

CURSO DE CIÊNCIA DA COMPUTAÇÃO

Tiago Silva Leal

**DETECÇÃO E ANÁLISE PROATIVA DE
ANOMALIAS NO TRÁFEGO DE REDE**

Santa Cruz do Sul
2017

Tiago Silva Leal

**DETECÇÃO E ANÁLISE PROATIVA DE
ANOMALIAS NO TRÁFEGO DE REDE**

Trabalho de Conclusão apresentado ao Curso de
Ciência da Computação da Universidade de Santa
Cruz do Sul, como requisito parcial para obtenção
do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Lucas Fernando Müller

Santa Cruz do Sul
2017

AGRADECIMENTOS

Agradeço primeiramente a minha esposa e aos meus pais por terem me apoiado durante toda esta jornada, principalmente nos momentos mais difíceis onde, sempre se fizeram presentes para me motivar a continuar esta jornada.

Agradeço a todos os meus amigos que de uma forma ou outra contribuíram para que este trabalho fosse possível, tanto com incentivo quanto com compreensão, principalmente da minha ausência durante esta etapa.

Um agradecimento especial ao meu orientador Lucas Fernando Müller pela amizade e por todo o apoio dado durante o desenvolvimento deste trabalho.

Não poderia deixar de agradecer também ao meu chefe que compreendeu muitas das minhas ausências para dedicação a este trabalho.

Por fim, agradeço aqueles que não foram citados, mas que de uma forma ou outra contribuíram para este trabalho.

Muito obrigado a todos.

“Quando penso que cheguei ao meu limite, descubro que tenho forças para ir além.”

Ayrton Senna

RESUMO

As redes de computadores são fundamentais no dia-a-dia das empresas. Em particular as redes locais (LANs) representam uma parte vital na indústria, o que ao longo do tempo gerou uma dependência operacional, isto é, resultados positivos dependem do seu correto funcionamento. Isso, somado a utilização das redes a níveis não estimados, onde o padrão de tráfego varia muito, tornou difícil de diagnosticar e manter operacionais as redes nas situações que fogem do padrão de comportamento normal. Para auxiliar na proteção e disponibilidade é necessário analisar o tráfego para detectar possíveis anomalias que podem ocorrer. Atualmente, diversas soluções baseadas em detecção e análise de anomalias no tráfego de rede são encontradas na literatura, porém não focadas em redes locais. Diante do exposto, o presente trabalho de conclusão de curso tem como objetivo propor uma ferramenta de detecção e análise proativa de anomalias no tráfego de redes locais. Para tanto, será utilizada a metodologia de detecção baseada em conhecimento junto com a análise de tráfego a fim de prover uma base de assinaturas de anomalias conhecidas nas redes locais. A base de assinaturas será utilizada como peça fundamental da ferramenta proposta para identificação de anomalias no tráfego da rede. Este trabalho ainda apresenta como objetivo a experimentação prática da ferramenta em infraestruturas de redes reais, considerando empresas de pequeno, médio e grande porte da região.

Palavras chave: Detecção de anomalia, Anomalia no tráfego de rede, Análise de tráfego de rede.

ABSTRACT

The computers networks are fundamental in the day-to-day of companies. In particular, the local networks (LANs) represent a vital part of the industry, which over time has generated operational dependence, positive results depend on the correct functioning. This situation, coupled with the use of networks at non-estimated levels, where the pattern of traffic varies greatly, has made it difficult to diagnose and maintain operational networks in situations that deviate from normal behavior patterns. To assist in the protection and availability it is necessary to analyze the traffic to detect possible anomalies. Currently, several solutions based on detection and analysis of network traffic anomalies are found in the literature, but not focused on local networks. Given the above facts, the present work of course completion aims to propose a tool for detection and analysis proactive of anomalies in local network traffic. Therefore, it will be using the detection methodology based on knowledge, together with traffic analysis to provide a database of signatures concerning anomalies known in local networks. The signature base will be a key part of the proposed tool for identifying anomalies in network traffic. This work also presents the goal of using the tool in real network infrastructures, considering small, medium and large companies in the region.

Keywords: Anomaly detection, Network traffic anomaly, Network traffic analysis.

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledgement
API	Application Programming Interface
ARP	Address Resolution Protocol
CRC	Cyclic Redundancy Check
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DPI	Deep Packet Inspection
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
LAN	Local Area Network
LTS	Long Term Support
MAC	Media Access Control
MS-SQL	Microsoft SQL Server
NAK	Negative Acknowledgment
NMAP	Network Mapper
PCA	Principal Component Analysis
PCAP	Packet Capture
RPC	Remote Procedure Call
SBS	Sequential Backward Selection
SDN	Software Defined Network
SIP	Session Initiation Protocol
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SPAN	Switched Port Analyzer
SPAM	Sending and Posting Advertisement in Mass
SSH	Secure Shell
SVM	Support Vector Machines
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
TTL	Time-to-live
TTS	Trouble Ticket Systems
UDP	User Datagram Protocol
VPN	Virtual Private Network
WAN	Wide Area Network

LISTA DE FIGURAS

Figura 1 - Principais componentes associados com as técnicas de detecção de anomalias.	14
Figura 2 - Processo de detecção de anomalia baseado em conhecimento.	19
Figura 3 - Rede com bloqueio de tráfego.	21
Figura 4 - Captura de tráfego (<i>sniffing</i>) em switches com SPAN.	23
Figura 5 - Captura de tráfego (<i>sniffing</i>) utilizando TAP.	24
Figura 6 - Coleta de dados com Wireshark.	28
Figura 7 - Exemplo de coleta de dados com Tshark.	28
Figura 8 - Mostra a confirmação de pacotes ACK.	29
Figura 9 - Retransmissão TCP, sinal que a conexão está fraca ou caiu.	29
Figura 10 - A ocorrência de retransmissões (problema de conectividade).	30
Figura 11 - Envio de ICMP (ping) do host 24.6.126.218 para um host remoto.	30
Figura 12 - Pacote de ping com um valor de time-to-live de 1 salto.	31
Figura 13 - Mostra o valor de TTL aumentando ao passar por mais redes.	31
Figura 14 - Host de origem reenvia solicitação após não receber retorno.	32
Figura 15 - Host de origem faz mais uma tentativa.	32
Figura 16 - Tempo de vida do pacote excedido.	32
Figura 17 - Estrutura do framework proposto.	36
Figura 18 - Estrutura da arquitetura proposta.	38
Figura 19 - Fluxograma do funcionamento do sistema.	46
Figura 20 - Menu de cadastros do sistema.	47
Figura 21 - Listagem categorias de anomalias.	47
Figura 22 - Cadastro de perfil de anomalia.	48
Figura 23 - Ativação perfil de anomalia.	49
Figura 24 - Alertas disponibilizados pelo sistema.	50
Figura 25 - Gráfico média de pacotes/segundo na rede.	51
Figura 26 - Gráfico latência média da rede.	52
Figura 27 - Gráfico média de pacotes retransmitidos na rede.	52
Figura 28 - Cadastro de usuário no sistema.	53
Figura 29 - Diagrama de sequência coleta de tráfego.	53
Figura 30 - Formato do arquivo de captura gerado.	54
Figura 31 - Definição da assinatura de identificação de anomalia.	55
Figura 32 - Tráfego normal solicitação de DHCP.	57
Figura 33 - Tráfego conflito de DHCP.	58
Figura 34 - Número de pacotes transmitidos na rede (<i>loop</i> na rede).	59
Figura 35 - Cenário conflito de DHCP na rede.	60
Figura 36 - Cenário de <i>loop</i> na rede.	61

Figura 37 - Tráfego referente ao conflito de DHCP do cenário 1.....	62
Figura 38 - Identificação em tempo real do conflito de DHCP.	63
Figura 39 - Identificação em tempo real do <i>loop</i> na rede.....	64
Figura 40 - Identificação em tempo real de IP duplicado na rede.....	65
Figura 41 - Interferência do cabo de força junto ao cabo de rede extraída de ambiente real.....	65
Figura 42 - Identificação em tempo real de erro de CRC na rede.	66
Figura 43 - Identificação em tempo real rede lenta (Internet).	68
Figura 44 - Alertas gerados conflito de DHCP.....	69
Figura 45 - Alertas gerados <i>loop</i> na rede.	71
Figura 46 - Alertas gerados IP duplicado na rede.....	72
Figura 47 - Alertas gerados erro de CRC na rede.....	73
Figura 48 - Alertas gerados rede lenta (Internet).....	74

LISTA DE TABELAS

Tabela 1 - Resultado de fusão para o tráfego SPAM.	35
Tabela 2 - Estatística da coleta nos honeypots.	39
Tabela 3 - Proporção de volume de tráfego para cada cenário de teste.....	42
Tabela 4 - Comparação das diferentes características dos trabalhos relacionados estudados.....	43
Tabela 5 - Especificação dos ambientes de avaliação.....	56

SUMÁRIO

1	INTRODUÇÃO	11
2	DETECÇÃO DE ANOMALIAS	13
2.1	Definição	13
2.2	Caraterísticas de um problema de detecção de anomalia	15
2.3	Classificação das técnicas de detecção de anomalias	18
2.4	Detecção de anomalia baseada em conhecimento	19
2.5	Considerações	20
3	CARACTERIZAÇÃO E ANÁLISE DE TRÁFEGO DE REDE	21
3.1	Análise de tráfego de rede	21
3.2	Captura de pacotes em redes de computadores	22
3.3	Técnicas para análise de pacotes	25
3.4	Principais ferramentas para captura e análise de tráfego de rede	26
3.5	Características e identificação de tráfego	29
3.6	Considerações	32
4	ESTADO DA ARTE	34
4.1	Detecção de anomalias	34
4.2	Análise de tráfego de rede	37
4.3	Síntese do estado da arte	42
4.4	Diferenças em relação à proposta atual	44
5	SISTEMA DE DETECÇÃO PROATIVA DE ANOMALIAS NO TRÁFEGO DA REDE LOCAL	45
5.1	Funcionamento da ferramenta	45
5.2	Ambiente de avaliação	56
5.3	Especificação do ambiente	56
5.4	Premissas	57
5.5	Cenário de avaliação	60
5.6	Experimentos	62
6	AVALIAÇÃO E RESULTADOS	69
7	CONCLUSÃO E TRABALHOS FUTUROS	75
	REFERÊNCIAS	77

1 INTRODUÇÃO

As redes de computadores são fundamentais no dia-a-dia das empresas. Em particular as redes locais (LANs) representam uma parte vital na indústria, o que ao longo do tempo gerou uma dependência operacional, isto é, resultados positivos dependem do correto funcionamento dessas redes. Isso, somado a utilização das redes a níveis não estimados, onde o padrão de tráfego de rede varia muito, tornou difícil de diagnosticar as situações fora do padrão de comportamento (MARNERIDES; SCHAEFFER-FILHO; MAUTHE, 2014). Para auxiliar na proteção, segurança e taxa de disponibilidade destas redes é necessário analisar e observar o tráfego de rede. Neste contexto, o diagnóstico de anomalias possui como objetivo descobrir e caracterizar padrões de problemas que afetam a infraestrutura de rede, podendo ser encontrados problemas de natureza maliciosa (por exemplo, ataques), ou não intencionais (falha de configurações, equipamentos, tráfego de vírus na rede) (MARNERIDES; SCHAEFFER-FILHO; MAUTHE, 2014).

De acordo com Marnerides e Mauthe (2016) a importância de uma identificação precoce quanto à detecção de anomalias na rede consiste em prover mais segurança, sendo de fundamental importância conhecer o que está sendo trafegado em nossa rede. A detecção de anomalias é um problema importante que tem sido alvo de diversas áreas de pesquisa e domínios de aplicação (CHANDOLA; BANERJEE; KUMAR, 2009).

Um desafio fundamental nesse contexto é identificar a melhor estratégia de acordo com cada infraestrutura de rede. Segundo Marnerides *et al.* (2011), a detecção de anomalias deve ser feita o mais cedo possível junto a rede, para aumentar a eficiência e para detectar anomalias de uma forma mais coordenada e efetiva. Desta forma, o mecanismo de detecção pode identificar a raiz de um determinado problema para assim, facilitar o processo de correção.

Baseado em casos reais, identificados através de dificuldades operacionais de redes de todos os portes (pequeno, médio e grande) que o objetivo deste trabalho foi definido. O objetivo geral deste trabalho é o desenvolvimento de uma ferramenta para detecção e análise proativa de anomalias no tráfego de rede local.

Será observado o tráfego que está passando na rede em tempo real (*online*) de forma que, ao detectar alterações nos padrões de tráfego de rede serão realizadas consultas a uma base de conhecimento de assinaturas de tráfego anômalo, identificando assim o possível problema.

Os objetivos específicos para o desenvolvimento deste trabalho são:

- Caracterização dos perfis de tráfego anômalo, para realizar a criação das assinaturas de tráfego anômalo.
- Preparação de ambiente para experimentação.
- Desenvolvimento dos códigos necessários para viabilizar a execução e avaliação de estratégias para identificação de anomalias na rede.
- Desenvolvimento de uma ferramenta para contribuir com o estado-da-arte na identificação de anomalias através da análise de tráfego utilizando metodologia baseada em conhecimento.

A seguir, no Capítulo 2, serão apresentados os fundamentos de detecção de anomalias, descrevendo os conceitos chave de anomalia, além das características, métodos, técnicas utilizadas para identificar e detectar anomalias. No Capítulo 3, serão descritos os conceitos sobre análise de tráfego de rede, assim como as características de tráfego, formas de captura de pacotes e técnicas utilizadas para realizar a análise de tráfego de rede. Já no Capítulo 4 serão detalhados estudos realizados, recentemente, dedicados à detecção e análise de anomalias no tráfego de rede. No Capítulo 5 é descrita a proposta desenvolvida no presente trabalho. No Capítulo 6 é descrita a avaliação e os resultados obtidos e por fim, no Capítulo 7, serão apresentadas as considerações finais sobre os estudos realizados.

2 DETECÇÃO DE ANOMALIAS

Este capítulo apresenta as definições e características do processo de detecção de anomalias, assim como as técnicas utilizadas para detecção.

2.1 Definição

As anomalias no tráfego de rede são inerentes a maneira que a Internet funciona hoje. Essas anomalias são eventos que causam um desvio (alteração) em relação ao perfil padrão de comportamento. As anomalias podem ser induzidas nos dados através de uma série de razões, tais como atividade maliciosa, por exemplo, fraude de cartão de crédito, ataques de intrusão, atividade terrorista, problema no sistema, dentre outros aspectos. Em um nível abstrato, uma anomalia é definida como um padrão fora do comportamento normal esperado (CHANDOLA; BANERJEE; KUMAR, 2009).

De um modo amplo, uma anomalia no contexto de rede pode ocorrer devido a um ataque, falha de equipamento, problemas de configuração, sobrecarga ou uso abusivo ou inadequado de algum serviço ou recurso da rede (MARNERIDES; MAUTHE, 2016).

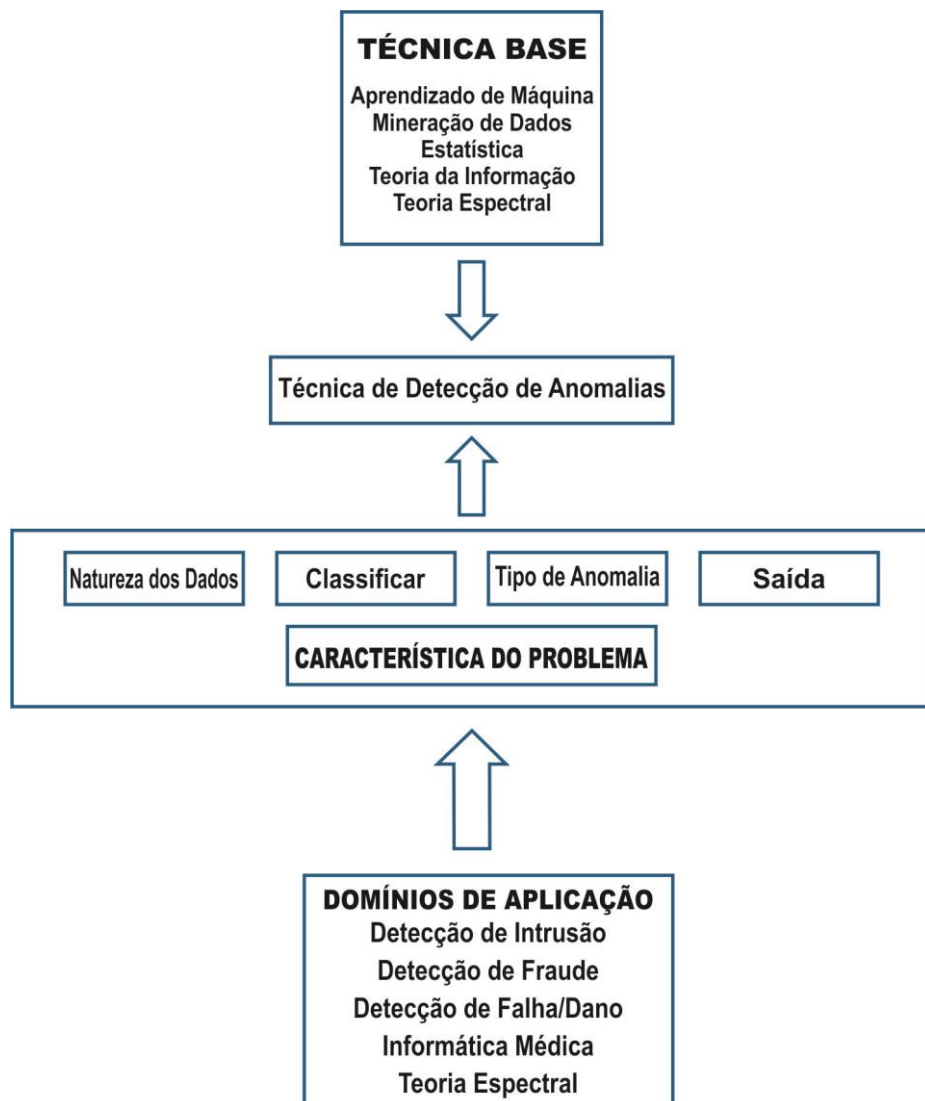
A detecção de anomalias em redes de computadores é uma área de estudo bastante ativa e várias técnicas são usadas. A classificação das técnicas de detecção de anomalias de rede presentes na literatura, é uma tarefa difícil devido à diversidade e ao desenvolvimento constante de novas técnicas (MARNERIDES; SCHAEFFER-FILHO; MAUTHE, 2014).

De acordo com Wattenberg *et al.* (2011) as propostas de investigação em detecção de anomalias normalmente seguem uma abordagem de quatro estágios, em que as três primeiras etapas é definir o método de detecção, enquanto que a última fase seja dedicada a validar a abordagem. Assim, na primeira fase, são coletados os dados do tráfego a partir da rede. Em segundo lugar, os dados são analisados para extrair suas características mais relevantes (análise de dados). Em terceiro lugar, o tráfego é classificado como normal ou anormal. Em quarto, toda a abordagem é validada com vários tipos de anomalias de tráfego (validação).

Segundo Chandola, Banerjee e Kumar (2009), as técnicas de detecção de anomalias existentes, além de seguirem as etapas mencionadas por Wattenberg *et al.* (2011), devem estar introduzidas no contexto do problema específico. Deve levar em conta alguns fatores tais como natureza dos dados, classificação dos dados, tipos de anomalias a serem

detectado. Muitas vezes estes fatores são determinantes para o domínio da aplicação em que a anomalia precisa ser detectada. Os pesquisadores adotam conceitos de diversas disciplinas, tais como estatística, aprendizagem de máquina, mineração de dados dentre outras. A Figura 1 mostra os componentes essenciais acima mencionados associados com qualquer técnica de detecção de anomalia.

Figura 1 - Principais componentes associados com as técnicas de detecção de anomalias.



Fonte: Adaptado de Chandola *et.al*, 2009.

A Figura 1 apresenta o escopo de uma técnica de detecção de anomalia, onde na primeira etapa temos a decisão de área da técnica base que vai dar sustentação para a técnica proposta. A segunda etapa consiste na decisão do domínio bem como a análise das

características do domínio da aplicação. Ao final, são obtidos os principais elementos para a definição da técnica de detecção de anomalias.

2.2 Características de um problema de detecção de anomalia

Esta seção explica os diferentes aspectos de detecção de anomalias. A formulação específica de um problema é determinada por fatores diferentes, tais como: a natureza dos dados de entrada, a disponibilidade (ou indisponibilidade) da classificação dos dados, bem como as restrições e requisitos induzidos pelo domínio da aplicação (CHANDOLA; BANERJEE; KUMAR, 2009).

Natureza dos dados de entrada

Um dos principais aspectos de qualquer técnica de detecção de anomalias é a natureza dos dados de entrada. A entrada é uma coleção de instâncias de dados (conhecidos como objetos, registro, ponto, vetor, padrão, evento, caso, amostra, observação, entidade) (TAN, STEINBACH, KUMAR, 2005). Cada instância de dados pode ser descrita utilizando um conjunto de atributos, sejam eles variáveis ou características. Os atributos podem ser de tipos diferentes como binário, categórica, ou contínua. A natureza dos atributos determina a aplicabilidade da lógica de detecção de anomalias.

Classificação de Anomalias

De uma forma ampla, as anomalias podem ser classificadas de três formas: ponto de anomalia, anomalias contextuais, coletiva (GALVÃO, 2013).

- Ponto de anomalia: refere-se a uma instância de dados individuais que podem ser considerados anômalos referentes ao resto dos dados, então o exemplo é denominado como um ponto anomalia. Este é o tipo mais simples de anomalia e é o foco da maioria das pesquisas sobre a detecção de anomalias. Podemos citar como exemplo uma operação de fraude em cartão de crédito onde elencamos apenas uma característica: montante gasto. Uma transação para a qual o montante gasto é muito elevado em comparação com o intervalo normal das despesas deste cliente pode caracterizar um ponto de anomalia.

