UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

CORRELAÇÃO DE ALERTAS EM UM INTERNET EARLY WARNING SYSTEM

DISSERTAÇÃO DE MESTRADO

Tarcisio Ceolin Junior

Santa Maria, RS, Brasil

2014

CORRELAÇÃO DE ALERTAS EM UM INTERNET EARLY WARNING SYSTEM

Tarcisio Ceolin Junior

Dissertação apresentada ao Curso de Mestrado Programa de Pós-Graduação em Informática (PPGI), Área de Concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Mestre em Ciência da Computação

Orientador: Prof. Dr. Osmar Marchi dos Santos

Co-orientador: Prof. Dr. Raul Ceretta Nunes

Santa Maria, RS, Brasil

2014

Ceolin Junior, Tarcisio

Correlação de Alertas em um Internet Early Warning System / por Tarcisio Ceolin Junior. – 2014.
66 f.: il.; 30 cm.

Orientador: Osmar Marchi dos Santos
Co-orientador: Raul Ceretta Nunes
Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2014.

1. Correlação de Alertas. Detecção de Intrusão. Internet Early Warning System. Consciência Situacional. I. Marchi dos Santos, Osmar. II. Ceretta Nunes, Raul. III. Título.

E-mail: ceolin@inf.ufsm.br

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

CORRELAÇÃO DE ALERTAS EM UM INTERNET EARLY WARNING SYSTEM

elaborada por
Tarcisio Ceolin Junior

como requisito parcial para obtenção do grau de
Mestre em Ciência da Computação

COMISSÃO EXAMINADORA:

Osmar Marchi dos Santos, Dr.
(Presidente/Orientador)

Carlos Alberto Maziero, Prof. Dr. (UTFPR)

Andrei Piccinini Legg, Prof. Dr. (UFSM)

Santa Maria, 28 de Fevereiro de 2014.

À minha esposa Sinéia, sem ela nenhum sonho seria possível ou valeria a pena.

AGRADECIMENTOS

Agradeço em especial ao Prof. Dr. Osmar Marchi dos Santos pela confiança, incentivo, paciência, dedicação e profissionalismo como orientador. Agradeço também a co-orientação do Prof. Dr. Raul Ceretta Nunes, pelas palavras-chaves despendidas durante todo o processo de pesquisa.

À toda minha família, em especial a minha esposa Sinéia pela compreensão e encorajamento durante todo este período, assim como minha irmã Simone pela preocupação, incentivo e apoio.

Agradeço também a Universidade Federal de Santa Maria e ao Programa de Pós-Graduação em Informática que, tanto pela infraestrutura ofertada quanto pelo corpo docente e de técnicos administrativos, proporcionou que esta dissertação tenha sido realizada.

“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que ninguém ainda pensou sobre aquilo que todo mundo vê.”

— ARTHUR SCHOPENHAUER

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

CORRELAÇÃO DE ALERTAS EM UM INTERNET EARLY WARNING SYSTEM

AUTOR: TARCISIO CEOLIN JUNIOR
ORIENTADOR: OSMAR MARCHI DOS SANTOS
CO-ORIENTADOR: RAUL CERETTA NUNES
Local da Defesa e Data: Santa Maria, 28 de Fevereiro de 2014.

Sistemas de Detecção de Instrução (*Intrusion Detection Systems – IDS*) são projetados para monitorar possíveis ataques à infraestruturas da rede através da geração de alertas. Com a crescente quantidade de componentes conectados na rede, os IDS tradicionais não estão sendo suficientes para a efetiva detecção de ataques maliciosos, tanto pelo volume de dados como pela crescente complexidade de novos ataques. Nesse sentido, a construção de uma arquitetura *Internet Early Warning Systems* (IEWS) possibilita detectar precocemente as ameaças, antes de causar algum perigo para os recursos da rede. O IEWS funciona como um coletor de diferentes geradores de alertas, possivelmente IDS, centralizando e correlacionado informações afim de gerar uma visão holística da rede. Sendo assim, o trabalho tem como objetivo descrever uma arquitetura IEWS para a correlação de alertas gerados por IDS dispersos geograficamente utilizando a técnica *Case-Based Reasoning* (CBR) em conjunto com Georreferenciamento de endereços IP. Os resultados obtidos nos experimentos, realizados sobre a implementação da técnica desenvolvida, mostraram a viabilidade da técnica na redução de alertas classificados como falsos-positivos. Isso demonstra a aplicabilidade da proposta como base para o desenvolvimento de técnicas mais apuradas de detecção dentro da arquitetura de IEWS estendida.

