resultados

Este capítulo apresenta os resultados obtidos por meio da implantação do DosMonitor que bloqueou os ataques DoS, reduzindo o consumo de memória, processamento e largura de banda do servidor Kamailio.

A seguir, os resultados são apresentados partindo do ponto onde as taxas de requisições por segundo são iguais a 0, 5 e 200. Os gráficos foram gerados com a ferramenta Matlab na versão R2015a e as informações dos gráficos foram obtidas através da aplicação SystemMonitor. A Figura 8 mostra a taxa de processamento de CPU durante 200 segundos quando o valor de rps é igual a 0. Nesse caso, o consumo máximo de CPU foi de 27,80%, o mínimo de 0,00%, tendo o valor médio de 3,54%.

Figura 8 – Processamento de CPU com rps igual a 0



Fonte: Elaborado pelo próprio autor.

Para analisar as diferenças nos consumos dos recursos computacionais do servidor Kamailio quando o valor de rps é alterado, foi registrado na Tabela 2 os consumos médio, mínimo e máximo de memória, CPU e largura de banda de entrada e saída.

De forma análoga, a Figura 9 mostra o consumo de CPU em 200 segundos quando o número de rps é igual a 5.

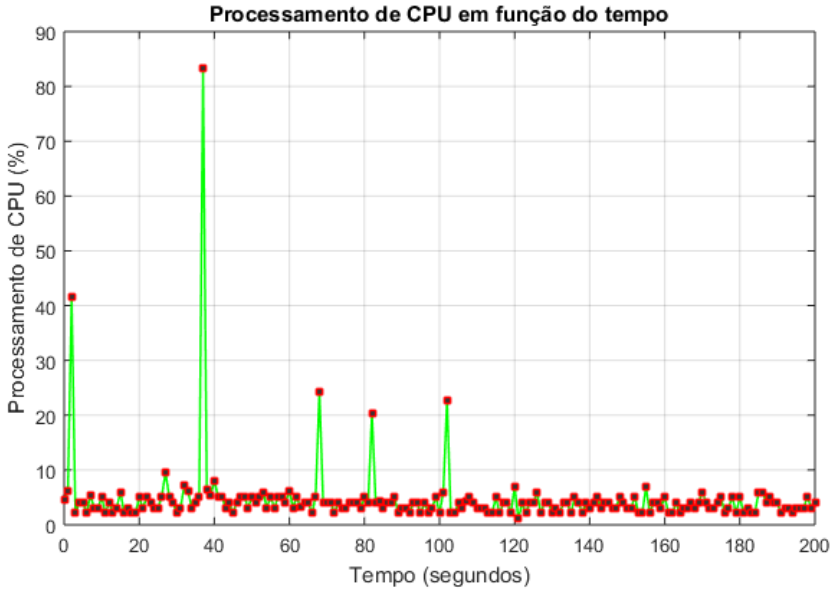
Representando o fluxo normal da rede, a Tabela 3 mostra o consumo de recursos quando o o valor de rps foi igual a 5. O processamento médio de CPU se manteve próximo do valor

Tabela 2 – Consumo de recursos com rps igual a 0

Descrição	CPU (%)	Memória (MB)	Entrada (KB)	Saída (KB)
Valor médio	3,54	283,77	0,01	0,02
Consumo máximo	27,80	284,00	1,00	1,00
Consumo mínimo	0,00	283,00	0,00	0,00

Fonte: Elaborado pelo próprio autor.

Figura 9 – Processamento de CPU com rps igual a 5



Fonte: Elaborado pelo próprio autor.

registrado na Tabela 2 dado que a diferença foi de 1,12%. A média de utilização de memória foi praticamente a mesma, ao contrário da largura de banda que utilizou em média 16,32 KB de entrada e 11,24 KB para a saída.