- Anomalias contextuais: caso uma instância de dados seja anômala em um contexto específico, mas não em outro, é denominada como anomalia contextual. O contexto é induzido pela estrutura do conjunto de dados analisado. Cada instância de dados é definida de acordo com os atributos contextuais e comportamentais. O comportamento anômalo é determinado usando os valores para os atributos comportamentais dentro de um contexto específico. A instância de dados pode ser uma anomalia contextual em um determinado contexto, mas um exemplo de dados idênticos (em termos de atributos comportamentais) poderia ser considerado normal num contexto diferente. Esta propriedade é fundamental na identificação dos atributos contextuais e comportamentais para uma técnica de detecção de anomalias contextual.
- Anomalias coletivas: ocorre quando uma instância de dados relacionados é anômala no que diz respeito a todo o conjunto de dados. As instâncias de dados individuais em uma anomalia coletiva podem não ser anomalias por si mesmas, mas a sua ocorrência em conjunto, como uma coleção é denominada anomalia. Como exemplo dentro do contexto de rede de computadores, temos a seguinte sequência de tráfego coletada: http-web, buffer-overflow, http-web, http-web, smtp-mail, ftp, http-web, buffer-overflow, ssh, ftp ssh, smtp-mail, http-web, ssh, buffer-overflow, ftp, http-web, ftp, smtp-mail, http-web.

A sequência destes eventos destacados (buffer-overflow, ssh, ftp) correspondem a um ataque web típico baseada por uma máquina remota que copia os dados de um determinado computador para o destino remoto via FTP (*File Transfer Protocol*). Esta coleção de eventos é uma anomalia, mas os eventos individuais não são anomalias quando ocorrerem separadamente.

Podem ocorrer anomalias pontuais em qualquer conjunto de dados, anomalias coletivas podem ocorrer apenas em conjunto de dados em que instâncias de dados estejam relacionadas. Já nas anomalias contextuais depende da disponibilidade do contexto e seus atributos de dados. Um ponto de anomalia ou anomalia coletiva também pode vir a ser uma anomalia contextual se analisarmos no que diz respeito ao contexto.

Análise e Classificação de Dados

Para realizar a classificação de dados e determinar se uma instância é normal ou anômala, dispomos de três formas de detecção: supervisionado, semi supervisionado e não supervisionado.

Detecção de anomalia supervisionada: nesta categoria se assume que há disponibilidade de um conjunto de dados de treinamento que tem marcado as instâncias referentes ao comportamento normal bem como a classe de anomalia. É construída uma previsão de modelo normal comparado com as classes de anomalias e qualquer instância de dados é comparada com o modelo para determinar à qual classe pertence. Um grande desafio é a obtenção exata e a classificação representativa, especialmente para a classe de anomalia. Algumas técnicas têm sido propostas para injetar anomalias artificialmente em dados normais definidos para obter um conjunto de dados de treinamento (ABE *et al.*, 2006) (STEINWART, HUSH, SCOVEL, 2005).

Detecção de anomalia semi supervisionada: assume-se que os dados de treinamento só possuem tipos de dados da classe normal. A abordagem típica usada em tais métodos é a construção de um modelo para a classe normal, e usar o modelo para identificar anomalias nos dados de teste. Há um conjunto limitado de técnicas de detecção de anomalias existentes que assumem disponibilidade apenas nos casos de formação de anomalias. Tais técnicas não são normalmente usadas, principalmente, pela dificuldade em obter um conjunto de dados de treinamento que abrange todos os possíveis comportamentos anômalos que podem ocorrer nos dados.

Detecção de anomalia não supervisionada: não requer a utilização de um modo de supervisão e não exige treinamento, fatos estes que levam a uma maior adoção da técnica. As técnicas nesta categoria fazem a suposição implícita de que instâncias normais são muito mais frequentes do que as anomalias nos dados.

Possui como desvantagem a alta taxa de falsos alarmes. Muitas técnicas de modo semi supervisionada podem ser adaptadas para trabalhar em um modo sem supervisão, utilizando uma amostra não classificada como dado de treino. Esta adaptação assume que os dados de teste possuem poucas anomalias e o modelo aprendeu de forma robusta durante o treinamento estas poucas anomalias.

Saída de detecção de anomalias

Um aspecto bastante importante para as técnicas de detecção de anomalias é a maneira como a anomalia é relatada. Geralmente, as saídas produzidas pelas técnicas de detecção de anomalias são divididas em dois tipos: baseados em pontuação ou classificação.

Pontuação: é atribuída uma pontuação de anomalia para cada instância nos dados, dependendo do grau em que tal exemplo se encaixa é considerada anomalia. Logo a produção de tais técnicas gera como saída uma lista ordenada de anomalias.

Classificação: é atribuída de acordo com a classificação normal ou anômala, para cada instância de dados. Fornece como saída os dados populados nas classes normal e anômala.

2.3 Classificação das técnicas de detecção de anomalias

Segundo Silva *et al.* (2015) existem um conjunto de ferramentas para simulação e análise de tráfego anômalo, assim como uma vasta variedade de algoritmos e técnicas para classificação. De acordo com Teodoro *et al.* (2009), o autor classificou os métodos de detecção de anomalias de rede em métodos baseados: Conhecimento, Aprendizagem de Máquina e Análise Estatística.

- **Conhecimento:** Máquina de estados finitos, sistemas especialistas ou baseado em regras, busca por padrões (*Pattern Matching*).
- **Aprendizagem de Máquina:** Redes bayesianas, redes neurais, lógica difusa (Fuzzy), algoritmos genéticos, algoritmos de agrupamento (Clustering).
- **Análise de Sinais:** Análise estatística filtros de Kalman, CUSUM (CUMulative SUM), séries temporais, wavelets;

Os métodos baseados em conhecimento utilizam um conjunto de regras e parâmetros elaborados por um especialista implementando algum formalismo, como por exemplo, máquina de estados finita. Estes tipos de métodos são mais robustos, pois tendem a apresentar poucos falsos positivos, porém possuem uma desvantagem que consiste na dificuldade e demora em obter o conhecimento.

Os métodos baseados em aprendizagem de máquina utilizam como base um padrão implícito ao qual permite analisar os padrões e classificá-los. Para realizar a classificação destes padrões são utilizadas diversas técnicas, como por exemplo, redes neurais e algoritmos de agrupamento. Um ponto relevante desta abordagem está na necessidade de

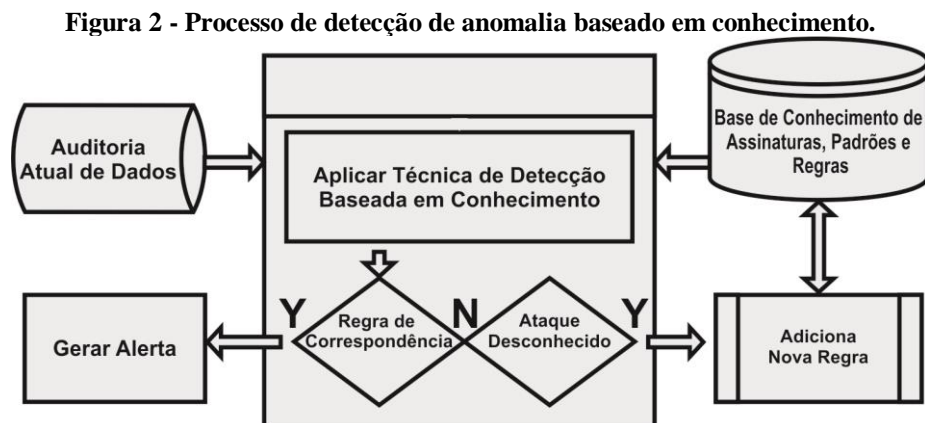
uma fase de treinamento com os dados classificados para diferenciação do comportamento normal ou anormal pelo sistema. Este método possui como vantagens a flexibilidade, adaptabilidade e capacidade de capturar dependências desconhecidas nos dados.

Na abordagem baseada na análise de sinais, é realizada a criação de um perfil de comportamento que posteriormente é passado para a rede. Este perfil utiliza algumas métricas tais como: número de pacotes por protocolo e número de conexões, dentre outras. Caso o comportamento atual da rede venha a divergir do comportamento definido no perfil, um alerta de anomalia é gerado. Este método possui como vantagem não precisar de conhecimento prévio, pois é capaz de se adaptar ao comportamento da rede, porém a dificuldade fica por conta da definição dos parâmetros, algo que influencia na taxa de detecção e de falsos positivos (TEODORO *et al.*, 2009).

A próxima seção destaca a detecção de anomalias baseada em conhecimento na qual foi realizado um estudo mais profundo, por se tratar da metodologia utilizada para desenvolvimento do trabalho.

2.4 Detecção de anomalia baseada em conhecimento

A detecção de anomalia baseada em conhecimento analisa as atividades do sistema procurando por eventos que correspondam a padrões pré-definidos de ataques, falhas de configurações e outras atividades maliciosas e as reporta para o administrador. Estes padrões são conhecidos como assinaturas e geralmente cada assinatura corresponde a anomalia específica. Nesta abordagem o sistema possui conhecimento sobre as anomalias e mediante a esse conhecimento prévio efetua tentativas com o objetivo de reconhecê-la (LARI; AMARAL, 2004). Um sistema baseado em conhecimento gera um alarme quando a anomalia é detectada. A Figura 2 mostra o fluxo do processo.



Fonte: Adaptado de Nadeem e Howarth, 2013.

O processo de detecção de anomalia baseada em conhecimento solicita o conhecimento sobre um determinado problema na rede e se caso não for explicitamente reconhecido como este problema logo é declarada como não-problema, caso seja reconhecido gera um alerta (NADEEM; HOWARTH, 2013).

De acordo com Lari e Amaral (2004), este tipo de detecção possui alguns aspectos importante tais como:

- Uma vantagem é que produz poucos falsos positivos, pois possui uma base de dados com as assinaturas de problemas já conhecidos, é realizada a verificação do evento capturado pelo sistema, com base nesses dados são gerados alarmes que denunciam a presença de ações de potencial malicioso junto ao sistema.
- A necessidade do conhecimento onde o sistema de detecção será instalado, é de total importância para criação das assinaturas, ou seja, conhecermos os sistemas operacionais, aplicativos envolvidos nos servidores e nas estações de trabalho, especificação do hardware assim com a topologia da rede interna.
- A atualização das assinaturas é uma tarefa que requer uma análise bastante cuidadosa, pois se houver assinaturas que não tenham utilidade para seu ambiente, pode vir a elevar a taxa de falsos positivos.

2.5 Considerações

Através dos estudos realizados nas diferentes bibliografias citadas ao longo deste capítulo, podem-se verificar os principais aspectos para identificar uma anomalia. Posteriormente foram elencados fatores para a caracterização de uma anomalia dentro de um determinado contexto, onde foram abordados os tópicos: natureza dos dados, classificação dos dados, tipos de anomalia, saída dos dados.

Após definidas as características de uma anomalia, passamos para a etapa de classificação dos dados, onde abordamos os principais métodos de detecção mencionados no estado da arte, sendo destacada a detecção de anomalias baseada em conhecimento a qual é foco deste trabalho.

Os estudos realizados permitiram esclarecer quais são as etapas necessárias para identificar e classificar as anomalias, bem como as metodologias que podem ser aplicadas, de acordo com o contexto do problema. Por fim, discutimos as técnicas que podem ser aplicadas para detecção de anomalias.

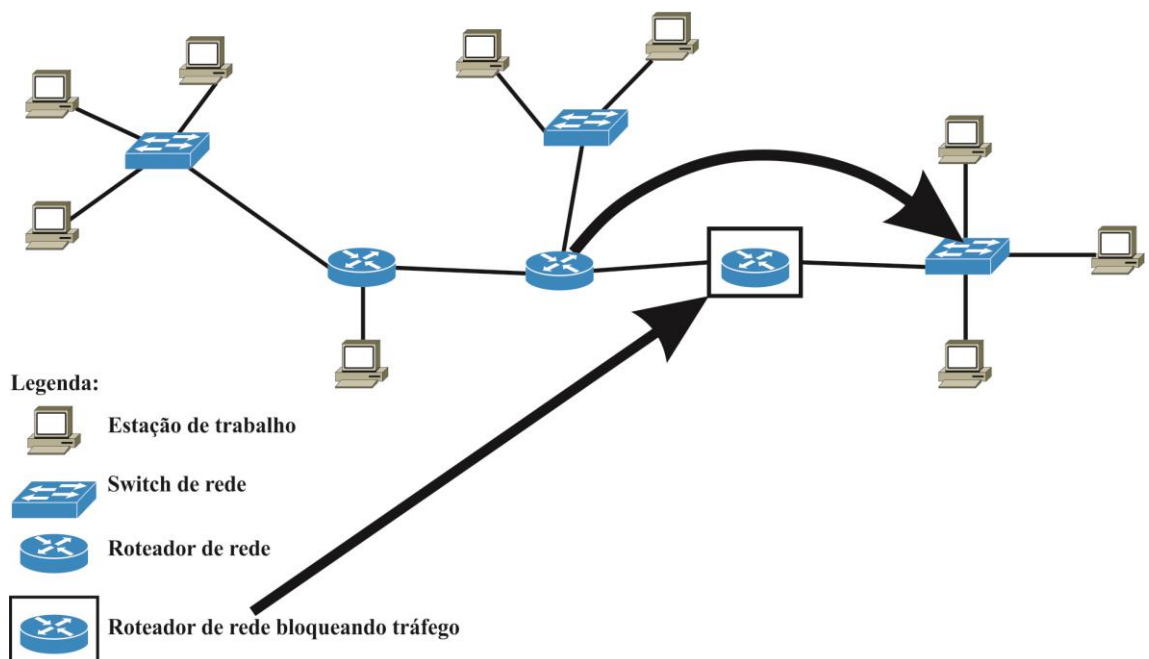
3 CARACTERIZAÇÃO E ANÁLISE DE TRÁFEGO DE REDE

Este capítulo busca explorar os conceitos da análise de tráfego de rede, assim como características de tráfego, formas de captura de pacotes e técnicas empregadas para realizar a análise de tráfego de rede.

3.1 Análise de tráfego de rede

A análise de tráfego é algo essencial para verificarmos como está nossa rede, através de uma análise podemos detectar rapidamente quais problemas estão ocorrendo em uma rede e onde eles estão. A análise de tráfego em tempo real permite investigar o fluxo de dados que está passando em um dado momento, permitindo assim identificar uma série de problemas que por vezes não são mostrados pelas aplicações, estações de trabalho ou equipamentos dentre eles: detectar anomalias na rede, encontrar pontos de bloqueio, descobrir equipamentos e cabearmentos defeituosos, observar importantes mensagens de sistema não mostradas pelas aplicações, detectar falhas de segurança, identificar tráfego de vírus na rede, pontos de bloqueio na rede deste outros (MOTA FILHO, 2013). Um exemplo é ilustrado na Figura 3.

Figura 3 - Rede com bloqueio de tráfego.



Fonte: Adaptado de Mota Filho, 2013.

Na Figura 3 é possível visualizar uma rede com uma obstrução, uma vez que um dos roteadores está bloqueando todo o tráfego. Devido a este problema não há comunicação entre os dois segmentos de rede interligada por este roteador. Esta é uma das situações onde se pode tirar proveito da análise de tráfego, que permite em meio a vários roteadores identificar qual é o que está com problema.

A análise de tráfego é realizada efetuando a leitura dos pacotes que estão trafegando na rede, estes pacotes carregam consigo cabeçalhos de diferentes protocolos de rede. Os protocolos definem o formato, ordem de mensagens enviadas e recebidas entre entidades de rede, e ações tomadas sobre transmissão e recepção de mensagens, influenciando diretamente no funcionamento de uma rede e dos serviços existentes.

Protocolo IPv4

De acordo com Kurose e Ross (2013), o protocolo IP possui grande importância na maioria dos processos de análises em tráfegos de redes, pois será preciso e necessário obter informações sobre IPs de origem e destino, como por exemplo:

- Identificação dos hosts de origem e destino referente ao tráfego analisado.
- Identificação de hosts com endereços IP não pertencentes à rede.
- Permite a identificação dos endereços de servidores e ativos de rede (baseando-se no tráfego de consulta/resposta e serviços em execução e/ou tráfego *broadcast* na rede).
- Identificação de ataques do tipo IP *Spoofing* (falsificação de endereço IP de origem).
- Verificar sessões ativas (o endereço IP é parte da informação que identifica um *socket*, juntamente com dados da camada de transporte).

3.2 Captura de pacotes em redes de computadores

Nesta subseção vamos abordar os conceitos e práticas sobre a captura de dados em redes de computadores onde será apresentando o formato dos dados capturados e técnicas para coleta de dados.

Segundo Mota Filho (2013), o principal agente envolvido na captura de dados em redes é chamado de *sniffer* (ou *packet sniffer*), técnica que envolve componentes de hardware e software, capaz de capturar tráfego em redes cabeadas ou sem fio.

Os *sniffers* ativam o modo de captura de pacotes, que consiste na alteração do comportamento padrão de uma interface de rede habilitando o seu modo “promíscuo” (ou modo “monitor”, no caso de redes sem fio). Com esta mudança, a interface de rede, cujo funcionamento normal consiste só em capturar e repassar os pacotes destinados ao próprio host ou pacotes de *broadcast*, passa a capturar e repassar às camadas superiores todo o tráfego que passa pelo canal físico de comunicação ao qual está conectada (independente da origem e do destino desses pacotes).

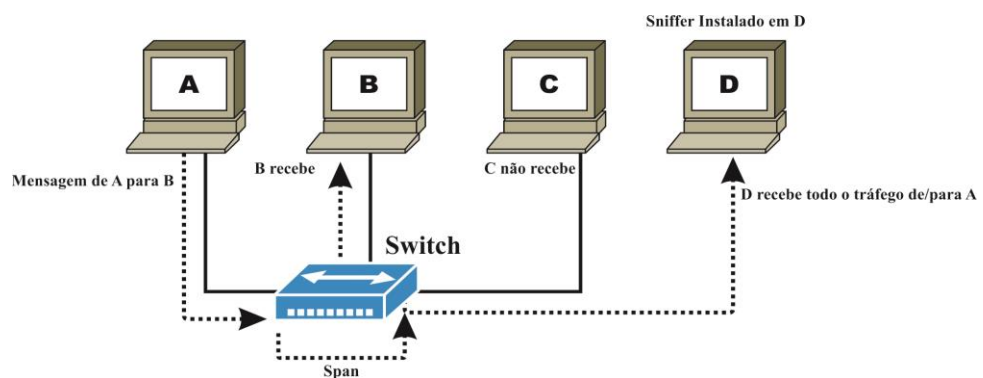
Desta forma todos os equipamentos que conectarem na rede, estão passíveis que seus dados trafegados na rede sejam capturados e analisados pelo software que gerência o processo de *sniffing*. Algo bastante importante no processo de captura é conhecer os detalhes da topologia da rede, como tipos de equipamentos, protocolo de comunicação que trafegam pela rede, pois estas informações podem implicar em limitações maiores ou menores no processo de captura de tráfego.

Captura de tráfego através de portas de monitoramento

As portas de monitoramento, também conhecidas como portas de espelhamento (*port mirror*) e SPAN (*Switched Port Analyzer*), são recursos disponíveis em alguns switches de rede que consistem em configurar uma porta específica no switch para receber, passivamente, cópias de todo o tráfego de uma ou mais portas do equipamento.

Assim, com o *sniffer* conectado a uma SPAN é possível monitorar o tráfego de uma ou mais portas do equipamento, de acordo com a configuração realizada, conforme Figura 4.

Figura 4 - Captura de tráfego (*sniffing*) em switches com SPAN.



Fonte: Adaptado de Galvão, 2013.

Na Figura 4 podemos visualizar que tanto o comportamento do switch como do host com o *sniffer* em execução, não interfere no tráfego das portas monitoradas, deixando esta solução imperceptível aos hosts monitorados. O único impacto na rede é em relação a ativação de SPAN que aumenta o uso dos recursos de processamento do switch quando habilitada.

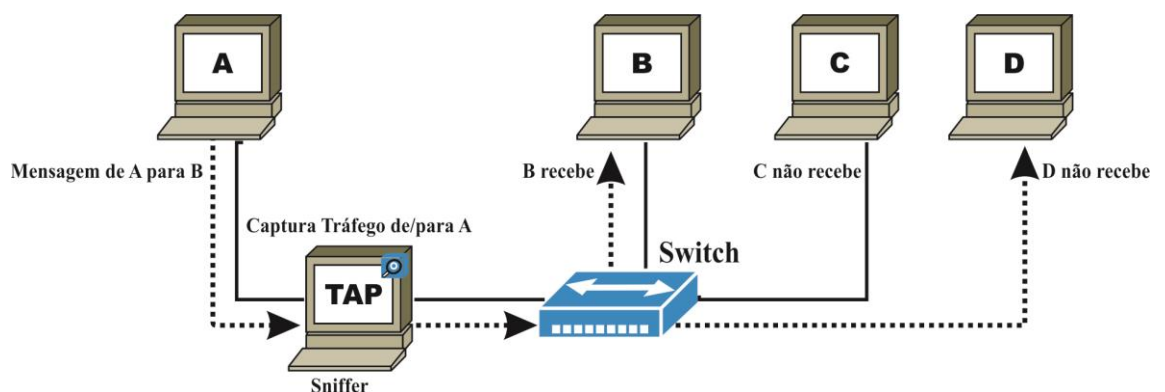
Captura de tráfego através de interceptação intermediária

Esta técnica de captura consiste em instalar um *sniffer* em uma rede e posicioná-lo entre o host e ou rede que se deseja monitorar e o ativo de rede (hub ou switch), forçando o tráfego de origem e destino para o host e ou rede monitorando o host onde o *sniffer* está instalado.

Para deixar este equipamento imperceptível ao host, é preciso configurar o mesmo em modo ponte (*bridge*), e não como roteador, não necessitando, assim, realizar quaisquer modificações nas configurações de rede do host monitorado (GALVÃO, 2013).

O equipamento intermediário onde está instalado o *sniffer* é conhecido como *network tap* onde necessita ter duas placas de rede para receber em uma delas o cabo de rede vindo do host e ou rede monitorado e na outra conectar o cabo que segue até o hub e ou switch da rede conforme Figura 5.

Figura 5 - Captura de tráfego (*sniffing*) utilizando TAP.



Fonte: Adaptado de Galvão, 2013.

O *network tap* não retém o tráfego, apenas repassa os pacotes de uma interface para outra. Em função disso, ativação de um *sniffer* nesse equipamento que pode ser um computador, pode realizar uma cópia de todo o tráfego que passa, ou parte dele, no caso do uso de filtros.