Palavras-chave: Correlação de Alertas. Detecção de Intrusão. Internet Early Warning System. Consciência Situacional.

ABSTRACT

Master's Dissertation
Post-Graduate Program in Informatics
Federal University of Santa Maria

ALERT CORRELATION IN AN INTERNET EARLY WARNING SYSTEM

AUTHOR: TARCISIO CEOLIN JUNIOR
ADVISOR: OSMAR MARCHI DOS SANTOS
COADVISOR: RAUL CERETTA NUNES

Defense Place and Date: Santa Maria, February 28st, 2014.

Intrusion Detection Systems (IDS) are designed to monitor the computer network infrastructure against possible attacks by generating security alerts. With the increase of components connected to computer networks, traditional IDS are not capable of effectively detecting malicious attacks. This occurs either by the distributed amount of data that traverses the network or the complexity of the attacks launched against the network. Therefore, the design of Internet Early Warning Systems (IEWS) enables the early detection of threats in the network, possibly avoiding eventual damages to the network resources. The IEWS works as a sink that collects alerts from different sources (for example, from different IDS), centralizing and correlating information in order to provide a holistic view of the network. This way, the current dissertation describes an IEWS architecture for correlating alerts from (geographically) spread out IDS using the Case-Based Reasoning (CBR) technique together with IP Georeferencing. The results obtained during experiments, which were executed over the implementation of the developed technique, showed the viability of the technique in reducing false-positives. This demonstrates the applicability of the proposal as the basis for developing advanced techniques inside the extended IEWS architecture.

Keywords: Alert Correlation. Intrusion Detection. Internet Early Warning System. Situational Awareness.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CERT	*Computer Emergency Response Team*
CERT.br	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
HDIS	*Host Intrusion Detection System*
IDMEF	*Intrusion Detection Message Exchange Format*
IDS	*Intrusion Detection System*
IEWS	*Internet Early Warning Systems*
IP	*Internet Protocol*
IDWG	*Intrusion Detection Work Group*
IETF	*Internet Engineering Task Force*
KBAM	*Knowledge Base Attack Monitoring*
NDIS	*Network Intrusion Detection System*
SNMP	*Simple Network Management Protocol*
TI	*Tecnologia da Informação*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
UFSM	Universidade Federal de Santa Maria
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Junto com a crescente dependência da sociedade sobre a Tecnologia da Informação (TI), as preocupações relativas à segurança de TI estão cada vez mais urgentes. O crescente número de ataques ocorridos na rede vem estimulando a conscientização de empresas e instituições a investir tempo e dinheiro em mecanismos para aumentar seu nível de segurança. Na medida em que o volume de dados que trafegam nas redes de computadores aumentam consideravelmente a cada instante, os tradicionais sistemas de detecção de intrusão utilizados pelas empresas tornam-se obsoletos para processar e analisar grandes quantidades de dados (GOLLING; STELTE, 2011).

Não é somente a Internet que vai mudar significativamente, mas também os ataques como conhecemos, que tendem a aumentar e serem mais prejudiciais. Nos últimos anos, além de cada vez mais infraestruturas críticas estarem disponíveis na Internet, houve também um aumento do conhecimento dos invasores no que tange técnicas de invasão. Além disso, é cada vez mais evidente a preocupação dos governos em relação ao ambiente cibernético. Segundo Golling (2011), muitas nações já possuem profissionais de cyber-segurança visando as consequências de uma possível guerra através da Internet. Além disso, há evidências da existência de agências ligadas a governos que possuem como objetivo coletar, através de atividades consideradas como maliciosas, como invasão de servidores, com objetivo de interceptar e analisar o tráfego da rede.