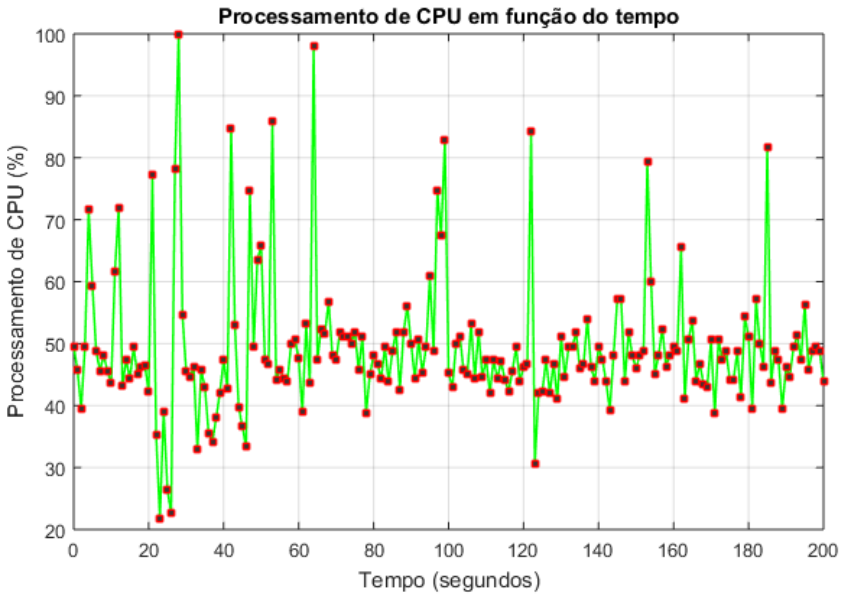
Tabela 3 – Consumo de recursos com rps igual a 5

Descrição	CPU (%)	Memória (MB)	Entrada (KB)	Saída (KB)
Valor médio	4,66	282,37	16,32	11,24
Consumo máximo	83,30	283,00	18,00	13,00
Consumo mínimo	1,10	282,00	14,00	10,00

Fonte: Elaborado pelo próprio autor.

A Figura 10 mostra o processamento de CPU no estado onde o sistema está sob ataque DoS com uma taxa de rps igual a 200.

Figura 10 – Processamento de CPU com rps igual a 200



Fonte: Elaborado pelo próprio autor.

Comparando os dados da Tabela 4 com a Tabela 3, existe uma diferença de 44,68% no processamento médio de CPU, visto que houve a mudança no valor médio de 4,66% para 49,34%.

Tabela 4 – Consumo de recursos com rps igual a 200

Descrição	**CPU (%)**	**Memória (MB)**	**Entrada (KB)**	**Saída (KB)**
Valor médio	49,34	282,38	369,70	259,72
Consumo máximo	100,00	284,00	1154,00	445,00
Consumo mínimo	21,70	281,00	99,00	72,00

Fonte: Elaborado pelo próprio autor.

De acordo com os dados contidos nas tabelas desse capítulo, com o tráfego normal da rede a taxa de utilização de CPU do servidor Kamailio alternou entre 1,10% e 83,3%, tendo como média o valor de 4,66%. O consumo médio de memória RAM foi de 282,37 MB, tendo como valores mínimo e máximo 282 MB e 283 MB, respectivamente. Para a largura de banda, foram enviados em média 259,72 KB/s (KiloBytes por segundo) e recebidos em média 369,80 KB/s.

As informações do consumo de memória, CPU e largura de banda em ambientes que não utilizaram o sistema desse trabalho foram coletadas para serem comparadas com os resultados dos testes que utilizaram o sistema de defesa proposto.

Logo abaixo é demonstrado o código usado para simular o tráfego normal da rede com o valor de rps igual a 5.

./sipp -r 5 -rp 1000 -l 0 -d 0 -p 5060 -rsa 192.168.100.100:5060 192.168.100.103:5060 -sf scenario.xml

O Quadro 6 construído com o auxílio da documentação encontrada em GAYRAUD et al. (2017) mostra qual é o recurso proveniente de cada opção adicionada ao comando da aplicação SIPp utilizada para simular o tráfego da rede.

Quadro 6 – Comandos SIPp

Comando	**Descrição**
-r	Número de requisições.
-rp	Período de repetição das requisições.
-l	Número máximo de chamadas simultâneas.
-d	Adiciona uma pausa na execução do cenário.
-p	Configura a porta local utilizada pelo SIPp.
-rsa	Informa o IP e porta do destino e origem da requisição, respectivamente.
-sf	Define o cenário do teste