3.3 Técnicas para análise de pacotes

Esta seção possui enfoque na análise dos dados capturados, onde vamos abordar conceitos, técnicas para identificação e análise de pacotes. Existem três técnicas para realização de análise de pacotes em redes de computadores:

- ***Pattern matching (casamento de padrões)***: consiste na identificação dos pacotes, com maior relevância, através da combinação de valores específicos durante a captura ou filtragem de pacotes a partir dos arquivos de captura, antes de uma análise mais detalhada (HONGYI *et al.*, 2014).
- ***Parsing Protocol fields (análise dos campos dos protocolos)***: trabalha através da extração do conteúdo dos campos do protocolo (campos específicos de cada cabeçalho e *payload*).
- ***Packet filtering (filtragem de pacotes)***: responsável por separar os pacotes com base nos valores dos campos de metadados de cada protocolo.

Pattern matching

Esta etapa se caracteriza por definir o escopo da investigação antes de iniciar a captura ou filtragem (pré-análise). Este escopo é definido de acordo com o contexto do que se esta procurando dentro de um determinado problema. Por exemplo, se a análise é baseada em suspeita de atividade maliciosa, a partir de um host, logo o escopo inicial é o tráfego de entrada e saída para este host.

Assim deve-se excluir do processo de captura e ou filtragem, pacotes que não dizem respeito ao host específico. Porém, quando a busca e ou suspeita inicial é baseada em suspeitas sem ter a origem e destino específico, deve-se utilizar uma ferramenta para buscar informações que estejam trafegando dentro dos *payloads* dos protocolos de aplicação independentemente de host.

Parsing protocol fields

Esta etapa consiste em realizar uma busca detalhada nos pacotes coletados na etapa anterior a fim de extrair dados dos campos de cabeçalho e *payload* dos protocolos envolvidos, através de ferramentas de extração como, por exemplo, o Tskark (WIRESHARK, 2016), onde informa os hosts identificados e os protocolos envolvidos e todos os campos significativos para uma extração completa.

Packet filtering

A filtragem de pacotes consiste na técnica de separação de pacotes por meio de filtros baseados em metadados dos protocolos e em suas cargas (*payloads*). Esta técnica consiste em aplicar filtros específicos identificados após a primeira captura, e realizar uma nova captura, porém adicionando parâmetros específicos para assim diminuir o volume do tamanho do arquivo de captura. De acordo com Nikhil *et al.* (2014) para realizar este procedimento podemos utilizar a ferramenta tcpdump, para assim realizar uma nova captura, por exemplo:

```
# tcpdump -X -vvv -n -i eth0 -s0 host 192.168.1.101 and host 11.22.33.44 /and port 28
-w captura2.pcap
```

Podemos verificar, neste exemplo, os filtros sendo aplicados ao host de origem, host de destino e porta de acesso. Na próxima seção veremos mais detalhes sobre a ferramenta tcpdump.

3.4 Principais ferramentas para captura e análise de tráfego de rede

De acordo com Shimonski (2014) uma ferramenta que realiza a captura de pacotes também é chamada de analisador de rede. Um analisador de rede é uma ferramenta para realizar a identificação e resolver problemas de comunicação em redes e também permite realizar otimizações nas redes.

Os analisadores capturam o tráfego que passa pela rede, decodificando o tráfego capturado de modo a identificar os diferentes protocolos. Os dados decodificados são mostrados em um formato que facilita a compreensão. Um analisador de rede também permite a utilização de filtros específicos, onde permite capturar somente o tráfego que seja relevante a um determinado problema.

Nesta subseção, serão abordadas as principais ferramentas utilizadas para análise de tráfego de rede, assim como seus conceitos, características e funcionalidades no cenário de análise de tráfego de rede.

Tcpdump

O tcpdump (TCPDUMP, 2016) é um analisador que captura pacotes de protocolos através de linha de comando. O tcpdump permite capturar pacotes e mostrar detalhes

específicos aos quais são fundamentais para análises mais detalhadas referentes há um determinado problema. Desta forma, o tcpdump mostra as conexões estabelecidas e o tráfego correspondente.

Ele é baseado na libcap, uma poderosa API para captura de pacotes de rede onde coloca a placa de rede em modo promiscuo possibilitando assim a captura de pacotes. O tcpdump permite comandos personalizados aos quais pode mostrar mais ou menos detalhes conforme for utilizado. É extremamente útil quando há necessidade de capturar dados frente há um determinado problema, pois no caso em sistemas UNIX já vem instalado por padrão. O tcpdump que é software livre roda em linha de comando, disponível em diversos sistemas operacionais, como LINUX, MAC OS, BSD entre outros.

Wireshark/Tshark

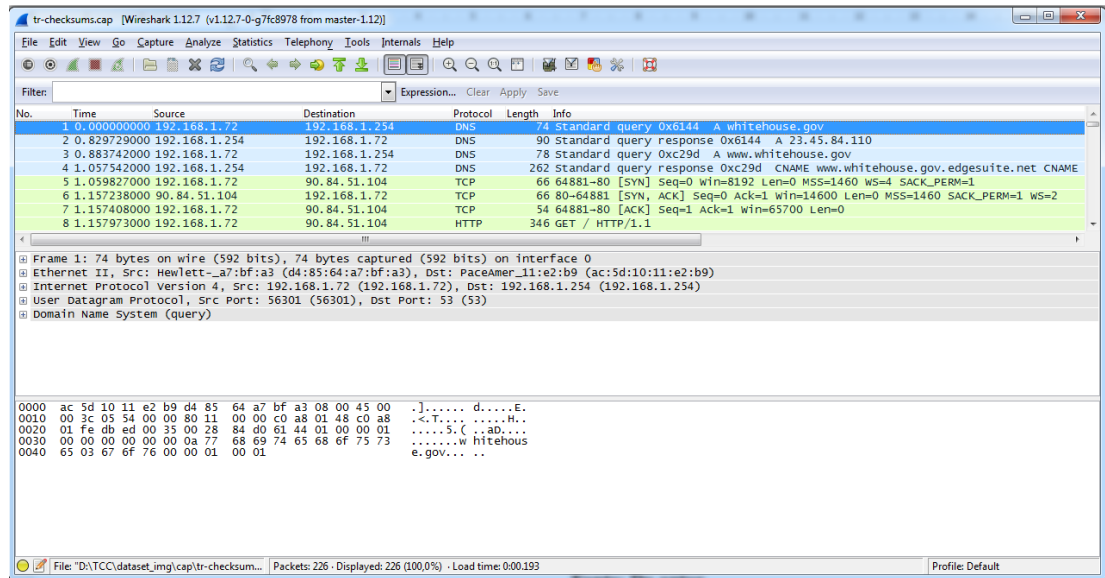
O Wireshark (WIRESHARK, 2016) é um analisador de protocolos de rede bastante utilizado, pode ser encontrado versões para as plataformas WINDOWS, MAC OS X, LINUX e UNIX, além disso trata-se de um software livre. O Wireshark é um analisador de protocolos de rede que também possui a funcionalidade de *sniffer*, que consistem em permitir realizar a captura dos pacotes em tempo real diretamente na ferramenta para realizar análises posteriormente.

A manipulação dos arquivos capturados são no formato pcap uma vez que também é baseado na libcap (WIRESHARK, 2016) assim com o tcpdump. O Wireshark trabalha sobre todas as camadas da pilha TCP/IP.

A principal vantagem de utilizar o Wireshark como *sniffer* e analisador de pacotes é a possibilidade de acompanhamento em tempo real, em sua interface gráfica, dos pacotes capturados que são exibidos detalhadamente pela ferramenta. Além disso, apresenta vários recursos que incluem inspeções detalhadas de protocolos, capturas em tempo real entre outros fatores (CHAPPEL, 2013).

É válido resaltar que não é aconselhável utilizar na maioria dos casos a funcionalidade de captura do Wireshark simultaneamente à análise dos pacotes, pois pode comprometer os recursos da máquina que está sendo utilizada (MOTA FILHO, 2013). A Figura 6 mostra uma coleta de dados em andamento.

Figura 6 - Coleta de dados com Wireshark.



Fonte: Do autor.

Na Figura 6 podemos verificar todo o tráfego que está passando em tempo real em nossa rede, através da interface gráfica do Wireshark.

Tshark

O Tshark (WIRESHARK, 2016) é um analisador de protocolo de rede, com funcionamento semelhante ao tcpdump. Ele permite capturar dados dos pacotes a partir de uma rede em tempo real, ou ler os pacotes a partir de um arquivo de captura salvo anteriormente. Utiliza o formato padrão de arquivo de captura pcap. A Figura 7 ilustra um exemplo de coleta de dados com Tshark. Possui a funcionalidade adicional de suporte à análise de arquivos compactados (gzip), extraindo e analisando simultaneamente, quando necessário, utilizando para isso a biblioteca zlib que está incorporada na ferramenta.

Figura 7 - Exemplo de coleta de dados com Tshark.

```
[root@localhost tmp]# tshark -nc 5
Capturing on 'wlp3s0'
 1 0.000000 31.13.85.36 -> 192.168.1.104 TCP 66 443 -> 36992 [ACK] Seq=1 Ack=1 Win=59 Len=0 TSval=3910995682 TSecr=15807992
 2 0.000042 31.13.85.36 -> 192.168.1.104 TLSv1.2 212 Server Hello, Change Cipher Spec, Encrypted Handshake Message
 3 0.000064 192.168.1.104 -> 31.13.85.36 TCP 66 36992 -> 443 [ACK] Seq=1 Ack=147 Win=237 Len=0 TSval=15808027 TSecr=3910995682
 4 0.005959 31.13.85.8 -> 192.168.1.104 TCP 74 80 -> 53154 [SYN, ACK] Seq=0 Ack=1 Win=13980 Len=0 MSS=1410 SACK_PERM=1
   TSval=3802669597 TSecr=15807992 WS=256
 5 0.005989 192.168.1.104 -> 31.13.85.8 TCP 66 53154 -> 80 [ACK] Seq=1 Ack=1 Win=229 Len=0 TSval=15808033 TSecr=3802669597
5 packets captured
```

Fonte: Do autor.

A Figura 7 mostra o Tshark analisando o tráfego que está passando na rede em tempo real, de forma bastante parecida com o tcpdump, vez que ambos utilizam a biblioteca libcap como base.

3.5 Características e identificação de tráfego

Nesta seção vamos abordar características e exemplos sobre identificação de tráfego com problema, onde será utilizado todo conhecimento abordado até o momento.

Na análise do tráfego de uma rede, seja em tempo real capturando e analisando simultaneamente ou em um processo de análise de um arquivo pcap, os dados mais importantes para identificar a origem e o destino são: endereço do IP de origem, endereço do IP de destino, porta de origem, porta de destino e protocolo de transporte.

Possuindo estas cinco informações é possível identificar qual dos hosts envolvidos na comunicação é o cliente e qual é o servidor (KUROSE, ROSS, 2013). Para exemplificar melhor vamos utilizar um exemplo que é bastante comum de ocorrer, que é a perda de conectividade de rede.

A Figura 8 mostra uma perda de conectividade, onde o arquivo começa com quatro pacotes TCP ACK padrão, enviados entre 10.3.71.7 e 10.3.30.1 (SANDERS, 2007).

Figura 8 - Mostra a confirmação de pacotes ACK.

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	10.3.71.7	10.3.30.1	TCP	1043 > 1048 [ACK] Seq=0 Ack=0 Win=8760 Len=0
2	0.000000	10.3.30.1	10.3.71.7	TCP	1048 > 1043 [PSH, ACK] Seq=5840 Ack=0 Win=8760 Len=648
3	0.000000	10.3.71.7	10.3.30.1	TCP	1043 > 1048 [ACK] Seq=0 Ack=2920 Win=8760 Len=D
4	0.000000	10.3.71.7	10.3.30.1	TCP	1043 > 1048 [ACK] Seq=0 Ack=5840 Win=8760 Len=D

Fonte: Adaptado de Sanders, 2007.

O problema inicia no pacote 5, onde aparece a primeira retransmissão de pacotes TCP (Figura 9).

Figura 9 - Retransmissão TCP, sinal que a conexão está fraca ou caiu.

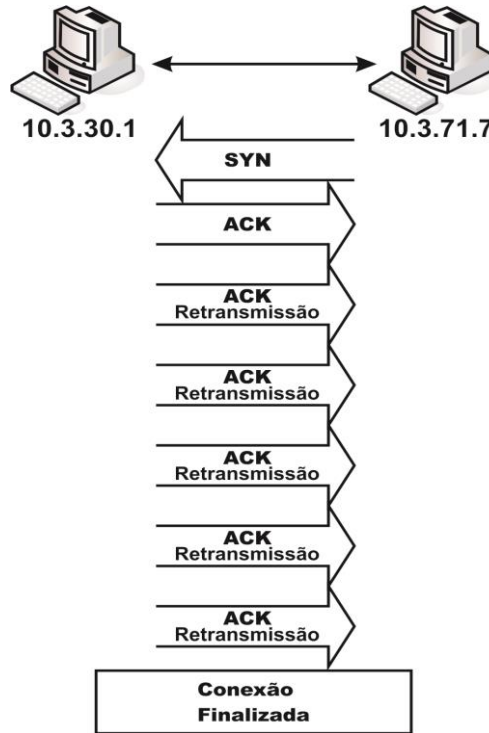
No. -	Time	Source	Destination	Protocol	Info
5	0.20600D	10.3.30.1	10.3.71.7	TCP	[TCP Retransmission] 1048 > 1043 (PSH, ACK) Seq=5840 Ack=0 Win=8760 Len=648
6	0.80600D	10.3.30.1	10.3.71.7	TCP	[TCP Retransmission] 1048 > 1043 (PSH, ACK) Seq=5840 Ack=0 Win=8760 Len=648
7	2.00600D	10.3.30.1	10.3.71.7	TCP	[TCP Retransmission] 1048 > 1043 (PSH, ACK) Seq=5840 Ack=0 Win=8760 Len=648
8	4.40600D	10.3.30.1	10.3.71.7	TCP	[TCP Retransmission] 1048 > 1043 (PSH, ACK) Seq=5840 Ack=0 Win=8760 Len=648
9	9.21100D	10.3.30.1	10.3.71.7	TCP	[TCP Retransmission] 1048 > 1043 (PSH, ACK) Seq=5840 Ack=0 Win=8760 Len=648

Fonte: Adaptado de Sanders, 2007.

Por padrão, quando o TCP envia um pacote para um destino e não obtém uma resposta, ele aguarda um período de tempo especificado, em seguida, retransmite o pacote original. Se uma resposta ainda não é recebida, a fonte de transmissão duplica a

quantidade de tempo que espera por uma resposta antes de enviar outra retransmissão. O conceito de uma retransmissão TCP é ilustrado na Figura 10.

Figura 10 - A ocorrência de retransmissões (problema de conectividade).



Fonte: Adaptado de Sanders, 2007.

A Figura 11 mostra um exemplo onde o processo é repetido durante cinco tentativas até que as retransmissões TCP sejam concluídas. Após a falha de cinco tentativas de retransmissão, a conexão é encerrada pelo protocolo TCP.

Outro caso que podemos citar é o problema de rede lenta, onde que o primeiro passo consiste em identificar a origem do problema. Ao analisarmos o arquivo de captura (Figura 11), a primeira coisa que percebemos são solicitações de pacotes ICMP (ping) enviados do host 24.6.126.218 para um host remoto (SANDERS, 2007).

Figura 11 - Envio de ICMP (ping) do host 24.6.126.218 para um host remoto.

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	24.6.126.218	198.173.244.32	ICMP	Echo (ping) request
2	3.364382	24.6.126.218	198.173.244.32	ICMP	Echo (ping) request
3	6.368126	24.6.126.218	198.173.244.32	ICMP	Echo (ping) request
4	9.371704	24.6.126.218	198.173.244.32	ICMP	Echo (ping) request

Fonte: Sanders, 2007.

Se acessarmos na opção de seção IP do painel *Packet Details* do Wireshark (Figura 12), veremos que estes pacotes diferem dos pacotes de ping regulares, a diferença é o *time-to-live* (TTL, tempo de vida) destes pacotes definido como 1 salto.

Figura 12 - Pacote de ping com um valor de time-to-live de 1 salto.

```

Internet Protocol, Src: 24.6.126.218 (24.6.126.218), Dst: 198.173.244.32 (198.173.244.32)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 92
  Identification: 0xb5f6 (46582)
  Flags: 0x00
  Fragment offset: 0
  Time to live: 1
  Protocol: ICMP (0x01)
  Header checksum: 0xb1fc [correct]
  Source: 24.6.126.218 (24.6.126.218)
  Destination: 198.173.244.32 (198.173.244.32)

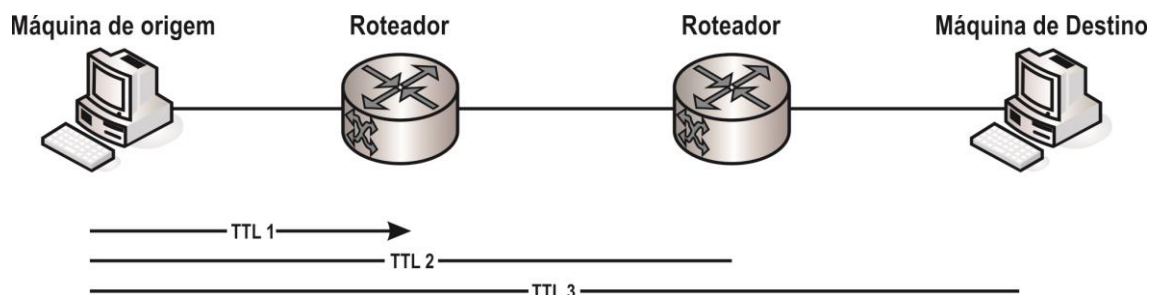
```

Fonte: Sanders, 2007.

O TTL (*time-to-live*) é um valor numérico que determina quantas vezes que um pacote pode saltar de um roteador para outro através de uma rede. Um meio de compreender melhor isto é através do utilitário Traceroute (TRACEROUTE, 2016). O valor 1 habilita o envio de um pacote de uma origem para um destino, porém o pacote irá expirar logo ao atingir o primeiro roteador ao longo da sua rota.

Ao ocorrer isso, o protocolo ICMP entra em ação, disparando uma mensagem notificando que o pacote teve seu TTL expirado antes de chegar ao seu destino final. Ao emissor receber essa mensagem, o mesmo incrementa o TTL do pacote em mais uma unidade. Esse processo então continua até que o pacote obtenha um valor de TTL suficiente para que chegue ao seu destino, de acordo com a Figura 13.

Figura 13 - Mostra o valor de TTL aumentando ao passar por mais redes.



Fonte: Adaptado de Sanders, 2007.

O problema ocorre porque o primeiro pacote enviado com o valor 1 de TTL é atingido no primeiro roteador e esta resposta de volta para o computador de origem não aconteceu.

Como o computador de origem não recebeu a resposta ele aguarda cerca de três segundos conforme Figura 14, e em seguida envia outro pedido.

Figura 14 - Host de origem reenvia solicitação após não receber retorno.

No. -	Time	Source	Destination	Protocol	Info
2	3. 364382	24. 6. 126. 218	198. 173. 244. 32	ICMP	Echo (ping) request
3	6. 368126	24. 6. 126. 218	198. 173. 244. 32	ICMP	Echo (ping) request

Fonte: Adaptado de Sanders, 2007.

Quando o host de origem não recebe qualquer resposta a está segunda tentativa, ele espera cerca de três segundos mais e envia um último pacote para o roteador e também não tem êxito, como mostrado na Figura 15.

Figura 15 - Host de origem faz mais uma tentativa.

No. -	Time	Source	Destination	Protocol	Info
3	6. 368126	24. 6. 126. 218	198. 173. 244. 32	ICMP	Echo (ping) request
4	9. 371704	24. 6. 126. 218	198. 173. 244. 32	ICMP	Echo (ping) request

Fonte: Adaptado de Sanders, 2007.

Neste ponto, Traceroute desiste de receber uma resposta a partir do primeiro roteador, assim que seu próximo pacote de número quatro tem um valor TTL de 2. Esse pacote atinge o segundo roteador com sucesso, e o host de origem recebe o retorno esperado ICMP tipo 11, código 0, pacote com tempo de vida excedido conforme mostrado na Figura 16.

Figura 16 - Tempo de vida do pacote excedido.

No. -	Time	Source	Destination	Protocol	Info
5	9. 393904	12. 244. 25. 161	24. 6. 126. 218	ICMP	Time-to-live exceeded (Time to live exceeded in transit)

Fonte: Adaptado de Sanders, 2007.

Este processo continua através do resto da captura onde o valor TTL é incrementado continuamente até que o destino seja atingido. Neste caso o problema está com roteador interno da nossa rede que nunca foi capaz de enviar uma resposta ICMP.

3.6 Considerações

Através dos estudos realizados neste capítulo, foi possível verificar as principais formas de captura de tráfego de rede, fatores os quais são essenciais para definirmos qual técnica de análise de tráfego será utilizada. Para realizar o procedimento de análise de tráfego de rede, dispomos de algumas ferramentas como Wireshark, Tshark, tcpdump.

Para a detecção e análise proativa de anomalias no tráfego de rede é de suma importância o administrador de rede conhecer o funcionamento dos principais protocolos de rede, além do modelo de camadas dos Protocolos Internet, conhecer a infraestrutura de rede, as ferramentas de análise de tráfego existentes, para com base neste conjunto de fatores definir as formas de captura, técnicas de análise de tráfego de rede para auxiliar na identificação de problemas no tráfego de redes TCP/IP.

4 ESTADO DA ARTE

Neste capítulo é apresentada uma visão geral dos principais trabalhos que compõem o estado da arte relacionado à detecção e análise de anomalias no tráfego de rede. Mais especificamente, a seção 4.1 descreve os trabalhos relacionados ao problema de detecção de anomalias, nos quais trata de técnicas para melhorar a eficiência do procedimento de detecção e classificação de padrões de anomalias, já a seção 4.2 descreve trabalhos relacionados aos aspectos de análise de tráfego de rede, com enfoque na inspeção detalhada no tráfego de rede, assim como técnicas para melhorar o desempenho durante a análise de tráfego. A seção 4.3 descreve a síntese do estado da arte. Por fim, a seção 4.4 discute as diferenças em relação à proposta atual, nas quais serão tratadas pela estratégia definida neste trabalho de conclusão.

4.1 Detecção de anomalias

Existe uma diversidade de técnicas sendo utilizada para a detecção de anomalias no tráfego de rede, como, por exemplo, a teoria da evidência discutida no trabalho proposto por (LINS, FEITOSA, SADOK, 2009), que teve como objetivo o desenvolvimento de uma ferramenta para detecção de anomalias no tráfego de rede utilizando a teoria da evidência, para utilização de diferentes métodos de detecção de anomalias para diminuir a taxa de falsos positivos e negativos maximizando a eficiência da detecção.