Sistemas *Internet Early Warning Systems* (IEWS) possuem como objetivo a detecção de ameaças na rede precocemente (ONWUBIKO, 2009). Esses sistemas permitem obter uma percepção da rede que possibilitam uma reação precoce a um evento malicioso, oferecendo um maior controle e monitoramento da rede, auxiliando em tomadas de decisões (GOLLING; STELTE, 2011). Estes sistemas trabalham no monitoramento de ambientes de rede e seu principal objetivo é detectar ameaças com antecedência, antes que elas possam causar qualquer ameaça a infraestrutura da rede (GOLLING; STELTE, 2011). Além disso, estes sistemas auxiliam na construção de uma consciência situacional do ambiente, criando uma imagem de segurança dos recursos de rede, auxiliando a equipe de segurança na tomada de decisão.

Em um trabalho anteriormente desenvolvido pelo Grupo de Gestão e Tecnologia em Segurança da Informação da Universidade Federal da Santa Maria (GTSEG-UFSM), foi proposta uma arquitetura inicial de IEWS com foco no armazenamento de alertas vindos de dife-

rentes sensores, no caso IDS distribuidos na rede. Mais especificamente, o trabalho de Petri (2013) apresenta uma base de conhecimento, chamada *Knowledge Base for Attack Monitoring* (KBAM), para o armazenamento de diferentes aspectos da rede com foco em eventos relacionados a detecção de intrusão.

Uma vez que a KBAM tem como objetivo armazenar informações, um dos problemas rapidamente detectados é a grande quantidade de falsos-positivos gerados por Sistemas de Detecção de Intrusão (IDS) que compõem a arquitetura e são armazenados na base de dados. O alto volume de falsos-positivos é um grande problema na segurança da rede, tornando difícil de identificar um ataque legítimo frente a tantos falsos-positivos. Assim, o crescimento da quantidade e da complexidade das informações transitadas diariamente entre a rede local das corporações e a Internet torna difícil o processo de análise realizado dia-a-dia por analistas de segurança.

Para auxiliar nesse processo, o presente trabalho tem como objetivo principal estender o trabalho previamente desenvolvido dentro do GTSEG-UFSM provendo uma nova arquitetura de IEWS. Nessa arquitetura, propõem-se a criação de um componente de correlação de alertas que utiliza uma técnica baseada em cálculo de similaridade, denominada *Case-Based Reasoning* (CBR), utilizada para a correlação de alertas e na redução de falsos-positivos originados de diferentes sensores (IDS) da rede. Em conjunto ao cálculo de similaridade, também é proposto a utilização de Georreferenciamento de Endereços IP (GeoIP) como um fator integrante no cálculo de similaridade. GeoIP é um método utilizado para determinar a localização no planeta de determinado Endereço IP na rede, nesse sentido, obtêm-se uma diversidade maior de atributos que podem ser utilizados para compor o cálculo de similaridade, pois informações como país, cidade e estado do atacante estão disponíveis para o cálculo de similaridade.

1.1 Objetivos

O objetivo desta dissertação é apresentar uma arquitetura de correlação de alertas que possibilite a criação de uma consciência situacional de uma ou diversas redes de computadores através de sensores distribuídos geograficamente em um *Internet Early Warning Systems*. Pretende-se alcançar tal objetivo utilizando e estendendo o modelo de dados de uma base de conhecimento KBAM apresentado por Petri (2013) em uma arquitetura de correlação, com a utilização da técnica de *Case-Based Reasoning* em conjunto ao Georreferenciamento de Endereços IP.

Desta forma, os principais objetivos desta dissertação incluem:

- Estender a arquitetura de IEWS proposta em Petri (2013), com a adição de um mecanismo de correlação de alertas;
- Prover uma mecanismo de correlação de alertas baseado em similaridade, utilizando a técnica CBR;
- Utilizar Georreferenciamento de endereços IP para mensurar a localidade no cálculo de similaridade para a correlação de alertas utilizando CBR;
- Embarcar sensores IEWS em *hardware* de baixo custo, possibilitando um amplo monitoramento da rede.

1.2 Organização

O texto da dissertação está organizado da seguinte forma. O Capítulo 2 apresenta uma fundamentação teórica que aborda conceitos importantes para a compreensão deste trabalho, além de apresentar os trabalhos relacionados com esta dissertação. O Capítulo 3 propõem uma arquitetura de correlação de alertas para IEWS. No Capitulo 4 é apresentado a Correlação de Alertas em um *Internet Early Warning System*. O Capítulo 5 descreve os experimentos realizados através da correlação de alertas, apresentando a redução de falsos-positivos. Por fim, o Capítulo 6 apresenta as conclusões do trabalho e sugestões de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico, descrevendo os principais conceitos de *Intrusion Detecion System (IDS)*, *Internet Early Warning System (IEWS)* e trabalhos relacionados.