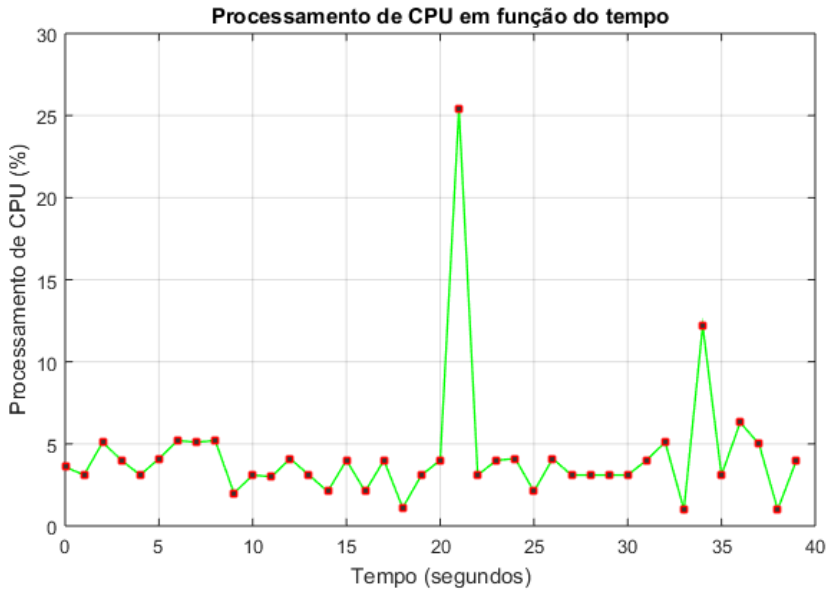
Fonte: Elaborado pelo próprio autor.

Durante os primeiros 39s (segundos) foi simulado o tráfego normal da rede com uma taxa de rps igual a 5. A Figura 11 mostra o gráfico do processamento de CPU durante esse tempo, onde é notado um comportamento de consumo com valores baixos.

Conforme mostra a Tabela 5, os valores médios registrados nesse intervalo são parecidos com os da Tabela 3 pois até esse momento o ataque não havia sido iniciado.

A simulação do ataque DoS foi iniciada aos 40s, permanecendo em execução junto ao tráfego normal da rede até os 200s. De acordo com a Figura 12 há um aumento no consumo de processamento após o ataque ter sido iniciado. Para essa parte, o comando SIPp foi executado como é mostrado abaixo utilizando um valor de rps igual a 200.

Figura 11 – Processamento de CPU nos primeiros 39s da simulação



Fonte: Elaborado pelo próprio autor.

Tabela 5 – Consumo de recursos durante os primeiros 39s da simulação

Descrição	CPU (%)	Memória (MB)	Entrada (KB)	Saída (KB)
Valor médio	4,28	285,68	16,23	11,25
Consumo máximo	25,40	286,00	17,00	12,00
Consumo mínimo	1,00	285,00	15,00	10,00

Fonte: Elaborado pelo próprio autor.

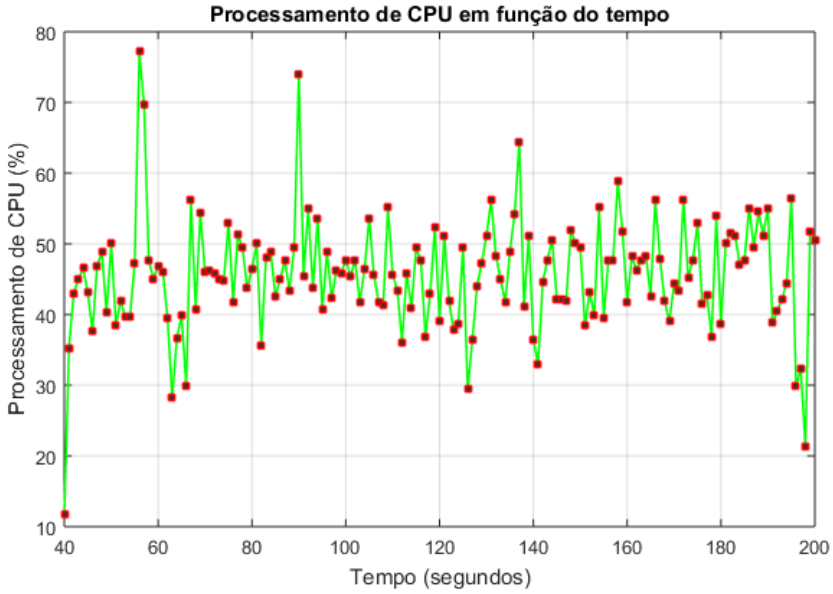
./sipp -r 200 -rp 1000 -l 0 -d 0 -p 5060 -rsa 192.168.100.254:5060 192.168.100.102:5060 -sf scenario.xml

Há um aumento notável no consumo de largura de banda e CPU no intervalo de 40s até 200s, mantendo uma média de processamento de CPU igual a 45,68% até o fim da simulação. A Tabela 6 mostra o consumo dos recursos usados para esse período.