A composição da ferramenta proposta pelos autores é dividida em módulos integrados tais como módulo de coleta, sensores e um mecanismo para fusão dos dados. O módulo de coleta é responsável por realizar o monitoramento de tráfego da rede e gerar o arquivo de coleta no formato pcap, posteriormente a análise dos dados é realizada *offline*. Após a coleta dos dados, os sensores realizam toda análise dos dados gerados na etapa de coleta para detectar possíveis anomalias que possam vir a existir no tráfego da rede.

Os sensores também realizam a definição dos padrões para cada uma das induções geradas, a partir do mecanismo de classificação. Finalmente o mecanismo de fusão realiza a tomada de decisão onde faz uso das regras de combinação definidas pela teoria de evidência, onde é relacionada às análises realizadas pelos sensores e fornece como saída inferências mais precisas e com maior grau de exatidão.

Para o procedimento de validação da ferramenta foi estruturado um ambiente em laboratório onde foram realizados alguns testes onde foi injetado tráfego SMTP com o

intuito de criar “SPAM”, para um servidor de email externo. A tabela abaixo demonstra o grau de exatidão de identificação por sensores e também demonstra o percentual da fusão de ambos os sensores resultando no percentual geral de detecção da ferramenta ADS-Fusion.

Tabela 1 - Resultado de fusão para o tráfego SPAM.

Tempo	Profiling	TCPModel	ADS-Fusion
80-100	80%	81%	98,10%
100-120	19%	34%	13,27%
120-140	22%	81%	21,24%
140-160	37%	78%	23,07%
160-180	41%	82%	44,69%

Fonte: Adaptado de Lins, Feitosa, Sadok, 2009.

A tabela demonstra que no intervalo de tempo entre 100-120 a 140-160 é representado o tráfego anômalo e a detecção pelos sensores. Neste caso como um dos sensores não havia identificado o tráfego de “SPAM” à combinação de padrões utilizada pela técnica de teoria da evidência contribui significativamente para identificação do tráfego, na qual atribuiu um valor maior as evidências geradas pelo sensor que conseguiu identificar o desvio no padrão de tráfego caracterizado como anômalo. A contribuição do trabalho apresentado é na utilização da teoria da evidência aplicada como técnica de fusão de dados na qual realizou a avaliação em ambos os sensores onde foi possível identificar como resultado final que se tratava realmente de uma anomalia no tráfego e contribuiu no estabelecimento das regras para avaliação de anomalias, auxiliando assim na detecção de anomalias no tráfego de rede.

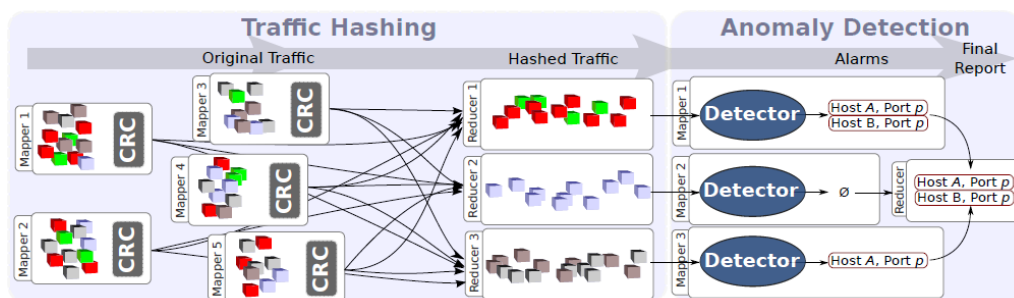
Dentro deste contexto da detecção de anomalias, como no tráfego de “SPAM”, a evolução das técnicas de detecção permitiu integrações com diferentes métodos de detecção, diferentes tipos de arquitetura dentre outros fatores que venha há contribuir na detecção de anomalias. Dentro deste novo conceito temos o trabalho proposto por (FONTUGNE, MAZEL, FUKUDA, 2014), que consiste no desenvolvimento de *framework* MapReduce junto ao Hadoop para auxiliar no processo de detecção de anomalias, onde pode ser utilizado qualquer método de detecção e inclusive combinar vários métodos diferentes para um melhor resultado. Dentro deste novo paradigma

proposto foi utilizado o Hadoop que é responsável por dividir o conjunto de dados em subconjuntos e distribui em um cluster para processamento independente.

O *framework* proposto é chamado de Hashdoop, consiste em programar um ambiente MapReduce que divide o tráfego com uma função hash para preservar estruturas de tráfego e portanto, se beneficiar das infraestruturas de computação distribuída afim de executar vários detectores de anomalias em paralelo, fornecendo assim uma detecção em tempo real. O modelo MapReduce, consiste em um *framework* introduzido pelo Google para suportar computações paralelas em grandes coleções de dados em clusters de computadores. Desta forma o conjunto de dados deve ser dividido em partes para que os detectores computem as estatísticas de tráfego através de estruturas espaciais e temporais. De acordo com Fontugne, Mazel e Fukuda (2014) a opção por utilizar MapReduce junto ao Hadoop foi por fornecer escalonamento e tolerância a falhas que são características cruciais para segurança de Internet.

O funcionamento do *framework* consiste em propor um novo quadro MapReduce onde será composto por duas etapas: primeiro, usando uma função hash na qual divide o tráfego onde ambas as estruturas espaciais e temporais de tráfego sejam preservadas. Em segundo lugar, os detectores identificam as anomalias em cada grupo de dados, posteriormente as anomalias são relatadas aos operadores de rede. A Figura 17 ilustra funcionamento do *framework*.

Figura 17 - Estrutura do framework proposto.



Fonte: Fontugne, Mazel, Fukuda, 2014.

A Figura 17 apresenta o tráfego de rede original onde é realizada a divisão de tráfego com a função hash, utilizando o endereço IP como chave para função hash na qual permite a divisão de tráfego em pequenos pedaços preservando ao mesmo tempo a duração de tempo e do fluxo de consistência. A função hash utiliza o endereço IP de origem e de

destino para garantir que o tráfego enviado ou recebido por um determinado host chegue e até um determinado detector conforme ilustrado na Figura 17. Na etapa de detecção de anomalia, foram utilizados dois detectores de anomalias para o procedimento de mapa, um detector de anomalia baseado na contagem de pacotes, um detector de anomalias de monitoramento de tráfego estacionário.

Foi realizada, para o procedimento de validação a coleta de informações em um arquivo de tráfego capturado em um link trans-pacífico, entre o Japão e EUA. Em experimentos foram analisados traços de tráfego capturado (somente IP e informações de porta) de janeiro a março em 2001, 2004, 2007, 2010 e 2013. Foram utilizados dados nesta captura como fonte de alimentação para o Hadoop.

O *framework* MapReduce proposto em conjunto com o Hadoop apresentou como resultado um melhor desempenho na detecção de anomalias quanto a solução clássica na qual quebrava estruturas de tráfego espaciais e temporais e também deteriorava consideravelmente o desempenho do detector de anomalia.

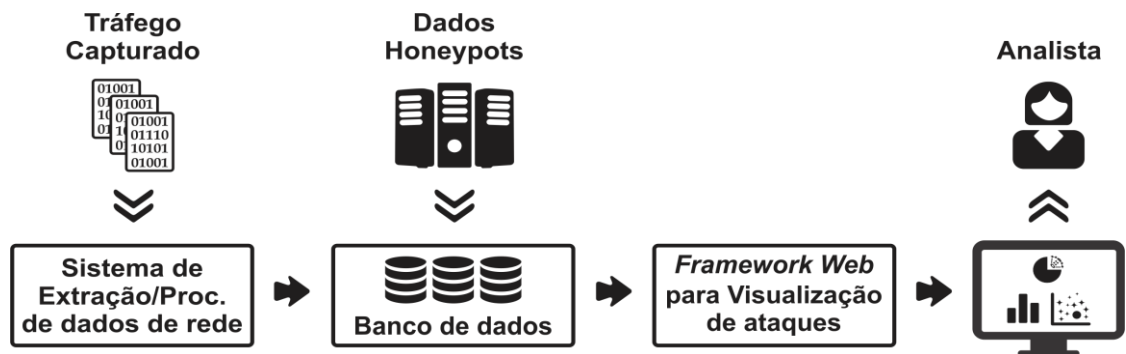
4.2 Análise de tráfego de rede

Em se tratando de análise de tráfego de rede, existem diferentes ferramentas como Wireshark, Tshark, tcpdump, nas quais se disponham a realizar a análise de tráfego da rede, porém para um diagnóstico mais preciso frente a um determinado problema, temos que ir mais afundo na análise de tráfego de rede, como, por exemplo, tráfego de vírus na rede.

Siqueira *et al.* (2015) propôs uma arquitetura para investigar e reconhecer padrões de tráfego de rede gerado durante a execução de um malware, onde utilizando um método de análise comportamental e através de uma inspeção detalhada no conteúdo dos pacotes capturados e trafegados por *malwares*, é gerado estatísticas que posteriormente são apresentadas através de um *framework* web.

O *framework* foi dividido em dois componentes conforme a Figura 18, onde possui um sistema responsável pela extração e processamento de dados e um *framework* web para visualização dinâmica das informações processadas.

Figura 18 - Estrutura da arquitetura proposta.



Fonte: Siqueira *et al.*, 2015.

A Figura 18 mostra a arquitetura utilizada que com base no tráfego capturado, realiza a extração e processamento de dados onde posteriormente passa para a etapa que consulta os dados dos honeypots os quais consistem em recursos computacionais que podem ser comprometidos pelos atacantes. Após ter realizado a extração e processamento de dados e identificado às estruturas que seriam atingidas pelos *malwares* é disponibilizado estes dados para o analista através de *framework web*.

O funcionamento do *framework* inicialmente começa na etapa de coleta de dados onde é realizada a coleta de exemplares de *malwares* (coleta que foi realizada durante o período de um ano) validados junto ao VirusTotal (VIRUS TOTAL, 2016) através de detecção por engines de antivírus. Posteriormente, foi aplicado em ambiente controlado, e todo o tráfego gerado pela rede foi capturado e armazenado em arquivos pcap analisados pela arquitetura proposta.

A análise do tráfego de rede foi feita através da ferramenta Bro (BRO, 2016), onde todo o tráfego é transformado em uma série de eventos em alto nível, como por exemplo, uma conexão a um site ou transferência de arquivo. Estes eventos são tratados por um *script* também desenvolvido no trabalho que permite a inspeção aprofundada no tráfego de rede e também a identificação de ataques e outras atividades maliciosas, armazenando resultados em um banco de dados.

As estatísticas dos ataques são disponibilizados através do *framework web* desenvolvido, onde permite, entre outros fatores, estimar a posição geográfica do ponto de ataque. Desta forma, podem ser observadas as tendências dos tipos de ataques no decorrer do dia com base nos protocolos, serviços e locais de origem. Por fim, as estatísticas são disponibilizadas através de uma interface web, para uma melhor avaliação do analista.

Tabela 2 - Estatística da coleta nos honeypots.

Serviços	Porta	Total de ataques	IPs únicos
FTP	21	654 (0,03%)	381 (1,49%)
HTTP	80	31.001 (1,35%)	4.319 (16,94%)
RPC	135	369 (0,02%)	89 (0,35%)
SMB	445	2.153.774 (94,12%)	15.659 (61,41%)
MS-SQL	1433	18.665 (0,82%)	1.292 (5,08%)
SIP	5060	83.771 (3,66%)	3.757 (14,73%)
	Total	2.288.234 (100%)	25.497 (100%)

Fonte: Adaptado de Siqueira *et al.*, 2015.

A Tabela 2 mostra as portas e o total de ataques sofridos, assim como o número de IPs associados às atividades maliciosas referentes a ataques de *pharming* e *phishing*. Nestes tipos de ataques, as vítimas são levadas a um domínio malicioso acreditando ter acessado algum serviço legítimo. A análise de tráfego realizada de forma dinâmica apresentou resultados positivos, quanto a tendência dos ataques do tipo *pharming* e *phishing*, onde foi possível realizar estatísticas, bastantes precisas quanto a estes ataques.

A análise detalhada de tráfego de rede, seja ela em tempo real capturando e analisando simultaneamente ou em processo de análise de um arquivo pcap *offline*, permite identificarmos pontos de gargalo na rede, e nos permite realizar otimizações no fluxo de tráfego de rede. Dentro deste contexto o trabalho proposto por (BREMLER *et al.*, 2014) apresenta o desenvolvimento de um algoritmo para realizar a análise detalhada de pacotes DPI (*Deep Packet Inspection*), de forma centralizada, que extrai o motor DPI dos diferentes *middleboxes* para fornecê-los como serviço para os vários *middleboxes* na rede.

Os autores (BREMLER *et al.*, 2014) partem da premissa de que atualmente o tráfego é inspecionado a partir do zero por todos os *middleboxes* presentes no caminho. O trabalho então propõe que o tráfego seja inspecionado somente uma vez contra os dados de todos os *middleboxes* da rede. Este tipo de serviço é fornecido por meio da implementação de um DPI controlador centralizado que gerência os demais DPI presentes nos *middleboxes* na rede.

Desta forma um pacote na rede irá passar somente por uma instância de serviço DPI e posteriormente passará para os demais *middleboxes* de acordo com sua política estabelecida. Desta forma um pacote não precisa ser verificado novamente, uma vez que já foi verificado ao passar pelo DPI controlador no qual fornece os resultados da

verificação junto ao pacote original. Após receber o resultado do serviço de DPI, cada *middlebox* aplica as regras correspondentes para os padrões conforme sua lógica interna.

O algoritmo primeiramente realiza a combinação dos vários conjuntos de padrões, originados através de diferentes *middleboxes* onde cada pacote é digitalizado e recebe um identificador único. Após isso, o padrão é registrado pelo *middlebox*, utilizando a metodologia baseada em conhecimento.

Após finalizada a etapa anterior é realizada a construção do algoritmo que é constituído com base em um autômato finito determinístico, onde é definido um conjunto padrão que é extraído do motor do DPI dos *middleboxes*, onde cada segmento possui um estado de aceitação. A segunda etapa consiste em armazenar em uma matriz as definições realizadas na etapa anterior, fornecendo como saída uma lista ordenada com os padrões identificados.

A terceira etapa consiste em armazenar em uma tabela o mapeamento entre uma política, que consiste na definição do identificador da cadeia e os *middleboxes* correspondentes. Em outra tabela é realizado a definição das propriedades que dizem respeito ao critério de parada. Os pacotes devem ser comparados com os padrões definidos, para determinar quais conjuntos são aplicáveis para cada pacote.

Finalmente é verificado se o fluxo corresponde a algum padrão. Caso corresponda os dados são repassados para os *middleboxes* para realizar o procedimento de atualização dos padrões. A implementação foi realizada através de uma máquina virtual com o Mininet (MININET, 2016), em uma Rede Definida por Software (SDN) onde foi utilizada uma topologia básica que consiste em dois hosts de usuários, dois hosts *middlebox* e uma DPI central que pode utilizar múltiplos núcleos. A instância de serviço DPI experimental recebe os padrões, e caso algum pacote corresponda, o DPI marca o pacote para o *middlebox* identificar que ocorreu alguma alteração.

A validação do algoritmo foi realizada com sucesso, pois o algoritmo diminuiu significativamente o processamento nos *middleboxes* uma vez que a inspeção de pacotes era feita do zero a partir de todos estes equipamentos e através do algoritmo foi disponibilizada de forma central sendo inspecionada somente uma única vez. Como resultado apresentou um ganho de 86% mais rápido do que o processo sem a utilização do algoritmo de gerenciamento e controle de padrões baseado na análise de tráfego.

Os métodos utilizados para realizar análise de tráfego, possuem um papel determinante quanto à eficiência e desempenho durante análise, pois podem contribuir de forma positiva ou negativamente no procedimento de identificação do fluxo de tráfego de rede. Com base nessa premissa (BISOL *et al.*, 2016), propôs a implementação de um novo algoritmo de seleção de características de padrões para análise de tráfego de rede, o *Sequential backward Selection* (SBS). Este algoritmo, qual é analisado e comparado com estratégias baseadas em algoritmos genéticos e análise de componente principal (*Principal Component Analysis* - PCA) e também é proposta a utilização de técnicas de seleção de características para aprimorar a precisão de dois algoritmos de classificação de fluxo de tráfego, SVM e K-means.

O método utilizado para comparação com o algoritmo proposto, consiste na seleção de características, é chamado de análise do componente principal. Este método avalia a correlação existente entre as diferentes variáveis de um problema, neste contexto seriam as características dos fluxos de tráfego, criando os chamados Componentes Principais.

O segundo algoritmo utilizado para comparação é o algoritmo genético, no qual realiza uma busca por heurística no espaço através da combinação de soluções (*crossover*) e inserção de novas características as soluções (*mutation*), efetuando assim o processo de evolução a cada geração. Este algoritmo é composto pelas seguintes características: uma população inicial, onde cada indivíduo pode ser uma possível solução para o problema, uma função de avaliação para determinar a qualidade de cada indivíduo e a quantidade de gerações que devem ser executadas.

O primeiro algoritmo proposto *Sequential Backard Selection* (SBS), consiste em eliminar as caraterísticas que interferem na classificação, onde é tomado como ponto de partida um subconjunto no qual é avaliada a qualidade deste subconjunto com cada característica eliminando as mesmas uma a uma. O segundo algoritmo proposto para realizar a classificação de fluxos de tráfegos é o K-means, no qual é baseado em aprendizado não supervisionado, e possui como foco de utilização processos de clusterização.

Para realizar a validação da solução proposta foi desenvolvido um ambiente de avaliação composto por uma topologia em árvore com vinte switches, sessenta e quatro hosts, utilizando o emulador Mininet (MININET, 2016), e um controlador Foodlight (FLOODLIGHT, 2016). Os fluxos gerados na rede são de quatro tipos diferentes: um

servidor HTTP é alvo de ataques DDoS (*Distributed Denial of Service*) partindo de determinados hosts, um servidor de stream de vídeo alocado em um host onde os demais hosts também utilizam este serviço, trocas de arquivos entre hosts por FTP, geração de tráfego através do scrapy (SCAPY, 2016).

Os experimentos foram realizados em três ambientes diferentes, nas quais utilizaram os quatro perfis de tráfego. A Tabela 3 ilustra a proporção do volume de tráfego para cada cenário de teste.

Tabela 3 - Proporção de volume de tráfego para cada cenário de teste.

	Ataques DDoS	Tráfego FTP	Stream de Vídeo	Background
Cenário A	10%	10%	50%	30%
Cenário B	30%	10%	40%	20%
Cenário C	50%	7,5%	25%	17,5%

Fonte: Adaptado de Bisol *et al.*, 2016.

Para cada cenário foi utilizado um total de quinze mil amostras para treinamento e duas mil e quinhentas amostras para testes, assim como para clusterização realizada pelo K-means.

4.3 Síntese do estado da arte

Com o intuito de sintetizar a discussão do estado da arte foi elaborada uma tabela com as informações essenciais das propostas discutidas neste capítulo. A Tabela 5, exibida na próxima página mostra o comparativo entre os trabalhos relacionados estudados. O objetivo da tabela ilustrada é demonstrar os pontos de interesse de cada trabalho que servem como inspiração para a construção da solução do referido trabalho de conclusão de curso.

Tabela 4 - Comparação das diferentes características dos trabalhos relacionados estudados.

Trabalho relacionado	Objetivo	Técnica de Análise Utilizada	Metodologia de detecção
Aplicando a teoria da evidência na detecção de anomalias (LINS, FEITOSA, SADOK, 2009)	Desenvolvimento de uma ferramenta para detecção de anomalias no tráfego de rede utilizando a técnica da teoria da evidência.	Teoria da evidência	Detecção baseada em comportamento
Uma Arquitetura para Análise e Visualização de Tráfego de Rede Malicioso (SIQUEIRA <i>et al.</i> , 2015)	Propôs uma arquitetura para investigar e reconhecer padrões de tráfego de rede gerado durante execução de um malware.	Análise e inspeção de pacotes	Detecção baseada em comportamento
Hashdoop: A MapReduce Framework for Network Anomaly Detection (FONTUGNE, MAZEL, FUKUDA, 2014)	Desenvolvimento de um novo quadro para ser utilizado no <i>framework</i> MapReduce junto ao Hadoop para melhorar o desempenho no processo de detecção de anomalias no tráfego de <i>backbone</i> da Internet.	Análise estatística de pacotes	Detecção baseada em comportamento
Deep Packet Inspection as a Service (BREMLER <i>et al.</i> , 2014)	Desenvolvimento de um algoritmo para realizar a análise detalhada de pacotes (DPI), de forma centralizada.	Análise e inspeção de pacotes	Detecção baseada em conhecimento
Coleta e Análise de Características de Fluxo para Classificação de Tráfego em Redes Definidas por Software (BISOL <i>et al.</i> , 2016)	Implementação de um novo algoritmo de seleção de características, o <i>Sequential backward Selection</i> (SBS), junto há técnicas de seleção de características para aprimorar a precisão de dois algoritmos de classificação de fluxo de tráfego, SVM e K-means.	Algoritmo SVM (Support Vector Machine), Algoritmo K-means	Detecção baseada em comportamento e conhecimento

Fonte: Do autor.

4.4 Diferenças em relação à proposta atual

No contexto de detecção e análise de anomalias no tráfego de rede, podem ser identificadas duas diferenças principais. Primeiro, as abordagens de (FONTUGNE, MAZEL, FUKUDA, 2014), (LINS, FEITOSA, SADOK, 2009), assim como as propostas por (BISOL *et al.*, 2016), (BREMLER *et al.*, 2014), (SIQUEIRA *et al.*, 2015), atacam a detecção e análise de anomalias no tráfego de rede com foco em identificar possíveis anomalias oriundas da rede de larga escala (redes WAN), enquanto que o contexto de redes locais (redes LAN) concentra propostas para lidar com ataques ou fluxos de tráfego maliciosos. Não se propõem a detectar e analisar anomalias dentro de redes locais, nas quais pode haver falhas de configurações, defeito em equipamentos dentre outros fatores quais podem acabar inserindo tráfego indevido na rede, provocando congestionamento e o colapso da rede com um todo. A segunda limitação surge pelo modo em que a análise e detecção de anomalias são desempenhadas na literatura, no caso através de operações *offline*.