2.1 Intrusion Detecion System – IDS

Uma intrusão é definida como uma sequência de ações realizadas por um atacante malicioso que resulta no comprometimento do sistema alvo. Detecção de intrusão é o processo de identificar e responder a atividades maliciosas direcionadas a computadores ou recursos de rede. Esta definição introduz a noção de detecção de intrusão como um processo, que envolve tecnologia, pessoas e ferramentas. É uma abordagem complementar em relação às abordagens tradicionais à segurança como controle de acesso e criptografia (SCARFONE; MELL, 2007).

Sistemas de Detecção de Intrusão (IDS) são aplicações dedicadas à detecção de ataques maliciosos contra recursos de uma rede de computadores. IDSs não se destinam a substituir os métodos tradicionais de segurança, mas sim ser um complemento. Segundo Bace (2001), um Sistema de Detecção de Intrusão é o processo de monitoramento de eventos que podem ocorrer em uma rede ou sistema de computadores e a análise destes para detectar sinais de intrusão, definidos como tentativas de comprometer a confidencialidade, integridade, disponibilidade, ou burlar mecanismos de segurança de uma rede ou computador.

Sendo assim, segundo Pereira (2011) um IDS monitora um ambiente computacional assim como um alarme:

> "Antes de viajar, o dono tranca os objetos de valor dentro da casa usando correntes e cadeados. De nada adiantaria deixar a casa sozinha, se um ladrão astuto tem a liberdade para testar chaves em cadeados e serrar as correntes. Portanto, para melhorar a segurança, o dono coloca alarmes e câmeras em pontos estratégicos da casa. Se o ladrão investir contra o patrimônio, os vigias serão alertados automaticamente pelo sistema de segurança".

A seguir é apresentada uma ordem cronológica do surgimento e amadurecimento de Sistemas de Detecção de Intrusão segundo Innella (2010) e Pereira (2011):

1980 - James Anderson descreve no artigo *Computer Security Threat Monitoring and Surveillance* a primeira noção de detecção de intrusão através de auditoria;

1984 - Dorothy Denning apresenta o primeiro modelo para detecção de intrusão denominado na época IDES no artigo *An Intrusion Detection Model*;

1988 – Criado o projeto *Haystack* na Universidade da Califórnia, resultando em um IDS

baseado em análise de dados de auditoria;

1989 - *Haystack* se torna uma sociedade comercial e lança o *Stalker*, um IDS baseado em *host*;

1990 - Na Universidade da Califórnia, Davis Todd Heberlein introduz a ideia do primeiro IDS baseado em rede denominado *Network Security Monitor* (NSM);

1990 - O IDS baseado em host chamado CMDS (*Computer Misuse Detection System*) é desenvolvido pela SAIC (*Science Applications International Corporation*);

1991 - A Força Aérea dos Estados Unidos desenvolve um sistema chamado *Automated Security Measurement System* - ASIM para monitorar tráfego de rede, mais tarde o projeto transforma-se em uma empresa denominada *Wheel Group*;

1994 - *Wheel Group* lançou o *NetRanger* que foi o primeiro dispositivo IDS baseado em rede comercialmente viável;

1997 - *Internet Security System* (ISS), lança a primeira versão comercial de seu IDS denominado *RealSecure*;

1998 - A Cisco compra a *Wheel Group* para fornecer soluções de segurança a seus clientes;

1998 - Uma companhia de IDS chamada de *Centrax Corporation* surge da fusão de pessoas da equipe do projeto *Haystack* e do projeto CMDS;

1998 - Martin Roesch desenvolve um leve analisador de tráfego de rede em tempo real multiplataforma de código aberto denominado Snort;

1998 - Laboratório Lincoln do MIT (*Massachusetts Institute of Technology*) realiza a primeira avaliação de IDS para a *Defense Advanced Research Projects Agency* (DARPA).

Pode-se perceber que, a partir da década de 1980, surgiram as primeiras propostas de Sistemas de Detecção de Intrusão. Já na década de 1990 foram apresentadas pesquisas e os primeiros Sistemas de Detecção de Intrusão em tempo real. A seguir, será exposta a classificação de Sistemas de Detecção de Intrusão.