Foi observado pela Figura 13 que o processamento de CPU aumenta consideravelmente a partir do instante que o ataque foi iniciado. Como nesse momento o sistema deste trabalho não estava em execução, não houve uma diminuição nos consumos registrados.

O processamento médio de CPU durante o a simulação foi de 37,44%, tendo um valor menor que o registrado entre 40s e 200s devido ao consumo médio de CPU nos primeiros 39s ser igual a 4,28%. A Tabela 7 mostra os valores mínimos, máximos e médios dos

Figura 12 – Processamento de CPU no intervalo de 40s até 200s



Fonte: Elaborado pelo próprio autor.

Tabela 6 – Consumo de recursos no intervalo de 40s até 200s

Descrição	CPU (%)	Memória (MB)	Entrada (KB)	Saída (KB)
Valor médio	45,68	283,46	295,52	212,69
Consumo máximo	77,2	285	573	329
Consumo mínimo	11,7	280	61	41

Fonte: Elaborado pelo próprio autor.

recursos que estão sendo monitorados.

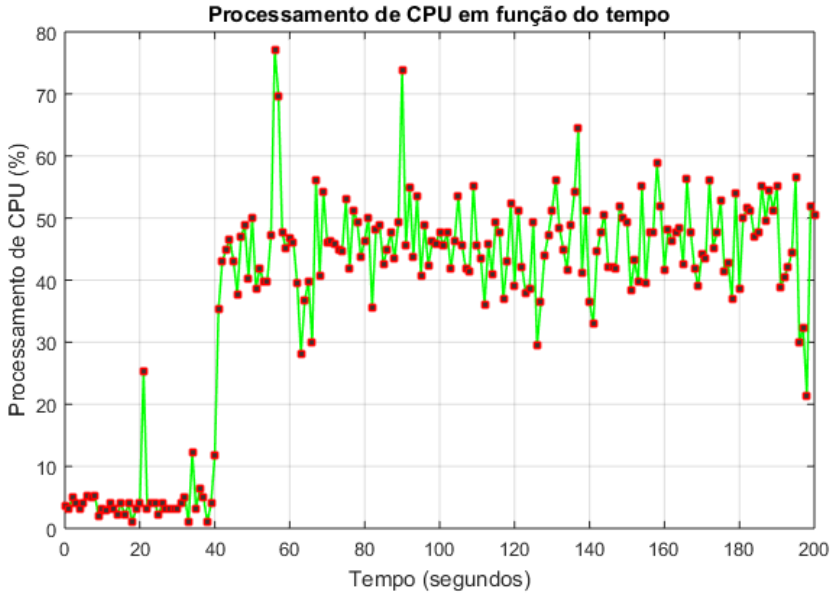
Tabela 7 – Consumo de recursos no intervalo de 0s até 200s

Descrição	CPU (%)	Memória (MB)	Entrada (KB)	Saída (KB)
Valor médio	37,44	283,90	239,94	172,60
Consumo máximo	77,20	286,00	573,00	329,00
Consumo mínimo	1,00	280,00	15,00	10,00

Fonte: Elaborado pelo próprio autor.

Cada requisição enviada implicou em uma resposta do servidor Kamailio, o que pode ser observado na Figura 14. Os recursos de processamento, memória e largura de banda