A análise e detecção de anomalias em modo *offline* limitam os benefícios que podem ser obtidos pelas soluções propostas, vez que são soluções aplicadas sobre traços coletados no passado da infraestrutura de rede. Desta forma, uma extensão diante destas propostas busca fornecer a análise e detecção de anomalias no tráfego de redes locais (LAN) na qual permita a análise em tempo real.

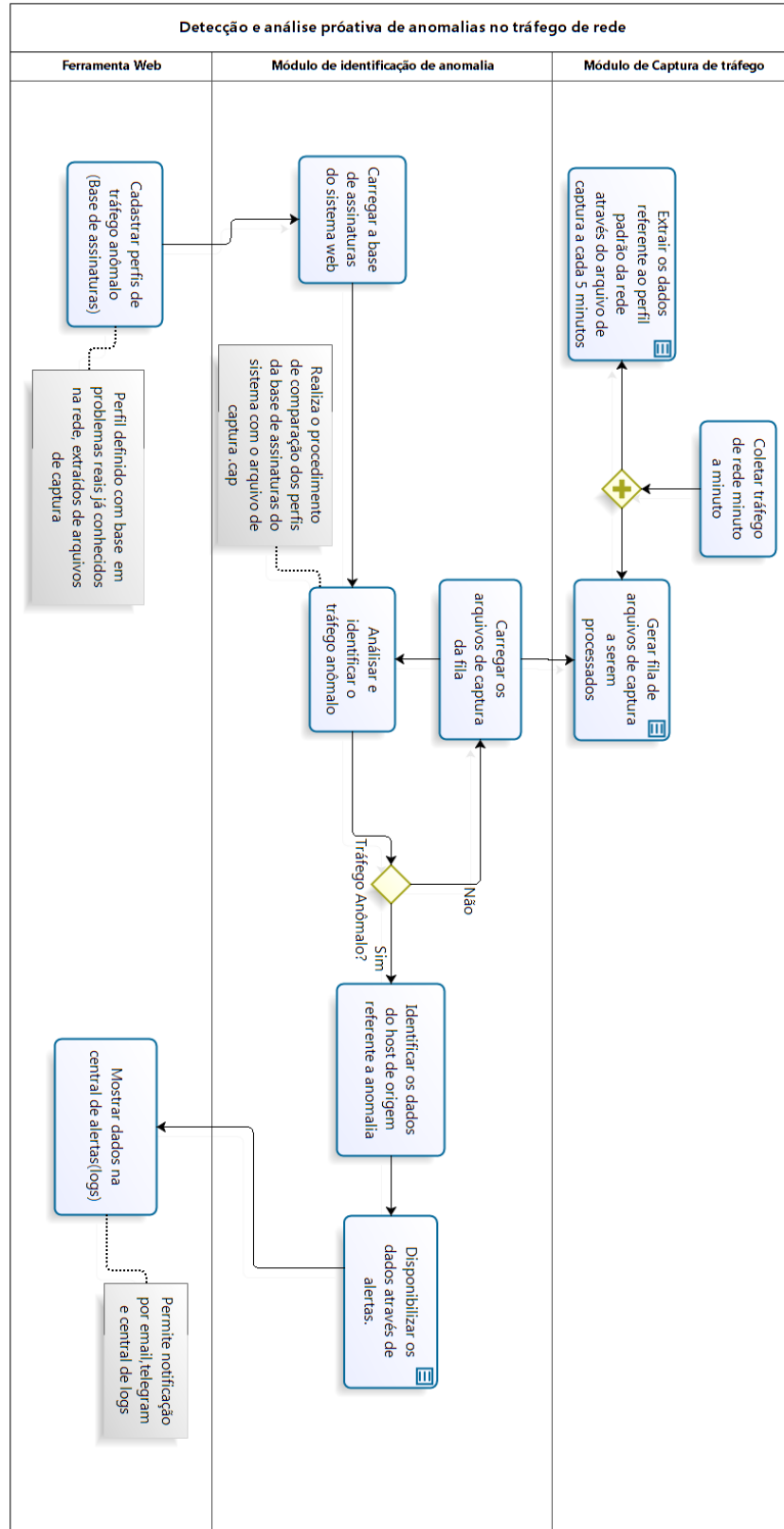
5 SISTEMA DE DETECÇÃO PROATIVA DE ANOMALIAS NO TRÁFEGO DA REDE LOCAL

Neste capítulo será apresentado o sistema de detecção proativa de anomalias no tráfego da rede local. Além disso, serão apresentados todos os aspectos funcionais da ferramenta, destacando, casos de uso, experimentos realizados, cenário de avaliações e resultados obtidos.

5.1 Funcionamento da ferramenta

O sistema desenvolvido consiste na detecção e análise proativa de anomalias no tráfego de rede local. Tal objetivo foi alcançado através da análise de pacotes e o emprego da metodologia baseada em conhecimento (NADEEM; HOWARTH, 2013), permitindo a identificação do ponto de origem de um determinado problema (anomalia). O sistema é dividido em três módulos: módulo de captura de tráfego, módulo de identificação de anomalia e definição das assinaturas na ferramenta web. A Figura 19, mostra a arquitetura de funcionamento do sistema.

Figura 19 - Fluxograma do funcionamento do sistema.

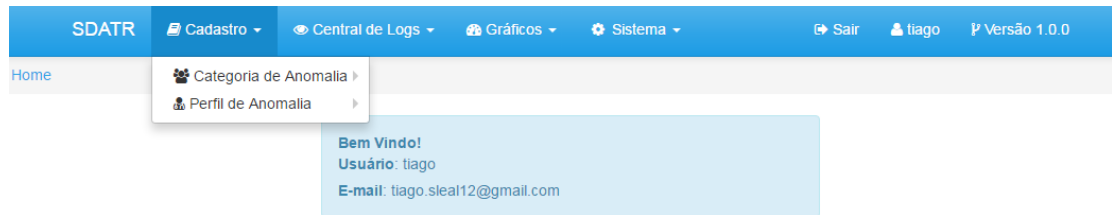


Fonte: Do autor.

Ferramenta Web

A interface do sistema web desenvolvido é composta por quatro menus principais: cadastro, central de logs, gráficos e sistema.

Figura 20 - Menu de cadastros do sistema.



Fonte: Do autor.

Inicialmente, temos o menu de cadastro, onde dispomos das opções: cadastrar novas categorias e criar novos perfis de anomalias. Este menu é um dos mais importantes do sistema pois, é nele que realizamos a criação da categoria de anomalia na qual posteriormente, é vinculado o perfil de comportamento da anomalia. A Figura 21 exibe a listagem das categorias definidas neste trabalho.

Figura 21 - Listagem categorias de anomalias.

Código	Descrição	Ativo	Data Cadastro	Opções
1	Rede lenta	1	14/04/2017 11:12	✎ 🗑
2	Conflito DHCP	1	14/04/2017 11:12	✎ 🗑
3	Loop na Rede	1	14/04/2017 11:12	✎ 🗑
4	Erro de CRC	1	14/04/2017 11:12	✎ 🗑
5	ip duplicado na rede	1	23/04/2017 15:00	✎ 🗑

Fonte: Do autor.

Depois de realizada a criação da categoria de anomalia, é preciso definir o perfil de anomalia que consiste em definir as assinaturas do perfil. As assinaturas são criadas utilizando os filtros da ferramenta Tshark (WIRESHARK, 2016). A Figura 22 apresenta a tela de cadastro de perfil de anomalias.

Figura 22 - Cadastro de perfil de anomalia.

Perfil de Anomalia

Categoria de Anomalia:
 Erro de CRC

Prioridade de Execução:
 4

Identificador:
 erro_de_crc

Assinatura 1:
 tshark -nr /tmp/coleta/#ARQUIVO -Y
 'tcp.analysis.retransmission' | grep 'Retransmission'
 | wc -l

Assinatura 2:
 tshark -nr /tmp/coleta/#ARQUIVO -T fields -e
 arp.src.proto_ipv4 -e arp.src.hw_mac -e
 arp.dst.proto_ipv4 -e arp.dst.hw_mac -Y 'eth.type ==
 0x0806 && eth.fcs_bad' | awk '{ print \$1,\$2,\$3,\$4}'

Assinatura 3:

Cadastrar **Cancelar**

Fonte: Do autor.

Para definição do perfil de anomalia, inicialmente seleciona-se a qual categoria nosso perfil se refere, posteriormente temos a opção de definir uma prioridade de execução que consiste na ordem que será realizado o processo de identificação da anomalia, como, por exemplo, a prioridade 1 possui maior relevância frente as demais por isso executa por primeiro, enquanto as demais seguem a ordem natural. Um exemplo de perfil com prioridade 1 é o *loop* na rede, pois devido ao alto número de pacotes que deve ser processado, já que em alguns casos pode levar a total inoperância da rede.






Após temos um campo identificador que é gerado automático pelo sistema com base na categoria selecionada, este campo é responsável por identificar a assinatura no módulo de identificação de anomalia. Por fim, aparecem os campos que se referem à criação das assinaturas. Um aspecto importante na definição destas assinaturas é a utilização da expressão “curinga” #ARQUIVO, que será substituído no processo de identificação pelo nome do arquivo corrente que esta sendo processado.

Uma vez criado o perfil temos de acessar a listagem de perfis para realizar o procedimento de ativação do perfil, que consiste em clicar na opção ativar perfil conforme Figura 23. Após a ativação do perfil, o mesmo vai ficar disponível no módulo de identificação de anomalia.

Figura 23 - Ativação perfil de anomalia.

Listagem Perfil de Anomalias

Excel PDF Copy CSV Print Ocultar Colunas Pesquisar:

Categoria	Identificador	Assinatura1	Assinatura2	Assinatura3	Prioridade	Opções
Loop na Rede	loop_na_rede	capinfos -izyx /tmp/colela/#ARQUIVO tail -n +2 cut -d : -f 2	tshark -nr /tmp/colela/#ARQUIVO -z endpoints.ip -q tail -n +5 awk '{print \$1,\$2}' head -10		1	
Rede lenta	rede_lenta	tshark -nr /tmp/colela/#ARQUIVO -T fields -e frame.number -e ip.src -e eth.src -e _ws.col.info -Y 'tcp.analysis.window_update' awk '{ print \$2,\$3}'	tshark -nr /tmp/colela/#ARQUIVO -T fields -e ip.dst -e eth.dst -Y 'frame.len == 1514 && tcp.analysis.initial_rtt > 0.1 && tcp.analysis.lost_segment' awk '{print \$1,\$2}'		2	
ip duplicado na rede	ip_duplicado_na_rede	tshark -nr /tmp/colela/#ARQUIVO -T fields -e arp.src.proto_ipv4 -e arp.src.hw_mac -Y 'arp.duplicate-address-detected' awk '{ print \$1,\$2}' uniq			3	
Erro de CRC	erro_de_crc	tshark -nr /tmp/colela/#ARQUIVO -Y 'tcp.analysis.retransmission' grep 'Retransmission' wc -l	tshark -nr /tmp/colela/#ARQUIVO -T fields -e arp.src.proto_ipv4 -e arp.src.hw_mac -e arp.dst.proto_ipv4 -e arp.dst.hw_mac -Y 'eth.type == 0x0806 && eth.fcs_bad' awk '{ print \$1,\$2,\$3,\$4}'		4	
Conflito DHCP	conflito_dhcp	tshark -r /tmp/colela/#ARQUIVO -Y 'bootp.option.type==53 && bootp.option.value==6 && udp.srcport == 67' -T fields -e ip.src -e eth.src -e bootp.id awk '{ print \$1,\$2}'			5	

Mostrando de 1 até 5 de 5 registros Anterior Próximo

Fonte: Do autor.

A tela de listagem dos perfis permite a manutenção dos perfis criados tais como: editar as informações de um determinado perfil, alterar a prioridade de execução, alterar os dados referente a base de assinaturas e também possui a opção de ativar/inativar o perfil, opção bastante interessante, vez que os perfis inativos não serão carregados no processo de identificação de anomalia.

A segunda opção no menu do sistema consiste na central de logs do sistema na qual toda e qualquer anomalia identificada irá aparecer, desde que seu perfil esteja definido no sistema. Estes logs também são enviados por email e Telegram (TELEGRAM, 2017). A Figura 24 abaixo mostra os alertas enviados pela central de logs.

Figura 24 - Alertas disponibilizados pelo sistema.

The screenshot shows the 'Central de Logs' interface. At the top, there are buttons for 'Excel', 'PDF', 'Copy', 'Print', 'CSV', and 'Ocultar Colunas'. A search bar labeled 'Pesquisar:' is on the right. Below this is a 'Mensagem' section with a 'Data hora' dropdown. The main content area displays a 'Conflito de DHCP na rede' alert with the following details: IP: 192.168.50.1 | MAC: 00:e0:20:2b:06:e5 and IP: 192.168.0.1 | MAC: e8:de:27:55:03:2b, dated 17/05/2017 20:41:45. Below the alert, there is a section for 'sulvale.testes@gmail.com' (20:41 (Há 2 horas)) with options to 'Traduzir mensagem' and 'Desativar para: inglês'. A 'Log Email' button is also present. On the right, a 'Log Telegram' section shows the same alert details.

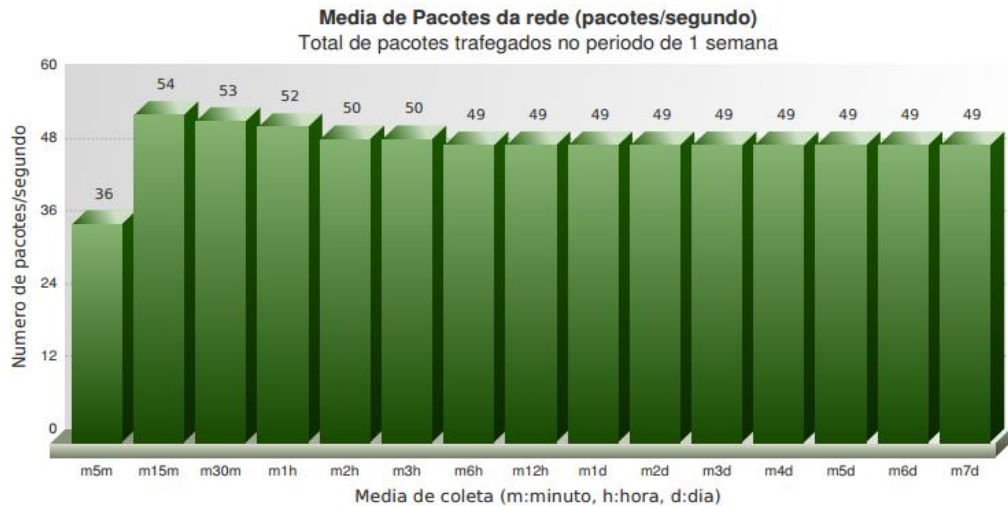
Fonte: Do autor.

A terceira opção no menu do sistema chamada de gráficos permite realizar a análise de tráfego da rede em tempo real, onde temos o número de pacotes por segundo, latência média da rede e total de pacotes retransmitidos.

Para geração dos gráficos foi realizado os cálculos da média de pacotes por segundo, latência da rede e pacotes retransmitidos, onde a cada cinco minutos de coleta, são extraídos esses dados da rede e o tempo (em *timestamp*) referente a essas coletas, ambos os dados são armazenados no banco de dados Mysql (MYSQL, 2017). Posteriormente é realizado o cálculo da média para cada gráfico que consistem em somar os valores armazenados no banco de dados separadamente referente a cada um dos três casos, de acordo com o seu respectivo tempo de coleta e divide-se pelos intervalos de tempo pré-determinados (m5m, m15m, m30m, m1h, m2h, m3h, m6h, m12h, m1d, m2d, m3d, m4d, m5d, m6d, m7d).

Para auxiliar no entendimento vamos considerar o gráfico da Figura 25, com os seguintes valores: na coleta de cinco minutos o valor 36; na coleta de dez minutos o valor 88; na coleta de quinze minutos o valor 38. Para realizarmos a média referente a coleta realizada a quinze minutos pegamos os valores armazenados nas coletas de cinco, dez e quinze minutos e dividimos pelo número de coletas que neste caso seria três, totalizando assim o valor de 54 no período de quinze minutos (m15m) como ilustrado no gráfico da Figura 25, referente a média de pacotes por segundo trafegado na rede. Este mesmo procedimento é realizado para os outros intervalos de tempo e também para os casos de latência da rede e pacotes retransmitidos.

Figura 25 - Gráfico média de pacotes/segundo na rede.

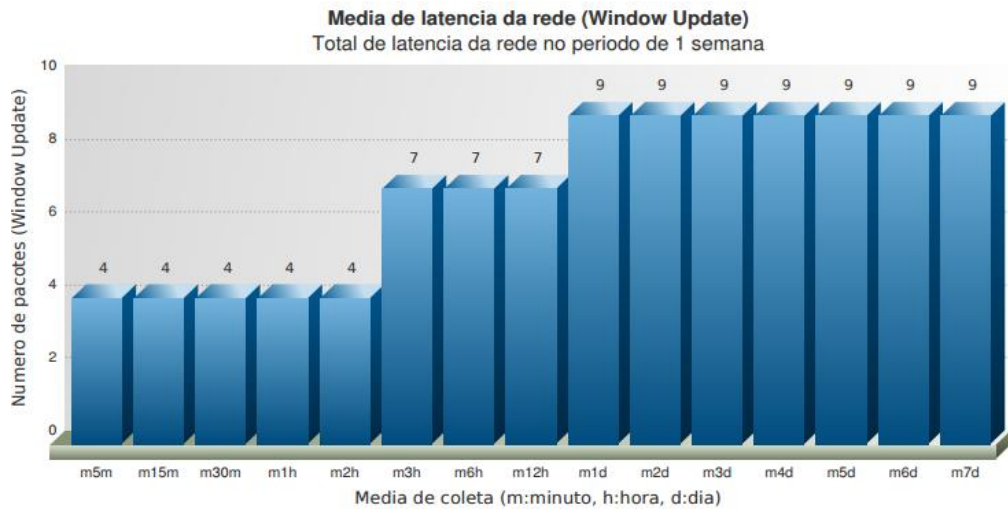


Fonte: Do autor.

O conjunto de dados disponibilizados conforme Figura 25 permite identificarmos como está a média de pacotes trafegados na rede. Esta métrica é utilizada para auxiliar o módulo de identificação de anomalia, na aplicação das assinaturas (base de conhecimento) quanto ao perfil relacionado a *loop* na rede.

A Figura 26 ilustra o gráfico referente à média de latência da rede (*window update*), que significa que está sempre sendo necessário aumentar o tamanho da janela de transmissão de dados e por consequência disso acaba abrindo uma nova janela de transmissão deixando assim a rede lenta. O gráfico mostra o número de pacotes em que ocorreu este caso, assim como o seu período de captura. Essa métrica é utilizada para auxiliar o módulo de identificação de anomalia na aplicação das assinaturas referente ao perfil de rede lenta (sentida no acesso a Internet pelo usuário final).

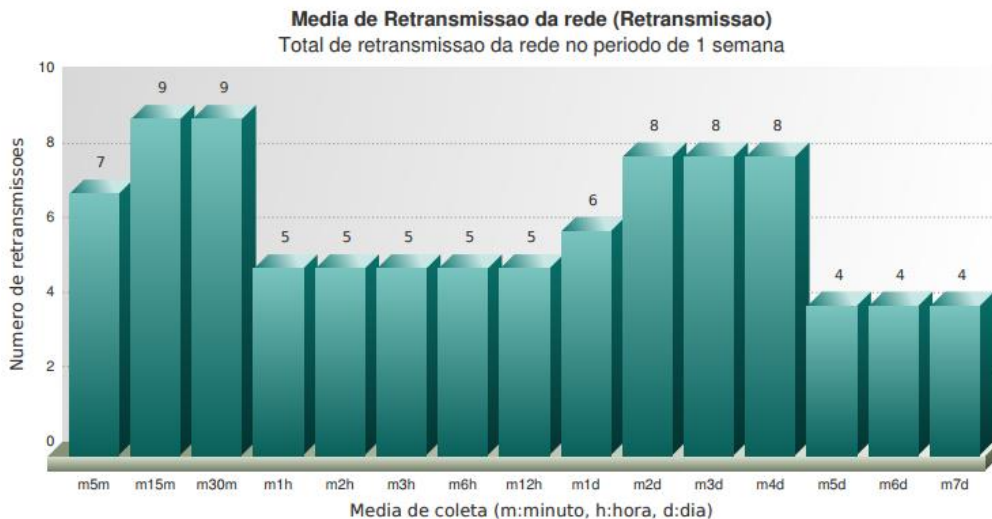
Figura 26 - Gráfico latência média da rede.



Fonte: Do autor.

Posteriormente temos a Figura 27 que apresenta o gráfico referente à média de pacotes retransmitidos na rede, que significa que o pacote não está conseguindo chegar ao seu destino por algum motivo (meio físico inadequado, servidores sobrecarregados, aplicação sobrecarregada, etc.) e não consegue receber o retorno do host de destino por isso fica sempre tentando estabelecer uma nova conexão. No geral devido a oscilação no tráfego de rede sempre haverá um percentual mínimo de retransmissão, porém caso este valor cresça exponencialmente pode ser um indicio que temos algum problema na rede. Essa métrica é utilizada para auxiliar o módulo de identificação de anomalia na aplicação das assinaturas referente aos perfis de rede lenta e erro de CRC (problema físico) na rede.

Figura 27 - Gráfico média de pacotes retransmitidos na rede.



Fonte: Do autor.

A quarta opção no menu da ferramenta chamada de Sistema permite a criação de novos usuários, assim como a manutenção dos usuários existentes, troca de senha, ativar e desativar usuários. A Figura 28 ilustra o cadastro de um novo usuário no sistema.

Figura 28 - Cadastro de usuário no sistema.