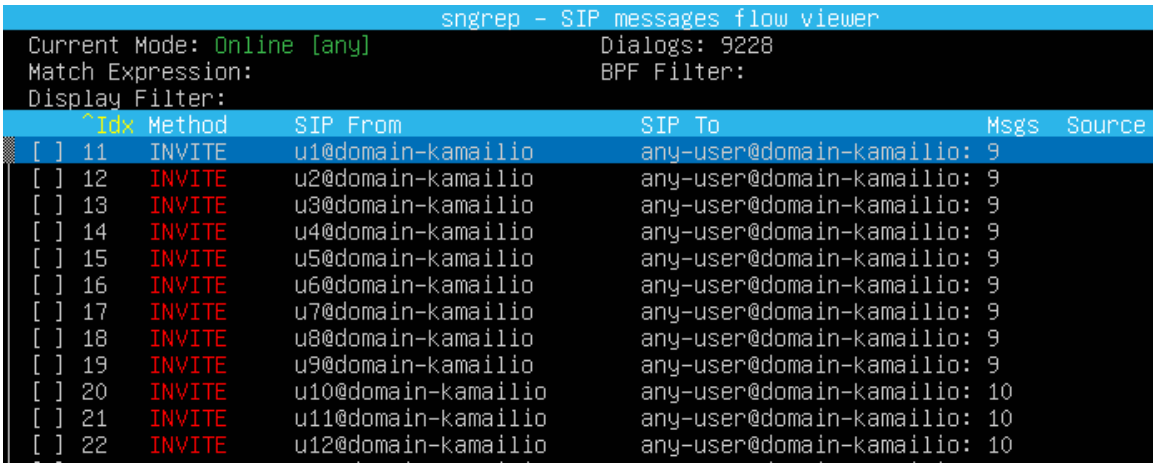
Figura 13 – Processamento de CPU no intervalo de 0s até 200s



Fonte: Elaborado pelo próprio autor.

disponíveis no servidor alvo foram elevados devido ao grande número de mensagens que foram analisadas e enviadas para o emissor do ataque em um curto período de tempo, dado que 200 mensagens por segundo foram disparadas.

Figura 14 – Mensagens SIP registradas com o sngrep



Fonte: Elaborado pelo próprio autor.

Repetindo a simulaçao com o sistema proposto em execução, o processamento do servidor Kamailio voltou para os níveis normais no instante que as regras do Firewall foram atualizadas. O fluxo desse processo foi realizado da seguinte forma:

- **Tráfego normal da rede iniciado:** Foram enviadas mensagens para o Kamailio através da ferramenta SIPp no intervalo de 0s até 200s.

- **Ataque iniciado:** Foram enviadas mensagens para o servidor SIP através da ferramenta SIPp no intervalo de 40s até 200s.

- **Snort:** Após detectar o ataque, escreveu no arquivo alert.log os endereços IP de origem e destino e uma breve descrição do ataque.

- **DoSMonitor:** Interpretou o alerta emitido pelo Snort, atualizando as regras do Firewall pfSense para bloquear os endereço IP do emissor do ataque.

- **pfSense** Bloqueou os pacotes que continham o IP que originou o ataque de forma que essas requisições não chegassem ao servidor Kamailio.

Conforme mostra a Tabela 8, o consumo de memória permaneceu constante, já os consumos médios de CPU e largura de banda foram reduzidos quando comparados com os dados da Tabela 7. Nesse caso, as diferenças médias de CPU foi de 32,08%, largura de banda de entrada de 221,03 KB e largura de banda de saída foi de 158,93 KB.

Tabela 8 – Teste do sistema DoSMonitor

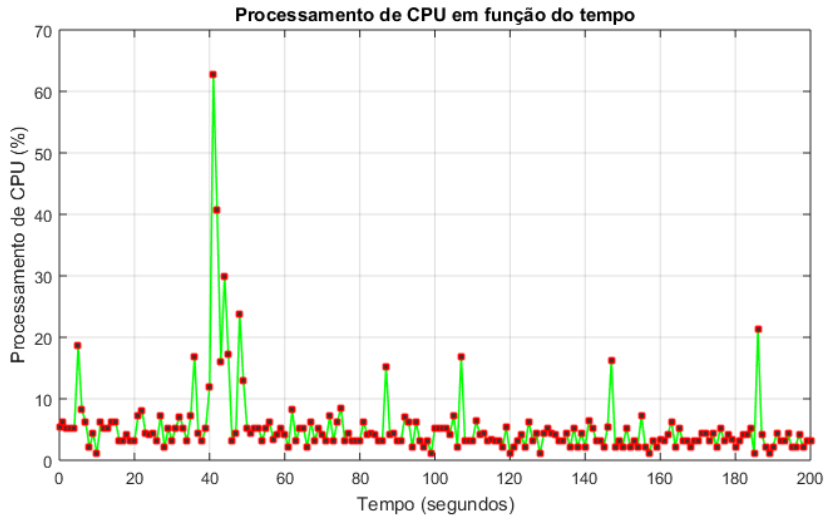
Descrição	**CPU (%)**	**Memória (MB)**	**Entrada (KB)**	**Saída (KB)**
Valor médio	5,36	274,00	18,91	13,67
Consumo máximo	62,70	274,00	178,00	129,00
Consumo mínimo	1,10	274,00	13,00	10,00

Fonte: Elaborado pelo próprio autor.

A Figura 15 mostra o consumo de CPU com o DoSMonitor em execução, onde é possível observar que em pouco tempo a ação executada pelo sistema proposto conseguiu diminuir o consumo de CPU do servidor Kamailio.

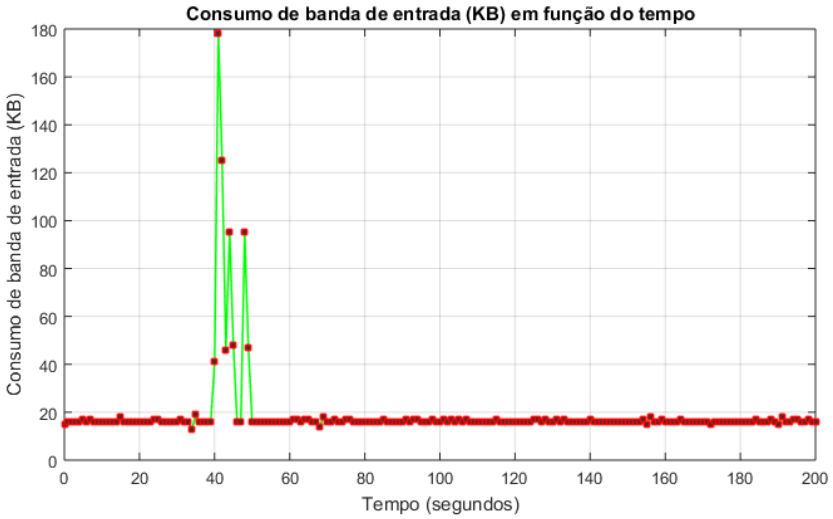
A Figura 16 e a Figura 17 mostram os consumos de largura de banda de entrada e saída no servidor Kamailio, onde é notado um aumento de consumo iniciado aos 40s, voltando aos níveis visuais registrados antes dos 40s após 10s.

Figura 15 – Processamento de CPU com o sistema DoSMonitor



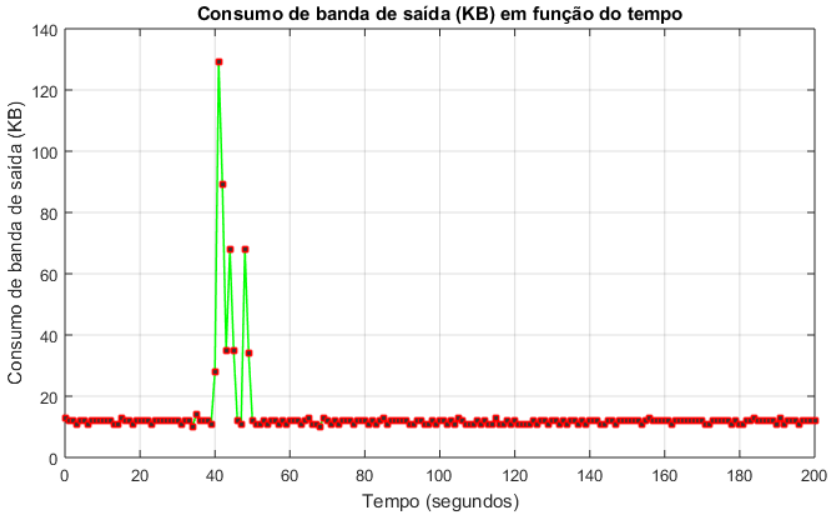
Fonte: Elaborado pelo próprio autor.

Figura 16 – Largura de banda de entrada com o sistema DoSMonitor



Fonte: Elaborado pelo próprio autor.

Figura 17 – Largura de banda de saída com o sistema DoSMonitor



Fonte: Elaborado pelo próprio autor.