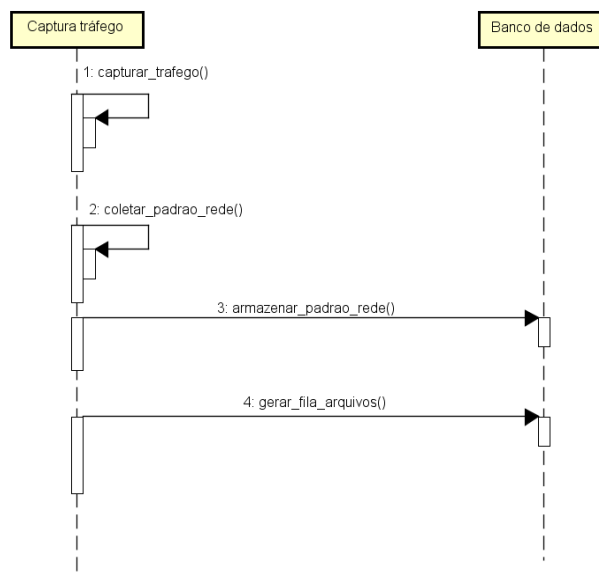
The screenshot shows a web interface for user registration. At the top, there is a navigation bar with the following items: SDATR, Cadastro, Central de Logs, Gráficos, Sistema, Sair, Tiago, and Versão 1.0.0. Below the navigation bar, a dropdown menu is open under the 'Sistema' menu item, showing options: Alterar Senha, Cadastro, Editar, and Pesquisa. The 'Cadastro' option is highlighted. Below the dropdown menu, there is a form titled 'Informe seus dados' with the following fields: Nome (Tiago Leal), Usuário (tiago), E-mail (tiago.sleal12@gmail.com), Senha (masked with dots), and Confirmação de Senha (masked with dots). At the bottom of the form, there are two buttons: 'Cadastrar' and 'Cancelar'.

Fonte: Do autor.

Módulo de captura de tráfego

O módulo de captura é composto por três etapas: (i) captura de tráfego minuto-a-minuto, (ii) extração do perfil normal da rede e (iii) geração da fila de arquivos a serem analisados. A Figura 29 mostra o fluxo detalhado da rotina de captura de tráfego e geração dos arquivos a serem processados.

Figura 29 – Diagrama de sequência coleta de tráfego.

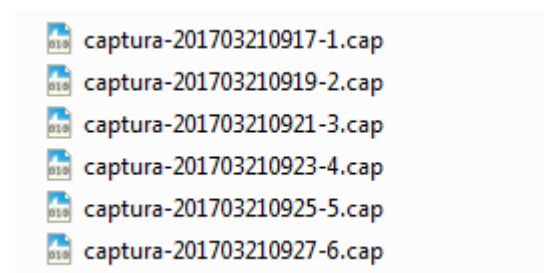


Fonte: Do autor.

Inicialmente temos a primeira etapa de captura de tráfego de rede em tempo real na qual consiste em ficar capturando tráfego de dados de forma contínua de minuto-a-minuto, na segunda etapa a cada 5 minutos são extraídos dados estatísticos quanto ao padrão normal da rede, tais como: número de pacotes, número de retransmissão, média de latência. Estes valores são armazenados no banco de dados, onde é mantido um histórico médio dos valores dessas métricas em minutos que posteriormente são agrupadas em horas e depois em dias durante o período de uma semana. Essas informações são necessárias para que tenhamos conhecimento de como é o padrão da nossa rede, para que quando ocorra alguma situação fora do padrão conhecido possa ser realizado o procedimento de análise e identificação de tráfego anômalo.

Após extraídos os dados estatísticos e finalizado a geração do arquivo de captura, chegamos à terceira etapa onde o arquivo é movido para um diretório chamado de coletas e, posteriormente, é realizado um escaneamento neste diretório para geração da fila de processamento dos arquivos de captura qual gravamos em banco de dados Mysql (MYSQL, 2017). Este procedimento se faz necessário, pois em casos como o de *loop* na rede o tamanho do arquivo fica grande e como os processos de captura e identificação de anomalia são distintos, podemos estar realizando a checagem das assinaturas em um determinado arquivo e paralelo a isso a captura de um novo arquivo. Os arquivos de captura são nomeados conforme o intervalo de tempo de captura, a Figura 30 ilustra o formato padrão dos arquivos gerados.

Figura 30 - Formato do arquivo de captura gerado.



Fonte: Do autor.

Módulo de Identificação de Anomalia

O módulo de identificação de anomalia é responsável por carregar a base de assinaturas definida no cadastro de perfil realizado através da ferramenta web. A base de assinaturas foi gerada a partir dos arquivos de captura com base em anomalias ocorridas

em ambientes reais (por razões de confidencialidade não estou autorizado a citar os nomes das empresas). Além disso, a base de assinaturas desenvolvida também foi validada frente ao dataset do Wireshark University (WIRESHARK, 2017), e ao final foram realizadas simulações inclusive em ambiente produção. Este processo utiliza metodologia baseada em conhecimento, na qual possui como principal objetivo diminuir a taxa de falsos positivos (LARI, AMARAL, 2004).

Após, carregada à base de assinaturas, no módulo de identificação podemos realizar a definição de cada assinatura, para este procedimento temos o campo identificador que foi gerado automaticamente pelo sistema web ao realizarmos o cadastro das assinaturas. A Figura 31 ilustra a definição da lógica de processamento de uma assinatura.

Figura 31 - Definição da assinatura de identificação de anomalia.

```

if ident_assinatura == "ip_duplicado_na_rede":
    print("=====")
    print("Ip Duplicado na rede")
    print("=====")

    cmd = assinatura1
    ret = os.popen(cmd)

    mat_dados = []
    conta = 0

    for index,lista_ip in enumerate(ret):
        ip_mac_src = (lista_ip.split(" "))
        controle = ip_mac_src in mat_dados

        if controle == False:#somente se nao esta na lista insere
            mat_dados.append(ip_mac_src)
            conta +=1

    if conta > 0:
        strMsg = ""
        strMsg = "IP(s) duplicado(s) na rede:\n\n"

        for linha1 in mat_dados:
            ip_src = linha1[0]
            mac_src = linha1[1]

            strMsg+= "IP: "+ip_src+" | MAC: "+mac_src+"\n"

        sislog(strMsg); #insere os logs no banco
    else:
        print ("Sem IP duplicado na rede\n")

    #atualiza fila
    update_query = ""
    update_query = ("update system_monitor_arquivo_fila set status='1' where id='%d';" % id_arquivo_fila)
    cursor.execute(update_query)
    conexao.commit()

#atualiza status final da fila
update_query = ""
update_query = ("update system_monitor_arquivo set status='1' where id='%d';" % id_arquivo)
cursor.execute(update_query)
conexao.commit()

```

Fonte: Do autor.

Para realizar a definição das assinaturas basta informar o identificador e empregar a lógica necessária que irá permitir a identificação da anomalia.

Posteriormente é carregada a fila com a ordem de processamento dos arquivos de captura que consiste em comparar os perfis de tráfego (base de assinaturas) do sistema

com os arquivos de captura, obtendo como resultado a identificação de existência ou não de uma anomalia.

Caso não seja identificada a ocorrência de uma anomalia é realizado o procedimento de leitura de um novo arquivo de captura, ao qual já estará disponível para análise, pois o fluxo de coleta fica capturando tráfego de modo contínuo. Caso seja identificada a ocorrência de uma anomalia são capturados os dados como endereço IP e MAC de origem e destino, e após é disparado um alerta para a ferramenta web na qual permite notificação por email, Telegram (TELEGRAM, 2017) e central de logs da ferramenta web.

5.2 Ambiente de avaliação

Para realizar o procedimento de avaliação da ferramenta foi optado por simulação real em laboratório, ao invés do emprego de simuladores (Core, NS2, NS3, etc) para obtermos um diagnóstico mais preciso em um cenário real uma vez que, depois de validada a ferramenta foi implantada em ambiente real de produção nas empresas C, F, M, P e X com sede na região de Santa Cruz do Sul e Vera Cruz.

5.3 Especificação do ambiente

O ambiente de homologação em laboratório possui: 10 computadores, 4 servidores, 1 switch, 2 links de Internet, 2 roteadores. Após finalizada a homologação da ferramenta em laboratório a mesma foi colocada em produção nas empresas C, F, M, P e X.

Tabela 5 - Especificação dos ambientes de avaliação.

Ambiente	Nº Computadores	Nº Servidores	Nº Switch	Nº de Internet	Outros equipamentos
Empresa C	150	10	5	2 links	5 hub 10 impressoras
Empresa F	40	3	2	2 links	6 roteadores 5 impressoras
Empresa M	200	20	5	3 links	10 coletores 15 impressoras 20 telefones voip
Empresa P	15	2	1	2 links	6 roteadores 2 impressoras
Empresa X	300	25	12	2 links	24 impressoras

Fonte: Do autor.

5.4 Premissas

A ferramenta proposta adota inicialmente duas premissas. Considera que todas as redes seguem o padrão IPv4. E, além disso, os cenários avaliados e considerados neste trabalho se tratam de redes cabeadas, padrão Ethernet, redes sem fio estão fora de escopo deste trabalho.

Datasets

Para desenvolvimento deste trabalho foram utilizados dois datasets, sendo o primeiro extraído a partir dos arquivos de captura com base em anomalias ocorridas nos ambientes reais estudados e o segundo dataset do Wireshark University (Wireshark, 2017). O primeiro foi utilizado para estudo dos cenários de: conflito de DHCP, IP duplicado, erro de CRC (físico), rede lenta (Internet) e *loop* na rede. Já o segundo dataset, foi adotado para os cenários de: rede lenta (Internet), erro de CRC (físico), conflito de DHCP.

Para verificarmos um conflito de DHCP, que consistem em dois servidores de DHCP oferecendo endereço IP para um determinado host, primeiramente temos de identificar como é o tráfego de rede de uma solicitação normal de DHCP. A Figura 32 mostra o fluxo correto desta solicitação.

Figura 32 - Tráfego normal solicitação de DHCP.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xaa755fe5
2	0.067227000	73.121.224.1	24.6.170.101	DHCP	346	DHCP Offer - Transaction ID 0xaa755fe5
3	0.069614000	0.0.0.0	255.255.255.255	DHCP	349	DHCP Request - Transaction ID 0xaa755fe5
4	0.084862000	73.121.224.1	24.6.170.101	DHCP	346	DHCP ACK - Transaction ID 0xaa755fe5

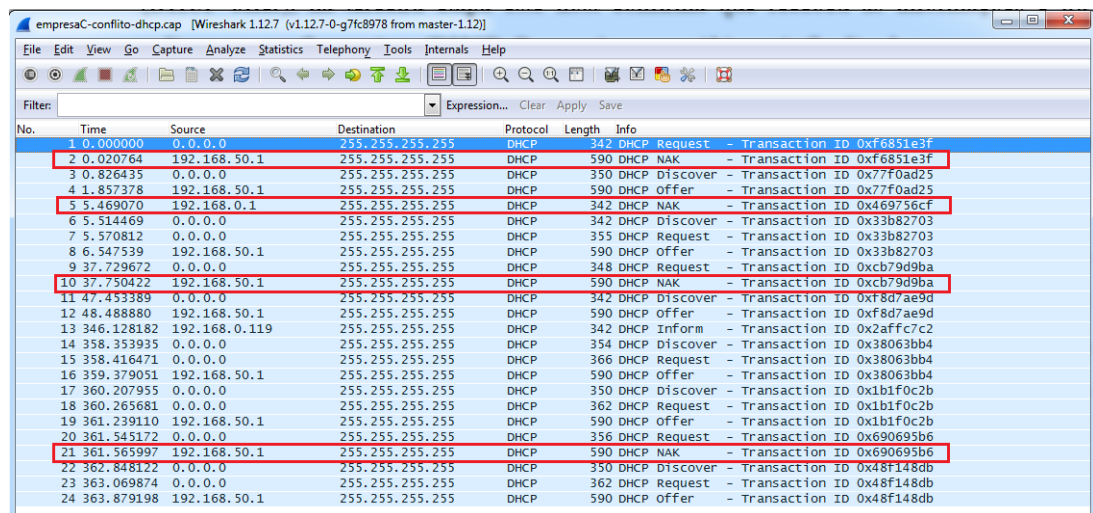
Fonte: Chappel, 2012.

Inicialmente temos a primeira etapa onde o host cliente (0.0.0.0), envia uma solicitação na rede para localizar o servidor de DHCP disponível (DHCP *Discover*) na rede, na segunda etapa o servidor oferece os parâmetros de configuração para host cliente (DHCP *offer*), na terceira etapa este host confirma que recebeu as informações e irá aplicar as configurações (DHCP *Request*) e por último é finalizado com sucesso a operação através da mensagem (DHCP *ACK*). Este arquivo esta disponível no diretório do

sistema desenvolvido no seguinte caminho: *sdatr/dataset/dhcp_normal.cap* (WIRESHARK, 2017).

Após conhecer o procedimento normal de uma solicitação DHCP na rede, podemos partir para o caso de conflito de DHCP. Este caso foi extraído de ambiente real na empresa C, no qual consiste em um computador solicitando IP e dois servidores de DHCP respondendo e o host em questão não conseguindo carregar as configurações de nenhum dos dois servidores de DHCP. A Figura 33 a seguir mostra a captura correspondente a este caso.

Figura 33 - Tráfego conflito de DHCP.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xf6851e3f
2	0.020764	192.168.50.1	255.255.255.255	DHCP	590	DHCP NAK - Transaction ID 0xf6851e3f
3	0.026435	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0x77f0ad25
4	1.857378	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x77f0ad25
5	5.469070	192.168.0.1	255.255.255.255	DHCP	342	DHCP NAK - Transaction ID 0x469756cf
6	5.514469	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x33b82703
7	5.570812	0.0.0.0	255.255.255.255	DHCP	355	DHCP Request - Transaction ID 0x33b82703
8	6.547539	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x33b82703
9	37.729672	0.0.0.0	255.255.255.255	DHCP	348	DHCP Request - Transaction ID 0xcb79d9ba
10	37.750422	192.168.50.1	255.255.255.255	DHCP	590	DHCP NAK - Transaction ID 0xcb79d9ba
11	47.453389	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf8d7ae9d
12	48.488880	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0xf8d7ae9d
13	346.128182	192.168.0.119	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x2affc7c2
14	358.353935	0.0.0.0	255.255.255.255	DHCP	354	DHCP Discover - Transaction ID 0x38063bb4
15	358.416471	0.0.0.0	255.255.255.255	DHCP	366	DHCP Request - Transaction ID 0x38063bb4
16	359.379051	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x38063bb4
17	360.207955	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0x1b1f0c2b
18	360.265681	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 0x1b1f0c2b
19	361.239110	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x1b1f0c2b
20	361.545172	0.0.0.0	255.255.255.255	DHCP	356	DHCP Request - Transaction ID 0x690695b6
21	361.565997	192.168.50.1	255.255.255.255	DHCP	590	DHCP NAK - Transaction ID 0x690695b6
22	362.848122	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0x48f148db
23	363.069874	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 0x48f148db
24	363.879198	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x48f148db

Fonte: Do autor.

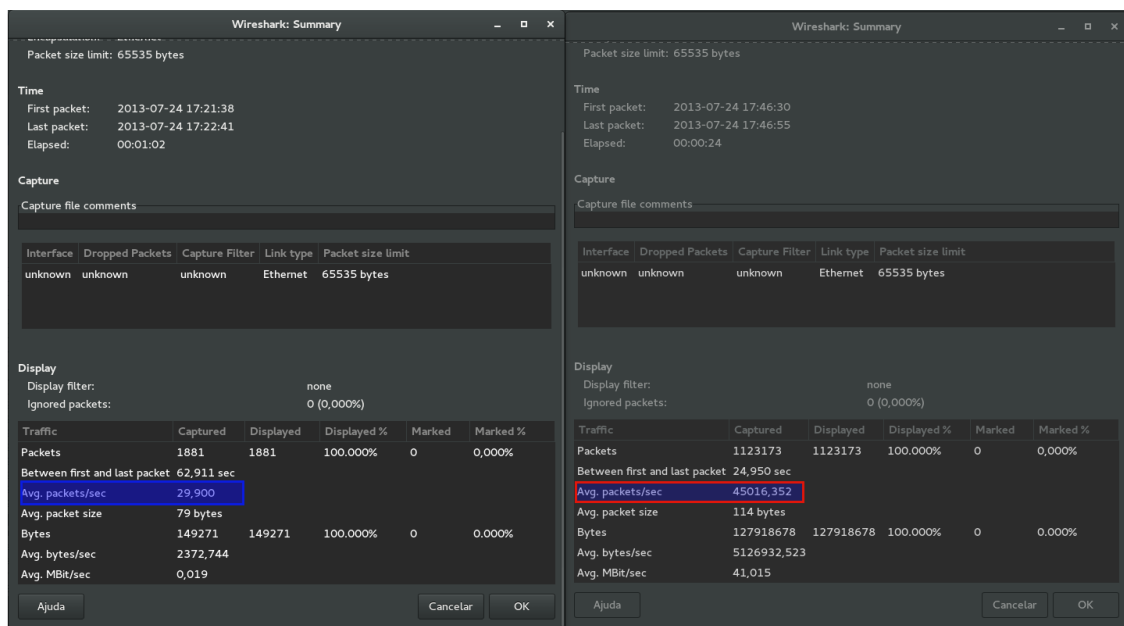
O tráfego coletado acima mostra uma máquina realizando a solicitação de IP na rede e recebendo como retorno uma mensagem do tipo (DHCP NAK), que consiste em endereço de rede incorreto. Este arquivo está disponível no diretório do sistema desenvolvido no seguinte caminho: *sdatr/dataset/empresaC-conflito-dhcp.cap*.

O segundo dataset foi extraído de ambiente real referente a empresa M que consiste em identificar IP duplicado na rede, onde devido a um problema no servidor de DHCP, o mesmo começou a distribuir endereços IP que já estavam sendo utilizados, ocasionando assim o erro de conflito de IP. Este arquivo está disponível no diretório do sistema desenvolvido no seguinte caminho: *sdatr/dataset/empresaM-ip-duplicado.cap*.

O terceiro dataset foi extraído de ambiente real referente, à empresa X que consiste em identificar um *loop* na rede. A métrica que precisa ser analisada para este caso é a alta taxa de pacotes por segundo transmitida e para isso é necessário conhecermos a média normal

de pacotes transmitidos pela rede (ARAGON; COMBS; CHAPPELL, 2014). A Figura 34 a seguir mostra um exemplo do número de pacotes normal de uma rede em azul e o alto número de pacote com o *loop* em execução em vermelho. Não foi possível disponibilizar este dataset na pasta do sistema, pois continha dados relevantes da empresa X (o que iria requerer um processo de ofuscação dos dados não previsto neste trabalho).

Figura 34 - Número de pacotes transmitidos na rede (*loop* na rede).



Fonte: Do autor.

O quarto dataset consiste em verificar erro de CRC na rede, para isto foi utilizado o dataset do (WIRESHARK, 2017), que pode ser encontrado através do endereço (<https://wiresharkbook.com/troubleshooting.html>) ou no caminho informado abaixo pelo nome (*tr-checksums.cap*), além disso também foi utilizado um dataset que foi extraído de um ambiente real referente a empresa I. Estes arquivos estão disponíveis no diretório do sistema desenvolvido no seguinte caminho: *sdatr/dataset/tr-checksums.cap*, *empresal-erro-crc.cap*.

O quinto dataset consiste em identificar uma rede lenta (sentida no acesso a Internet pelo usuário final), efeito do congestionamento da rede. Para este caso temos de verificar a alta quantidade de pacotes retransmitidos na rede, assim como um alto índice de “TCP window update”, que significa que está sempre sendo necessário aumentar o tamanho da janela de transmissão de dados e por consequência disso acaba abrindo uma nova janela de transmissão e isso leva ao atraso na confirmação (ACK) de resposta, deixando assim a

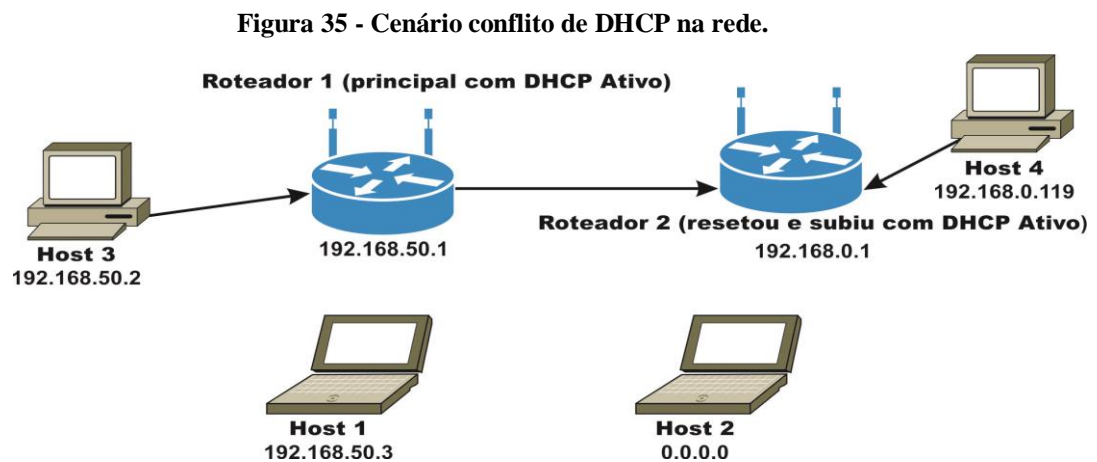
rede lenta (CHAPPELL, 2012). Este caso foi extraído do dataset da (WIRESHARK, 2014) e também retirado de ambiente real referente à empresa T. Este arquivo está disponível no diretório do sistema desenvolvido no seguinte caminho: *sdatr/dataset/empresaT-rede-lenta.cap*.

5.5 Cenário de avaliação

Nesta etapa de avaliação, serão considerados alguns casos de anomalias no tráfego de rede decorrentes em ambientes reais, nas quais via análise de tráfego, foi possível identificar a origem do problema.

São analisados os seguintes cenários: (1) conflito de servidores DHCP, (2) *loop* na rede, (3) IP duplicado na rede, (4) erro de CRC e (5) rede lenta (Internet).

1 - Conflito de DHCP na rede:



Fonte: Do autor.

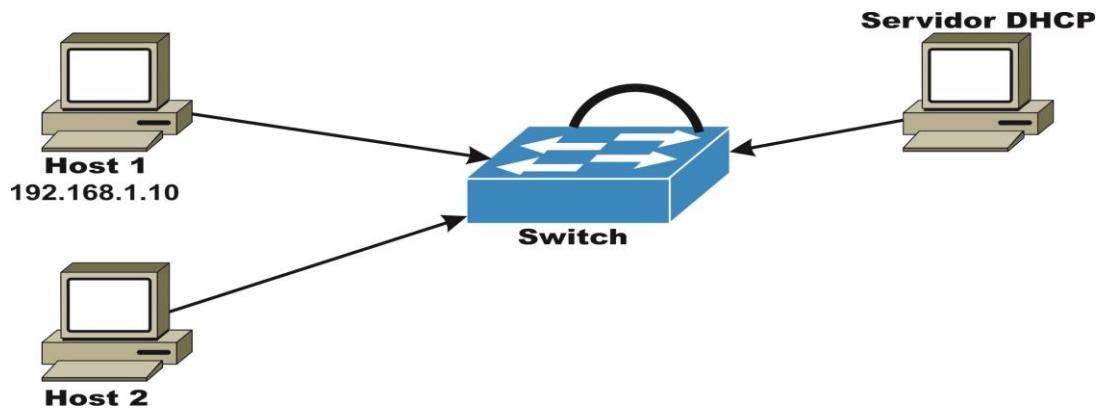
A Figura 35 mostra um problema ocorrido no roteador 2 que devido a uma falha de hardware e/ou software acabou por resetar a configuração padrão de fábrica que vem com o DHCP ativo em outra faixa. Por exemplo, quando os hosts 2 e 4 fazem uma solicitação de IP por DHCP na rede, logo os dois roteadores se apresentam como servidores de DHCP, acarretando em um elevado volume de tráfego de rede que acaba deixando a rede lenta, comprometendo o desempenho.

Ainda neste mesmo cenário, pode ocorrer em alguns casos, como o host 4 receber o endereço IP correspondente a outra faixa, pois o roteador 2 estava fornecendo uma faixa de IP diferente da faixa disponibilizada pelo roteador principal, inviabilizando assim a

comunicação destes hosts com a Internet e serviços disponibilizados dentro da infraestrutura de rede.

2 - Loop na rede:

Figura 36 - Cenário de *loop* na rede.



Fonte: Do autor.

A Figura 36 ilustra um exemplo de *loop* na rede onde havia um cabo com uma extremidade conectada ao switch e a outra extremidade não, por engano alguém pegou a extremidade que não estava conectada no switch e conectou ocasionando assim um *loop* na rede. Neste exemplo temos duas máquinas ligadas a um switch, onde o host 1 possui o IP 192.168.1.10 e o host 2 está com endereçamento IP automático e solicita um IP, essa solicitação é enviada por *broadcast* para todas as portas do switch que possuem hosts conectados, o servidor DHCP responde com um IP válido por *broadcast* para todas as portas deste switch, e o cabo com as duas extremidades conectadas ao switch fica replicando a requisição *broadcast* na rede entrando em um *loop* no qual não tem como sair, a menos que seja identificado o problema rapidamente a rede vai sobrecarregar chegando assim a total inoperância da rede.

3 - IP duplicado na rede: Ao realizar a configuração de IP de um host foi atribuído manualmente um endereço de IP já existente na rede.

4 - Erro de CRC: Um host começa a demonstrar problemas de conectividade com a rede de origem física.

5 - Rede lenta: A rede começa a ficar bastante lenta, apartir de um determinado período do dia.

5.6 Experimentos

Com base nos cenários detalhados, serão realizados os seguintes experimentos:

Considerando o cenário 1: Para simular este caso foi utilizado dois roteadores onde será realizada a configuração de forma equivocada do DHCP em ambos os roteadores, afim de realizar o conflito de DHCP na rede.

Figura 37 - Tráfego referente ao conflito de DHCP do cenário 1.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.119	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x2affc7c2
2	12.225753	0.0.0.0	255.255.255.255	DHCP	354	DHCP Discover - Transaction ID 0x38063bb4
3	12.288289	0.0.0.0	255.255.255.255	DHCP	366	DHCP Request - Transaction ID 0x38063bb4
4	13.250869	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x38063bb4
5	14.079773	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0x1b1f0c2b
6	14.137499	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 0x1b1f0c2b
7	15.110928	192.168.50.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0x1b1f0c2b
8	15.416990	0.0.0.0	255.255.255.255	DHCP	356	DHCP Request - Transaction ID 0x690695b6
9	15.417304	192.168.0.1	255.255.255.255	DHCP	342	DHCP NAK - Transaction ID 0x690695b6
10	15.437815	192.168.50.1	255.255.255.255	DHCP	590	DHCP NAK - Transaction ID 0x690695b6
11	16.719940	0.0.0.0	255.255.255.255	DHCP	350	DHCP Discover - Transaction ID 0x48f148db
12	16.941692	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request - Transaction ID 0x48f148db
13	17.751016	192.168.50.1	255.255.255.255	DHCP	590	DHCP offer - Transaction ID 0x48f148db

Fonte: Do autor.

A Figura 37 representa o tráfego capturado referente ao problema descrito no cenário 1, no qual podemos identificar um host realizando uma solicitação de IP na rede e os dois servidores de DHCP respondendo. Esta identificação é realizada através do retorno da mensagem DHCP NAK, na qual permite identificar a existência de dois servidores de DHCP ativo na mesma rede, porém indicando endereço de rede incorreto ao host. Neste caso o DHCP NAK é considerado como uma assinatura na ferramenta web referente a dois servidores de DHCP respondendo a uma mesma solicitação de um host. A assinatura utilizada foi `tshark -r /tmp/coleta/#ARQUIVO -Y 'bootp.option.type==53 && bootp.option.value==6 && udp.srcport == 67' -T fields -e ip.src -e eth.src -e bootp.id | awk '{ print $1,$2}'` onde é possível identificar três filtros: para identificação do protocolo DHCP foi utilizado o `bootp.option.type==53`; para verificar o DHCP NAK o filtro `bootp.option.value==6`; e para identificar a comunicação do DHCP o filtro `udp.srcport == 67` que consiste na porta de comunicação padrão utilizada pelo protocolo.

Após definir a assinatura é iniciado o módulo de identificação de anomalias e começado a realização dos testes, onde primeiramente foi validado contra um dataset gerado em laboratório no qual conseguimos identificar o conflito de DHCP rapidamente. Além disso, também foi validado frente a um dataset extraído de ambiente real onde ocorreu este problema. A Figura 38 mostra o exato momento que o módulo de identificação reconhece a anomalia.

Figura 38 - Identificação em tempo real do conflito de DHCP.

```

=====
DHCP Duplo na rede
=====
tshark -r /tmp/coleta/empresaC-conflito-dhcp.cap -Y 'bootp.option.type==53 && bootp.option.val
ue==6 && udp.srcport == 67' -T fields -e ip.src -e eth.src -e bootp.id | awk '{ print $1,$2}'
Running as user "root" and group "root". This could be dangerous.
Encontrado Conflito de DHCP na rede!
IP_SRC:192.168.50.1 MAC_SRC:00:e0:20:2b:06:e5

IP_SRC:192.168.0.1 MAC_SRC:e8:de:27:55:03:2b

```

Fonte: Do autor.

Considerando o cenário 2: para simular o caso de *loop* na rede foi utilizado dois roteadores, onde foi colocado um cabo ligado do roteador 1 para o roteador 2 e depois outro cabo do roteador 2 para o roteador 1 fechando assim um *loop* na rede. Para identificação deste problema devemos analisar a alta ocorrência de pacotes por segundo. Por exemplo, a média de tráfego normal referente aos testes realizados ficou entre 2-7 pacotes por segundo, enquanto que ao fechar o *loop* na rede este valor cresce exponencialmente chegando a 1447 pacotes por segundo no intervalo de 60 segundos de captura. Vale ressaltar que em ambiente de produção dependendo do tamanho da rede este valor é extremamente alto. A primeira assinatura utilizada foi *capinfos -izyx /tmp/coleta/#ARQUIVO |tail -n +2 | cut -d: -f 2* que permite identificar o número de pacotes por segundo tráfego na rede.

Por fim, é realizado o processo de identificação do ponto de origem (endereço IP de origem e número de pacotes) que gerou esse alto tráfego na rede. A segunda assinatura criada foi *tshark -nr /tmp/coleta/#ARQUIVO -z endpoints,ip -q |tail -n +5 | awk '{print \$1,\$2}' | head -10* na qual temos o filtro *-z endpoints,ip -q;* que retorna quais são os IPs que estão gerando este alto tráfego de dados.

Após criada a assinatura é realizado o procedimento de definição da métrica para o módulo de identificação de anomalia. Esta métrica consiste em verificar se a média de pacotes por segundo contido na captura atual é maior que a média que temos conhecimento dentro do período de uma semana, o qual foi gerado através das coletas de monitoramento realizadas para sabermos o padrão médio de pacotes na rede. Caso a média de pacotes por segundo do tráfego atual multiplicado por 3 (valor atribuído através de experimentos empíricos realizados) seja maior que o padrão da rede, é executada uma segunda assinatura para verificar quais são os IPs que estão gerando alto tráfego na rede.

Posteriormente, foi realizado o procedimento de coleta e análise de anomalia. O tamanho do arquivo gerado durante o *loop* na rede foi cerca de 210 MB, o módulo de identificação levou cerca de 3 minutos para identificar o problema devido ao tamanho do arquivo. Vale ressaltar que a interface gráfica travou devido a limitações físicas da placa de rede, porém os *scripts* de captura e identificação continuaram executando em *background* e conseguiram identificar está anomalia conforme o log da Figura 39 capturado em tempo real.

Figura 39 - Identificação em tempo real do *loop* na rede.

```

Loop na rede identificado - Top Hosts
capinfos -izyx /tmp/coleta/empresaD-loop-na-rede.cap | tail -n +2 | cut -d : -f 2
tshark -nr /tmp/coleta/empresaD-loop-na-rede.cap -z endpoints,ip -q | tail -n +5 | awk '{print $1,$2}' | head -10
IP                NUM PACOTE
192.168.0.3        32685
239.255.255.250   28681
224.0.0.251       18505
192.168.0.102     8713
255.255.255.255   5025
192.168.0.111    4222
192.168.0.1       4009
192.168.0.105    1339
0.0.0.0           1016
192.168.0.108    44
Fora do padrao normal da rede

```

Fonte: Do autor.

Considerando o cenário 3: para simular este caso de IP duplicado na rede foi utilizado um roteador com o DHCP ativo, distribuindo endereços IP na rede. A assinatura utilizada foi `tshark -nr /tmp/coleta/#ARQUIVO -T fields -e arp.src.proto_ipv4 -e arp.src.hw_mac -Y 'arp.duplicate-address-detected' | awk '{ print $1,$2}' | uniq` onde é possível identificar o filtro `arp.duplicate-address-detected`; no qual consiste em retornar a ocorrência de conflito de IP na rede.

Após definir a assinatura é iniciado o módulo de captura de tráfego e identificação de anomalias. Com os módulos ativos foi realizada a configuração manual de um host com um endereço IP que já estava sendo utilizado na rede, ocasionando assim o erro de IP duplicado na rede, abaixo temos a Figura 40 ilustrando o momento que a identificação é realizada.

Figura 40 - Identificação em tempo real de IP duplicado na rede.

```

=====
Ip Duplicado na rede
=====
tshark -nr /tmp/coleta/empresaM-ip-duplicado.cap -T fields -e arp.src.proto_ipv4 -e arp.src.hw_mac -Y
Running as user "root" and group "root". This could be dangerous.
IP(s) duplicado(s) na rede
IP: 192.168.1.210 | MAC: 00:0a:f7:44:2e:10
IP: 192.168.3.249 | MAC: c0:38:96:a1:6b:e5

```

Fonte: Do autor.

Considerando o cenário 4: para simular este caso de erro de CRC será realizado o procedimento de fazer um “chicote” de um cabo de rede junto a um cabo de força, que após determinado tempo de funcionamento começa a gerar problema de conectividade, problema identificado em ambiente real conforme Figura 41.

Figura 41 - Interferência do cabo de força junto ao cabo de rede extraída de ambiente real.



Fonte: Do autor.

Para verificar este caso, temos de identificar a ocorrência de retransmissão acima do normal trafegado na rede, para isso foi definida a seguinte assinatura `tshark -nr /tmp/coleta/#ARQUIVO -Y 'tcp.analysis.retransmission' | grep 'Retransmission' | wc -l` onde temos o filtro `tcp.analysis.retransmission;` para a identificação de retransmissões na rede.

Após é realizada a verificação quanto à ocorrência de erro de CRC na rede mais precisamente na camada de enlace no protocolo ARP, pois caso esteja ocorrendo este erro a nível da camada de rede ou transporte através dos protocolos IP, TCP, UDP pode significar simplesmente que a ferramenta Wireshark (Tshark), não conseguiu realizar o cálculo do *checksum* podendo gerar assim um falso positivo (Chappel, 2014).

A segunda assinatura criada foi `tshark -nr /tmp/coleta/#ARQUIVO -T fields -e arp.src.proto_ipv4 -e arp.src.hw_mac -e arp.dst.proto_ipv4 -e arp.dst.hw_mac -Y`

'eth.type == 0x0806 && eth.fcs_bad' | awk '{print \$1,\$2,\$3,\$4}' onde é possível identificar dois filtros: para identificar a camada ethernet foi utilizado *eth.type == 0x0806*; e para verificar a existência de erro de CRC no protocolo ARP o filtro *eth.fcs_bad*;

Após, criada a assinatura é realizado o procedimento de definição da métrica para o módulo de identificação de anomalia. Esta métrica consiste em verificar se o número de retransmissões contido na captura atual é maior que o número que temos conhecimento dentro do período de uma semana, o qual foi gerado através das coletas de monitoramento realizadas para sabermos o padrão de retransmissões da rede. Caso o número de retransmissões da captura atual seja maior que o padrão conhecido da rede, é executado a segunda assinatura que é responsável por verificar a existência de erros de CRC na rede.

Depois de definida as métricas começa o processo de coleta e análise onde devemos identificar qual host está gerando e/ou recebendo erro de CRC na rede, pois no caso do host estar gerando como ponto de origem, o problema pode ser na placa de rede, porém se estiver somente recebendo com erro, o problema pode estar na porta do switch e caso esteja gerando e recebendo com erro ao mesmo tempo, o problema pode ser no meio físico (cabo de rede). A Figura 42 mostra o log da identificação sendo realizada.

Figura 42 - Identificação em tempo real de erro de CRC na rede.

```

=====
Erro de CRC(Problema Fisico)
=====
tshark -nr /tmp/coleta/empresaI-erro-crc.cap -Y 'tcp.analysis.retransmission' | grep 'Retransmission' | wc -l
vl_retransmissao_normal: 9
vl_retransmissao_captura: 3689
tshark -nr /tmp/coleta/empresaI-erro-crc.cap -T fields -e arp.src.proto_ipv4 -e arp.src.hw_mac -e
arp.dst.proto_ipv4 -e arp.dst.hw_mac -Y 'eth.type == 0x0806 && eth.fcs_bad' | awk '{ print $1,$2,$3,$4}'
Identificado Erro de CRC na rede

IP_SRC:192.168.10.180 MAC_SRC:78:3a:84:05:3e:f2 => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.10.104 MAC_SRC:bc:4c:c4:77:f8:0b => IP_DST:192.168.10.1 MAC_DST:00:00:00:00:00:00
IP_SRC:192.168.10.191 MAC_SRC:f8:cf:c5:bc:6f:8a => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.10.60 MAC_SRC:28:83:35:aa:7d:25 => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.10.109 MAC_SRC:88:79:7e:c5:55:7e => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.10.76 MAC_SRC:0c:d2:92:92:9c:26 => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.11.247 MAC_SRC:c4:9a:02:9a:d0:f9 => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.10.86 MAC_SRC:10:3b:59:c3:b4:24 => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
IP_SRC:192.168.11.145 MAC_SRC:40:78:6a:f8:7d:1b => IP_DST:192.168.10.1 MAC_DST:90:e6:ba:9b:9a:24
Total Geral de Erros de CRC: 1054

```

Fonte: Do autor.

Considerando o cenário 5: para simular o caso referente a rede lenta (Internet) foram disparados três downloads com tamanho de 1,6 GB cada, em dois computadores distintos, realizando um consumo significativo de link, em um terceiro computador foi realizado uma cópia de um arquivo pela rede via ssh (acesso remoto) com tamanho de aproximadamente 3 GB, estes procedimentos deixaram a rede bastante lenta. A primeira assinatura definida foi *tshark -nr /tmp/coleta/#ARQUIVO -T fields -e frame.number -e ip.src -e eth.src -e _ws.col.Info -Y 'tcp.analysis.window_update' | awk '{ print \$2,\$3}'* onde é possível identificar o filtro *tcp.analysis.window_update*; que significa que está sendo necessário aumentar o tamanho da janela de transmissão de dados e por consequência disso acaba abrindo uma nova janela de transmissão, este processo ocorre de forma recorrente podendo gerar um alto índice de janelas de transmissão de dados deixando assim a rede lenta.

A segunda assinatura definida foi *tshark -nr /tmp/coleta/#ARQUIVO -T fields -e ip.dst -e eth.dst -Y 'frame.len == 1514 && tcp.analysis.initial_rtt > 0.1 && tcp.analysis.lost_segment' | awk '{print \$1,\$2}'* onde é possível identificar três filtros: para identificação do tamanho correto da janela foi utilizado o *frame.len == 1514*; e para a verificação da latência alta referente a confirmação (ACK) de pacotes o filtro *tcp.analysis.initial_rtt > 0.1*; e para identificar falhas na segmentação de pacotes o filtro *tcp.analysis.lost_segment*; a combinação destas duas assinaturas permite identificar uma rede lenta (Chappel, 2014).

Após, criadas as assinaturas é realizado o procedimento de definição da métrica para módulo de identificação de anomalia. Esta métrica consiste em verificar se a latência média do arquivo de captura atual é maior que a latência que temos conhecimento dentro do período de uma semana, o qual foi gerado através das coletas de monitoramento realizadas para definir a latência padrão da rede. Caso o valor de latência da captura atual seja maior que o valor médio de latência padrão da rede é executado a segunda assinatura que é responsável por verificar a demora na confirmação dos pacotes (ACK) e também identificar falha de segmentação dos mesmos.

Depois de definidas as métricas é realizado o processo de coleta e análise do tráfego onde devemos identificar quais hosts estão gerando alto volume de dados na rede, desta forma é capturado o endereço IP/MAC de origem do host assim como o valor de latência

da rede gerada por ele. A Figura 43 abaixo mostra o exato momento que ocorre a identificação da anomalia.

Figura 43 - Identificação em tempo real rede lenta (Internet).

```
=====
Rede lenta (internet lenta)
=====
tshark -nr /tmp/coleta/empresaT-rede_lenta.cap -T fields -e frame.number -e ip.src -e eth.src -e
_ws.col.Info -Y 'tcp.analysis.window_update' | awk '{ print $2,$3}'
avg_latencia_normal: 2
tshark -nr /tmp/coleta/empresaT-rede_lenta.cap -T fields -e ip.dst -e eth.dst -Y 'frame.len == 1514
&& tcp.analysis.initial_rtt > 0.1 && tcp.analysis.lost_segment'
Rede lenta top hosts
IP_SRC:10.0.52.164 MAC_SRC:08:00:46:f4:3a:09
Numero de pacotes c/ alta latencia: 99
Total de pacotes (TCP Window Update): 1605
```

Fonte: Do autor.

6 AVALIAÇÃO E RESULTADOS

Este capítulo faz uma análise dos resultados obtidos da ferramenta proposta em ambiente de produção em clientes reais. Os cenários estão dispostos da seguinte maneira: (1) conflito de servidores DHCP, (2) *loop* na rede, (3) IP duplicado na rede, (4) erro de CRC e (5) rede lenta (Internet). Para esta avaliação foi separado as empresas em três categorias: pequena, média e grande.

1 - Conflito de DHCP na rede

Na empresa F, que é de médio porte, devido a uma falha de hardware e/ou software um dos roteadores resetou a configuração padrão de fábrica que vem com o DHCP ativo e começou a distribuir endereços IP em outra faixa, por consequência disso havia algumas máquinas que não estavam conseguindo acessar os arquivos compartilhados na rede.

A infraestrutura da empresa F pode ser recordada através da Tabela 5 na qual apresenta a estrutura de equipamentos da empresa. Nesta infraestrutura, a empresa possui um roteador que é responsável por distribuir endereço de IP por DHCP na rede e os demais roteadores estão ligados somente com a função de repetidor de sinal na rede.

Na identificação do problema, o sistema desenvolvido contribuiu significativamente na rápida identificação desta anomalia, pois através da análise em tempo real conseguiu identificar a existência de dois servidores de DHCP respondendo a uma mesma solicitação na rede, após esta identificação foi realizado o acesso ao roteador e comprovamos que o mesmo esta com o DHCP ativo na rede. A Figura 44 ilustra os alertas da central de logs do sistema, email e telegram.

Figura 44 - Alertas gerados conflito de DHCP.

The screenshot displays the 'Central de Logs' interface. At the top, there are buttons for 'Excel', 'PDF', 'Copy', 'Print', 'CSV', and 'Ocultar Colunas', along with a search bar labeled 'Pesquisar:'. Below this is a 'Mensagem' section with a 'Data hora' dropdown set to '20/05/2017 14:50:57'. The main message content reads: 'Conflito de DHCP na rede: IP: 192.168.50.1 | MAC: 00:e0:20:2b:06:e5' and 'IP: 192.168.0.1 | MAC: e8:de:27:55:03:2b'. Below the message, there are two notification panels. The left panel, titled 'SDATR: Anomalia identificada', shows an email notification for 'sulvale.testes@gmail.com' with the subject 'Conflito de DHCP na rede' and a 'Log Email' button. The right panel, titled 'SulvaleMonitor 3 membros', shows a telegram notification with the same subject and a 'Log Telegram' button.

Fonte: Do autor.

2 - *Loop* na rede

Na empresa X, que é de grande porte, o problema começou através de um switch que havia um cabo com uma extremidade conectada a ele e a outra extremidade não, por engano alguém pegou a extremidade que não estava conectada no switch e conectou fechando assim um *loop* na rede. Neste caso em específico todo o tráfego gerado na rede era replicado novamente em todas as portas do switch, que por sua vez replicava para outro switch, deixando assim a rede completamente sobrecarregada, também vale ressaltar que os switches desta empresa não possuíam o recurso de *spanning tree* no qual consiste em prevenir este tipo de problema.

A infraestrutura da empresa X pode ser recordada através da Tabela 5 na qual apresenta a estrutura de equipamentos da empresa.

Para identificação deste caso, o sistema desenvolvido levou cerca de 3-4 minutos para identificar o problema devido ao alto volume de tráfego da rede e tamanho do arquivo de captura que ficou com cerca de 300 MB, além disso, também ocorreu o travamento da interface web do sistema devido a este alto fluxo de dados na rede, porém os *scripts* continuaram a execução e identificação da anomalia em *background*.