6 Considerações Finais

O protocolo SIP tem um papel muito importante na negociação das chamadas VoIP devido a responsabilidade de descobrir o endereço IP do destinatário, iniciar chamadas e encerrar ligações por meio da troca de mensagens entre os usuários. Se uma mensagem não chegar ao destino o serviço poderá ser comprometido, implicando em prejuízos financeiros. Isso pode acontecer quando um grande número de requisições é enviada para o servidor SIP em um curto período de tempo, pois dessa maneira os recursos de processamento de CPU e largura de banda ficarão sobrecarregados.

Visto que os experimentos foram executados em máquinas virtuais, foi necessário que o ataque DoS partisse da mesma rede em que se encontrava o servidor. Todas as requisições SIP foram enviadas para o Firewall pfSense, o qual continha uma regra que redirecionava todo o tráfego SIP para o Kamailio. Então, pelo motivo de que não foi encontrada uma maneira de realizar um ataque ao ambiente virtual partindo de uma rede externa, tanto o servidor SIP, Firewall pfSense, máquina DDoS e a máquina que simula o tráfego normal da rede foram configuradas na mesma rede, contendo os endereços de IP 192.168.100.100, 192.168.100.254, 192.168.100.103 e 192.168.100.102, respectivamente.

Os tópicos abaixo mostram os resultados do experimento quando o servidor SIP esteve submetido a um ataque DoS, onde o consumo médio dos recursos computacionais foi reduzido em comparação com os valores registrados antes da utilização do sistema DoSMonitor.

- **CPU:** Redução no processamento de 37,44% para 5,36%.
- **Largura de banda de entrada:** Diminuição do volume de dados recebidos de 239,94 KB para 18,91 KB.
- **Largura de banda de saída:** Redução do volume de dados enviados de 172,60 KB para 13,67 KB.

Apesar disso, não ocorreram mudanças no consumo de memória RAM em todos os testes realizados, incluindo os que não utilizaram o DoSMonitor. Portanto, com o ambiente de rede, máquinas virtuais e aplicações usadas nesse trabalho o recurso de memória RAM não está apto a comparações, pois não ocorreram alterações antes e após os ataques.

Considerando os testes realizados, o sistema desenvolvido mostrou ser eficiente trabalhando em conjunto com o Snort e pfSense utilizando regras de bloqueio baseadas em endereço IP. Quando o limite de 150 requisições por segundo foi atingindo ou ultrapassado, o endereço IP

que originou a requisição foi bloqueado pelo Firewall, impedindo que a mensagem enviada de forma maliciosa chegasse ao servidor SIP de destino.

Equiparando os resultados com trabalhos similares, foi notada a diferença entre o tempo que o servidor SIP demorou para voltar aos valores iniciais de processamento, dado que neste trabalho esses níveis foram restabelecidos em 10 segundos após as regras do Firewall terem sido atualizadas. No trabalho de Stanek e Kencl (2012), o tempo necessário para voltar aos níveis iniciais foi de 90 segundos. Isso ocorreu devido a uma lista real de usuários registrados no servidor SIP ter sido usada, resultando em novas tentativas de comunicação por parte do servidor.

Como trabalhos futuros, os testes realizados em máquinas virtuais podem ser remanejados para ambientes físicos. Dessa maneira, o Kamailio, pfSense e SIPp devem ser configurados em máquinas e redes diferentes de modo a aproximar da realidade das aplicações VoIP. Um outro trabalho possível seria a implantação de uma outra técnica que tenha o propósito de impedir que ataques DoS cheguem aos servidores SIP, dessa maneira os resultados obtidos poderão ser comparados.

Referências

ANDRADE, M. **DoSMonitor**. 2019. Disponível em: <https://github.com/marcusaurelioo/DosMonitor>. Acesso em: 28 de julho de 2019. Citado na página 24.

ANDRADE, M. **SystemMonitor**. 2019. Disponível em: <https://github.com/marcusaurelioo/SystemMonitor>. Acesso em: 21 de julho de 2019. Citado na página 24.

FOROUZAN, B. **Comunicação de Dados e Redes de Computadores**. 4ª edição. ed. [S.l.]: Mc Graw Hill, 2008. Citado 2 vezes nas páginas 11 e 15.

GAYRAUD, R. et al. **SIPp reference documentation**. 2017. Disponível em: <http://sipp.sourceforge.net/doc/reference.html>. Acesso em: 02 de novembro de 2017. Citado 2 vezes nas páginas 23 e 32.

KAMAILIO. **Kamailio - The Open Source SIP Server**. 2017. Disponível em: <https://www.kamailio.org/w/features/>. Acesso em: 02 de novembro de 2017. Citado na página 24.

KASPERSKY, L. **Kaspersky Lab Finds Businesses are Unclear on How to Combat Targeted Attacks and DDoS**. 2017. Disponível em: <https://usa.kaspersky.com/about/press-releases/2017_kaspersky-lab-finds-businesses-are-unclear-on-how-to-combat-targeted-attacks-and-ddos>. Acesso em: 09 de outubro de 2017. Citado na página 11.

KASPERSKY, L.; B2B, I. **Lose a Fortune: One DDoS Attack Can Cost a Company Over $1.6M**. 2016. Disponível em: <https://www.kaspersky.com/about/press-releases/2016_lose-a-fortune-one-ddos-attack-can-cost-a-company-over-1.6m>. Acesso em: 09 de outubro de 2017. Citado na página 12.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet**: Uma abordagem top-down. 6ª edição. ed. [S.l.]: Pearson, 2014. Citado 7 vezes nas páginas 14, 15, 16, 17, 18, 19 e 23.

PARK, P. **Voice Over IP Security**. [S.l.]: Cisco Press, 2009. Citado na página 11.

PERCATORE, J. **DDoS Attacks Advancing and Enduring: A SANS Survey**. [S.l.]: SANS Institute, 2014. Citado na página 11.

PETERSON, L.; DAVIE, B. **Redes de Computadores: Uma Abordagem de Sistemas**. 5ª edição. ed. [S.l.]: Elsevier, 2013. Citado 3 vezes nas páginas 12, 15 e 17.

PFSENSE. **Features List**. 2017. Disponível em: <https://doc.pfsense.org/index.php/Features_List>. Acesso em: 05 de novembro de 2017. Citado na página 23.

QUEROL, A. M. **VoIP Network Analyzer**. Dissertação (Mestrado) — Universitat Politècnica de Catalunya, 2016. Citado na página 24.

RIBEIRO, A.; PEREIRA, H. **Classification and Policing in the pfSense Platform**. Dissertação (Mestrado) — University of Minho, Department of Informatics, 2009. Citado na página 16.

SEMERCI, M.; CEMGIL, A. T.; SANKUR, B. An intelligent cyber security system against ddos attacks in sip networks. **Computer Networks**, v. 136, p. 137 – 154, 2018. ISSN 1389-1286. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128618300987>. Citado na página 21.

SEO, D.; LEE, H.; NUWERE, E. Sipad: Sip–voip anomaly detection using a stateful rule tree. **Computer Communications**, v. 36, n. 5, p. 562 – 574, 2013. ISSN 0140-3664. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366412004124>. Citado na página 21.

STANEK, J.; KENCL, L. Sip protector: Defense architecture mitigating ddos flood attacks against sip servers. In: **2012 IEEE International Conference on Communications (ICC)**. [S.l.: s.n.], 2012. p. 6733–6738. ISSN 1550-3607. Citado 3 vezes nas páginas 20, 23 e 40.

TANENBAUM, A. S. **Redes de Computadores**. 4ª edição. ed. [S.l.]: Elsevier, 2003. Citado na página 11.

TANENBAUM, A. S.; STEEN, M. V. **Sistemas Distribuídos**: Princípios e paradigmas. 2ª edição. ed. [S.l.]: Pearson, 2008. Citado 3 vezes nas páginas 15, 19 e 24.

TANENBAUM, A. S.; WETHERALL, D. **Redes de Computadores**. 5ª edição. ed. [S.l.]: Pearson, 2011. Citado 4 vezes nas páginas 12, 14, 15 e 18.

TAS, I. M.; UGURDOGAN, B.; BAKTIR, S. Novel session initiation protocol-based distributed denial-of-service attacks and effective defense strategies. **Computers & Security**, v. 63, n. Supplement C, p. 29 – 44, 2016. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404816300980>. Citado na página 21.

WILLIAMSON, M. **pfSense 2 Cookbook**: A pratical, example-driven guide to configure even the most advanced features of pfsense 2. [S.l.]: Packt Publishing, 2011. Citado na página 16.