Como resultado o sistema gerou logs dos 10 endereços IPs com maior número de pacotes trafegados na rede e através desta listagem conseguimos identificar o caminho de origem do *loop*, pois através do mapeamento de rede que a empresa possuía conseguimos realizar os cruzamentos dos endereços IP do log do sistema com os do mapeamento conseguindo assim chegar ao switch que interligava estes hosts. A Figura 45 mostra os logs gerados pelo sistema.

Figura 45 - Alertas gerados *loop* na rede.

Central de Logs

Excel PDF Copy Print CSV Ocultar Colunas Pesquisar:

Mensagem	Data hora
Loop na rede identificado - Top Hosts IP: 224.0.0.252 Num. Pacotes: 258544 IP: 239.255.255.250 Num. Pacotes: 170998 IP: 192.168.0.96 Num. Pacotes: 115248 IP: 192.168.0.22 Num. Pacotes: 108128 IP: 224.0.0.251 Num. Pacotes: 51495 IP: 192.168.0.255 Num. Pacotes: 49064 IP: 192.168.0.200 Num. Pacotes: 45683 IP: 192.168.2.1 Num. Pacotes: 35486 IP: 192.168.0.82 Num. Pacotes: 34159 IP: 192.168.1.1 Num. Pacotes: 27820	20/05/2017 15:49:15

Log Email

sulvale.testes@gmail.com 15:48 (Há 11 minutos) ☆
 para mim
 Loop na rede identificado - Top Hosts
 IP: 224.0.0.252 | Num. Pacotes: 258544
 IP: 239.255.255.250 | Num. Pacotes: 170998
 IP: 192.168.0.96 | Num. Pacotes: 115248
 IP: 192.168.0.22 | Num. Pacotes: 108128
 IP: 224.0.0.251 | Num. Pacotes: 51495
 IP: 192.168.0.255 | Num. Pacotes: 49064
 IP: 192.168.0.200 | Num. Pacotes: 45683
 IP: 192.168.2.1 | Num. Pacotes: 35486
 IP: 192.168.0.82 | Num. Pacotes: 34159
 IP: 192.168.1.1 | Num. Pacotes: 27820

Log Telegram

SU SulvaleMonitor 15:49:07
 Loop na rede identificado - Top Hosts
 IP: 224.0.0.252 | Num. Pacotes: 258544
 IP: 239.255.255.250 | Num. Pacotes: 170998
 IP: 192.168.0.96 | Num. Pacotes: 115248
 IP: 192.168.0.22 | Num. Pacotes: 108128
 IP: 224.0.0.251 | Num. Pacotes: 51495
 IP: 192.168.0.255 | Num. Pacotes: 49064
 IP: 192.168.0.200 | Num. Pacotes: 45683
 IP: 192.168.2.1 | Num. Pacotes: 35486
 IP: 192.168.0.82 | Num. Pacotes: 34159

Fonte: Do autor.

3 – IP duplicado na rede

Na empresa M, que é de grande porte, ocorreu um problema no servidor de DHCP, onde o mesmo começou a distribuir endereços IPs que já estavam sendo utilizados na rede, gerando assim duplicidade de endereços IPs na rede e deixando a rede completamente comprometida. A infraestrutura da empresa M pode ser recordada através da Tabela 5 na qual apresenta a estrutura de equipamentos da empresa.

Na identificação do problema, o sistema desenvolvido teve um papel importante, pois durante a identificação do problema foi possível verificar que o servidor de DHCP estava distribuindo vários endereços IPs que já estavam sendo utilizados, na Figura 46 mostra os alertas gerados pelo sistema. Após este diagnóstico, foram acessados as configurações do servidor de DHCP onde foi identificado que as configurações estavam erradas e gerando erros no seu logs internos.

Figura 46 - Alertas gerados IP duplicado na rede.

The screenshot displays a 'Central de Logs' interface. At the top, there are buttons for 'Excel', 'PDF', 'Copy', 'Print', 'CSV', and 'Ocultar Colunas'. A search bar on the right shows the date '29/04/2017'. Below this is a 'Mensagem' section with a 'Data hora' dropdown. The main content area shows a message from 'sulvale.testes@gmail.com' dated '29 de abr'. The message text is: 'IP(s) duplicado(s) na rede: IP: 192.168.1.210 | MAC: 00:0a:f7:44:2e:10 IP: 192.168.3.249 | MAC: c0:38:96:a1:6b:e5'. A sidebar on the right shows 'SulvaleMonitor' with 3 members and a search icon. The sidebar also displays the same IP duplication alert with a timestamp of '14:53:42'.

Fonte: Do autor.

4 – Erro de CRC

Na empresa C, que é de grande porte, o cliente havia relatado problema de rede lenta e perda de conectividade na maioria das máquinas, deixando assim a rede parcialmente comprometida. A infraestrutura da empresa C pode ser recordada através da Tabela 5 na qual apresenta a estrutura de equipamentos da empresa.

Na identificação do problema, o sistema desenvolvido contribuiu significativamente na rápida identificação desta anomalia, pois após realizar a coleta de alguns arquivos de captura na rede o sistema, levou de 2-3 minutos para identificar qual era a anomalia que estava acontecendo, que consistia em um computador disseminando erros de CRC, deixando assim a rede bastante comprometida. A Figura 47 ilustra os alertas gerados pelo sistema. Um aspecto bastante interessante neste caso foi a questão da empresa estar pronta para realizar a troca de um switch sendo que o problema não estava no mesmo.

Figura 47 - Alertas gerados erro de CRC na rede.

Central de Logs	
Excel PDF Copy Print CSV Ocultar Colunas	Pesquisar: <input type="text"/>
Mensagem	Data hora
Identificado Erro de CRC na rede IP_SRC: 192.168.1.49 MAC_SRC: 30:5a:3a:9e:a7:e7 => IP_DST: 192.168.1.76 MAC_DST: 00:00:00:00:00:00 IP_SRC: 192.168.1.49 MAC_SRC: 30:5a:3a:9e:a7:e7 => IP_DST: 192.168.1.16 MAC_DST: 00:00:00:00:00:00 Total Geral de Erros de CRC: 2	09/05/2017 11:39:28
Identificado Erro de CRC na rede IP_SRC: 192.168.1.49 MAC_SRC: 30:5a:3a:9e:a7:e7 => IP_DST: 192.168.1.9 MAC_DST: 00:00:00:00:00:00 Total Geral de Erros de CRC: 1	09/05/2017 11:24:51
Identificado Erro de CRC na rede IP_SRC: 192.168.1.49 MAC_SRC: 30:5a:3a:9e:a7:e7 => IP_DST: 192.168.1.1 MAC_DST: 00:00:00:00:00:00 Total Geral de Erros de CRC: 1	09/05/2017 11:18:26

Fonte: Do autor.

Na Figura 47 podemos identificar que a máquina que estava gerando tráfego indevido, consiste no IP: 192.168.1.49, que por um problema físico na placa de rede que posteriormente foi constatado pela parte técnica, começou a gerar e enviar erros de CRC para o gateway da rede (192.168.1.1), para o DNS primário da rede (192.168.1.9), DNS secundário da rede (192.168.1.16) e para a máquina responsável pelas câmeras (192.168.1.76), deixando assim a infraestrutura de rede comprometida. Após desligar a máquina o tráfego da rede normalizou, e ficou estável voltando a funcionar perfeitamente.

5 – Rede lenta (Internet)

Na empresa P, que é de pequeno porte, devido ao alto volume de transferência de arquivos através de FTP (rotina realizada pelo sistema interno), em determinados períodos do dia a rede começava a ficar extremamente lenta.

A infraestrutura da empresa P pode ser recordada através da tabela 5 na qual apresenta a estrutura de equipamentos da empresa.

Após realizarmos a instalação do sistema e realizar a coleta e análise de tráfego o sistema conseguiu identificar os pontos de origem de lentidão na rede, onde com base nos endereços IPs apresentados no log do sistema na Figura 48 abaixo, foi possível identificar que consistiam nas máquinas que recebiam dados por FTP referente a um sistema de uso interno. Para solução deste caso foi realizado a interligação de matriz e filial através de uma VPN, diminuindo assim o tráfego na rede.

Figura 48 - Alertas gerados rede lenta (Internet).

Central de Logs

Excel PDF Copy Print CSV Ocultar Colunas

Pesquisar:

Mensagem Data hora

Rede lenta top hosts

IP_SRC: 192.168.1.104 | MAC_SRC: 10:bf:48:9c:e5:2c

IP_SRC: 192.168.1.110 | MAC_SRC: c4:73:1e:e7:87:8c

Numero de pacotes c/ alta latencia: 1160

Total de pacotes (TCP Window Update): 507

17/05/2017 21:12:35

Message Preview:

21:12 (Há 1 hora) ☆ ↶

para mim

inglês > português Traduzir mensagem Desativar para: inglês x

Rede lenta top hosts

IP_SRC: 192.168.1.104 | MAC_SRC: 10:bf:48:9c:e5:2c

IP_SRC: 192.168.1.110 | MAC_SRC: c4:73:1e:e7:87:8c

Numero de pacotes c/ alta latencia: 1160

Total de pacotes (TCP Window Update): 507

Logs enviado por Email

Notification Card:

SulvaleMonitor 3 membros

Rede lenta top hosts 21:12:45

IP_SRC: 192.168.1.104 | MAC_SRC: 10:bf:48:9c:e5:2c

IP_SRC: 192.168.1.110 | MAC_SRC: c4:73:1e:e7:87:8c

Numero de pacotes c/ alta latencia: 1160 **Log Telegram**

Total de pacotes (TCP Window Update): 507

Fonte: Do autor.

A avaliação do sistema desenvolvido em ambiente de produção permitiu realizarmos a validação das assinaturas criadas, assim como as métricas desenvolvidas para identificação das anomalias. A avaliação em ambientes reais foi fundamental neste trabalho para a correta definição de métricas e identificação de problemas em ambientes reais do dia-a-dia das empresas.

7 CONCLUSÃO E TRABALHOS FUTUROS

Nos dias atuais a infraestrutura de rede tornou-se parte crítica das empresas e organizações de modo que o tempo de inoperância da rede por menor que seja acaba por causar prejuízos tanto na parte operacional quanto na parte financeira. Desta forma a utilização de uma ferramenta para analisar o tráfego e detectar anomalias na rede em tempo real se faz necessário, para identificarmos de forma rápida os problemas na rede, que se não forem identificados rapidamente, podem causar sérios prejuízos, como por exemplo, levar a total inoperância da rede.

Considerando as diversas possibilidades de estudo e desenvolvimento de soluções para os problemas aqui descritos quanto à detecção e análise de anomalias no tráfego de rede, este trabalho buscou gerar ganho científico através do desenvolvimento de uma ferramenta para contribuir com o estado-da-arte na identificação de anomalias através da análise de tráfego de rede utilizando metodologia de detecção baseada em conhecimento para prover uma base de assinaturas de anomalias conhecidas, referente a problemas nas redes locais.

Os resultados obtidos através da ferramenta permitiram, uma avaliação mais rápida e precisa referente aos problemas aqui definidos. Outro ponto importante foi poder realizar a validação em redes de diferentes portes, onde foi possível ver o sistema se adaptar a diferentes cenários e infraestruturas de redes reais. Também ressaltamos a grande importância que os arquivos de captura extraídos de ambientes reais tiveram no processo de criação e aprimoramento das assinaturas criadas.

Como trabalho futuro, sugere-se a criação de novos perfis, possibilitando ampliar a base de conhecimento e trabalhar com a análise e identificação de anomalias em redes Wireless. Hoje, a ferramenta desenvolvida trabalha com a identificação de anomalias em redes locais cabeadas.

Uma outra oportunidade de trabalho é a criação de um aplicativo *mobile* que habilite a integração com a ferramenta desenvolvida a fim de disponibilizar os dados da central de logs e criação de gráficos de ocorrência dos problemas para fornecer aos gestores da rede um monitoramento mais flexível onde possam acessar de qualquer lugar através do próprio celular.

Ao final, uma última oportunidade de trabalho diz respeito a ampliação da ferramenta desenvolvida através do desenvolvimento de relatórios com o histórico de ocorrência de anomalias identificadas, os equipamentos envolvidos e horário de registro. Por fim, sugere-se a criação de um módulo de chamados (TTS - Trouble Ticket Systems) onde seja possível realizar a abertura de um chamado automático para o setor de TI quando ocorrer a incidência de algum dos problemas atacados neste trabalho.

REFERÊNCIAS

ABE, N.; ZADROZNY, B.; LANGFORD, J. *Outlier detection by active learning*. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, New York, NY, USA, p.504-509, 2006.

ARAGON, Jim; COMBS Gerald; CHAPPELL, Laura. *Thoubleshooting with Wireshark, locate the source of performance problems*. USA: Chappell University, 2014, ISBN: 978-1-893939-97-4.

BISOL, Rodolfo Vebber; SILVA, Anderson Santos; MACHADO, Cristian Cleder; GRANVILLE, Lisandro Zambnedetti; SHAEFFER-FILHO, Alberto Egon. Coleta e Análise de Características de Fluxo para Classificação de tráfego em Redes Definidas por Software. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC). 2016.

BREMLER, Anat Barr; YOTAM, Harchol; HAY, David; YARON, Koral. *Deep Packet Inspection as a Service*. In Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies (CoNEXT '14). ACM, New York, NY, USA, p.271-282. 2014

BRO, *Network Security Monitor*. Disponível em: < <https://www.bro.org/>>. Acesso em: 23 out. 2016.

CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. *Anomaly Detection: A Survey*. University of Minnesota. ACM Comput. Surv. 2009.

CHAPPEL, Laura; COMBS, Gerald. *Wireshark 101, essential skills for network analysis*. USA: Chappell University, 2013, p 297-360.

CHAPPEL, Laura. *Wireshark network analysis, the official wireshark certified network study guide*. USA: Chappell University, 2º edição, 2012.

FLOODLIGHT. Disponível em: < <http://www.projectfloodlight.org/floodlight/>>. Acesso em: 15 out. 2016.

FONTUGNE, Romain; MAZEL, Johan; FUKUDA, Kensuke. *Hashdoop: A MapReduce framework for network anomaly detection*. Computer Communications Workshops (INFOCOM WKSHPs), 2014 IEEE Conference on, Toronto, ON, 2014, pp. 494-499.

GALVÃO, Ricardo Kléber M. *Introdução à análise forense em redes de computadores*. São Paulo: Novatec, 2013, ISBN: 978-85-7522-307-9, p.59-72.

HONGYI, Zeng; SHIDONG, Zhang; FEI, Ye; VIMALKUMAR, Jeyakumar; MICHEY, Ju; JUNDA, Liu; NICK, McKeown; AMIN, Vahdat. *Libra: divide and conquer to verify forwarding tables in huge networks*. In Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI'14). USENIX Association, Berkeley, CA, USA, p.87-99. 2014

KUROSE, James F.; ROSS, Keith W. *Redes de computadores e a Internet: Uma abordagem top down*. São Paulo: Person, 2013, p.244-245.

LARI, Paulo Augusto Moda; AMARAL, Dino Macedo. *Snort, Mysql, Apache e ACID*. Rio de Janeiro: Brasport, 2004, p.21-22

LINS, Bruno F. O.; FEITOSA, Eduardo L.; SADOK, Djamel F. H. *Aplicando a teoria da Evidência na Detecção de Anomalias*. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC). 2009.

MARNERIDES, Angelos K.; JAMES, C.; SCHAEFFER-FILHO, Alberto; SAIT, S.; MAUTHE, Andreas; Murthy, H. *Multi-level Network Resilience: Traffic Analysis, Anomaly Detection and Simulation*. ICTACT Journal on Communication Technology, Special Issue on Next Generation Wireless Networks and Applications (2011), Vol. 2, Issue 2, ISSN: 2229-6948. 2011.

MARNERIDES, Angelos K.; MAUTHE, Andreas. *Analysis and Characterisation of Botnet Scan Traffic*. International Conference on Computing, Networking and Communications (ICNC). 2016.

MARNERIDES, Angelos K.; SCHAEFFER-FILHO, Alberto; MAUTHE, Andreas. *Traffic Anomaly Diagnosis in Internet Backbone Networks: A Survey*. Computer Networks. 2014.

MININET. Disponível em: <<http://mininet.org/>>. Acesso em: 10 out. 2016.

MERINO, Borja. *Instant Traffic Analysis with Tshark How-to*. UK: Packtpub, 2013, ISBN: 978-1-78216-538-5.

MOTA FILHO, João Eriberto. *Análise de Tráfego em Redes TCP/IP*. São Paulo: Novatec, 2013, ISBN: 978-85-7522-375-8, p.34.

MYSQL, Sistema de gerenciamento de banco de dados.
Disponível em: <<https://www.mysql.com/>>.
Acesso em: 05 mai. 2017.

NADEEM, A; HOWARTH, M. P. *A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks*. in IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2027-2045, Fourth Quarter, 2013.

NIKHIL, Handigol; BRANDON, Heller; VIMALKUMAR, Jeyakumar; MAZIÈRES, David; MCKEOWN, Nick. *I know what your packet did last hop: using packet histories to troubleshoot networks*. In Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI'14). USENIX Association, Berkeley, CA, USA, p.71-85. 2014

SANDERS, Chris. *Practical Packet Analysis. Using Wireshark to Solve Real-World Network Problems*. San Francisco: No Starch Press, 2007, ISBN-13: 978-1-59327-149-7.
SCAPY. Disponível em: <<http://www.secdev.org/projects/scapy/>>.
Acesso em: 12 out. 2016.

SHIMONSKI, Robert. *Wireshark Guia Prático. Análise e resolução de problemas de tráfego de rede*. São Paulo: Novatec, 2014, ISBN: 978-85-7522-388-8.

SILVA, Anderson Santos; WICKBOLDT, Juliano Araujo; SCHAEFFER-FILHO, Alberto; MARNERIDES, Angelos K.; MAUTHE, Andreas. *Tool Support for the Evaluation of Anomaly Traffic Classification for Network Resilience*. IEEE Symposium on Computers and Communication (ISCC). 2015.

SIQUEIRA, Heitor Ricardo Alves; BARUQUE, Alexandre O. Cansian; GEUS, Paulo Lício; GRÉGIO André Ricardo Abed. *Uma Arquitetura para Análise e Visualização de Tráfego de Rede Malicioso*. Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSEG). 2015.

STEINWART, I.; HUSH, Don; SCOVEL, C. *A classification framework for anomaly detection*. Journal of Machine Learning Research 6, 211-232. 2005.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. *Introduction to Data Mining*. First Edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2005.

TEODORO, P. García; DÍAZ VERDEJO, J; MACIÁ FERNÁNDEZ, G.; VÁZQUEZ, E. *Anomaly-based network intrusion detection: Techniques, systems and challenges*. Computers & Security, Volume 28, Issues 1–2, February 2009, Pages 18-28. 2009.

TELEGRAM, API Telegram Bot. Disponível em: < <https://core.telegram.org/>>. Acesso em: 18 abr. 2017.

TCPDUMP, *Packet Analyzer*. Disponível em: < <http://www.tcpdump.org/>>. Acesso em: 02 out. 2016.

TRACEROUTE. Disponível em: <http://www.traceroute.org/>>. Acesso em: 05 out. 2016.

VIRUS TOTAL. Disponível em: < <https://www.virustotal.com/pt/>>. Acesso em: 12 out. 2016.

WATTENBERG, Federico Simmross; PÉREZ, Juan Ignacio Asensio; HIGUERA, Pablo Casaseca; FERNÁNDEZ, Marcos Martín; DIMITRIADIS, Ioannis A; Senior Member; IEEE; LÓPEZ, Carlos Alberola. *Anomaly Detection in Network Traffic Based on Statistical Inference and α -Stable Modeling*. IEEE Transactions on Dependable and Secure Computing (Volume: 8, Issue: 4). 2011.

WIRESHARK, *Sample Captures Datasets*. Disponível em: < <https://wiki.wireshark.org/SampleCaptures> >. Acesso em: 22 maio 2017.

WIRESHARK, *Wireshark Network Protocol Analyzer*. Disponível em: < <https://www.wireshark.org/>>. Acesso em: 16 out 2016.

ANEXO – INSTALAÇÃO, CONFIGURAÇÃO E UTILIZAÇÃO DO SISTEMA

As tecnologias utilizadas durante o desenvolvimento foram: linguagem de programação Python 3 com Django (versão 1.10), como servidor Web foi utilizado o Nginx, para captura de tráfego de rede foi utilizado o Tcpdump, para análise de tráfego utilizamos o Tshark. O sistema é executado em um servidor Linux Ubuntu Server 16.04 LTS.

Para realizar a instalação da ferramenta, é necessário copiar a pasta chamada *sdatr* (disponível na mídia entregue junto ao trabalho) para dentro do diretório */opt* do Linux e posteriormente acessar a pasta: *scripts* dentro da pasta do sistema no seguinte caminho: *sdatr/smatr/system/scripts/instalador.sh*. O *script instalador.sh* é responsável por realizar toda a instalação das dependências necessárias para o funcionamento do sistema. Após a instalação de todos os pacotes necessários vai aparecer uma tela para ser realizada a configuração de IP referente ao servidor web (Nginx), basta informar o IP do servidor e salvar o arquivo através do comando *esc:wq* (conforme padrão do editor VI Linux). Neste mesmo caminho temos todos os *scripts* utilizados no sistema tais como: *script* de backup do banco de dados *do_bkp_banco.sh* no qual vai gerar os arquivos com o formato (*sdatr* mais a data e hora do backup) dentro da pasta *sdatr/banco/*. Também dispomos do *script do_set_config.sh*, caso seja necessário alterar as configurações do servidor web tais como endereço IP e porta por exemplo. Por fim, temos o *script* responsável por iniciar o servidor web chamado *do_start.sh*, depois de executar este *script* podemos acessar a interface da aplicação web através de um navegador, informando o endereço IP e porta definido nas configurações.

Depois de finalizada a instalação e configuração do sistema, podemos realizar a coleta de tráfego e identificação de anomalias. Para realizar a captura de pacotes utiliza-se o *script do_captura.sh* e para realizar o procedimento de identificação de anomalia temos o *do_identifica_anomalia.sh*.

Os três *scripts do_start.sh, do_captura.sh do_identifica_anomalia.sh* devem estar executando simultaneamente para o correto funcionamento do sistema.

Santa Cruz do Sul, 22 de Junho de 2017.

Tiago Silva Leal

Prof. Me. Lucas Fernando Müller