2.1.1 Taxonomia

Segundo Bace (2001), Sistemas de Detecção de Intrusão podem ser classificados de várias formas. As formas mais comuns são pelo método de detecção e comportamento de detecção.

O método de detecção descreve as características do analisador na identificação da pos-

sível invasão. Métodos de detecção podem ser baseados em assinatura ou anomalia. Segundo Brandão (2007) a detecção por anomalia procura por situações que destoem do comportamento normal. A construção desses detectores inicia com a construção de um comportamento normal para então definir o comportamento anormal. O problema desta abordagem encontra-se na grande geração de falsos positivos, assim como grande quantidade de recursos computacionais necessários. Já em Sistemas de Detecção de Intrusão baseados em assinatura utiliza regras que buscam por padrões de ataques de intrusões previamente conhecidos, ou seja, qualquer atividade que não condiz com alguma regra é permitida.

O comportamento de detecção classifica o tipo de resposta após a detecção de uma possível ameaça. A forma mais comum de resposta é a passiva, na qual apenas é gerada uma notificação a respeito do incidente detectado. Uma resposta ativa é quando a detecção do incidente resulta em algum tipo de contra-medida pelo sistema, como por exemplo a ativação de novos sensores ou alteração de uma política de segurança, como *firewall*.

Os método de coleta de informações podem ser divididos entre *Host-based Intrusion Detection System* (HIDS) e *Network-based Intrusion Detection System* (NIDS). Segundo Bace (2001), HIDS é um sistema que monitora um único *host* através de logs do sistema com o objetivo de detectar atividades suspeitas. Segundo Silva (2007), NIDS são Sistemas de Detecção de Intrusão que realizam a monitoração do sistema através da captura e análise de cabeçalhos e conteúdo de pacotes de rede, os quais podem ser comparados com padrões de ataques conhecidos ou assinaturas previamente armazenadas em regras.

Nesta dissertação o trabalho focou no uso de NIDS uma vez que a abordagem de correlação desenvolvida tem como objetivo principal detectar ataques na rede. Alguns exemplos de Sistemas de Detecção de Intrusão baseados na taxonomia apresentada são Snort, Suricata e Bro.

Baseado código livre, Snort (SNORT, Acesso em 10/08/2013, Disponível em http://www.snort.org/), é um IDS que está inserido tanto no mercado corporativo como na comunidade de software livre. É um NIDS que utiliza a biblioteca PCAP[1] para capturar e filtrar pacotes de rede, fazendo a inspeção dos pacotes da rede utilizando regras e assinaturas de ataques conhecidos. É uma ferramenta popular na comunidade de software livre por ser confiável e robusta. A arquitetura do Snort se divide em 3 partes: decodificador de pacotes (pré-processador), módulo de detecção e subsistema de alerta e registro.

[1] http://www.tcpdump.org/

Suricata é um IDS/IPS de código aberto de alta performance de detecção de intrusões baseado em regras. Outra característica é ser multi-thread, podendo assim fazer o balanceamento de carga de uma instância entre todos processadores disponíveis da máquina.

The Bro Network Security Monitor (Bro) é um IDS baseado em assinatura que tem como função inspecionar o tráfego de rede para detecção de assinaturas de ataques. Bro tenta verificar strings dentro dos pacotes com as assinaturas existentes em sua base de assinaturas. Possui um flexível processador de assinaturas e que, além de reutilizar a base de assinaturas do Snort, tem como uma de suas principais características a utilização de assinaturas contextuais com objetivo de reduzir falsos-positivos. Segundo Silva (2007), Bro se apresenta como solução para resolver problemas de falsos-positivos. Detalhes adicionais relacionados à atividade exata e sua semântica, de modo a eliminar falsos-positivos provenientes das assinaturas fracas ou informação adicional de como o sistema atacado respondeu ao ataque.

2.2 Internet Early Warning Systems – IEWS

O cenário atual da Internet, juntamente com o acréscimo gradativo no número de informações compartilhadas pelas redes de computadores, têm motivado a construção de *Internet Early Warning Systems*. Um IEWS trabalha no monitoramento da Internet e tem como objetivo principal a detecção precoce de eventos que ameaçam as funcionalidades da Internet (BASTKE; DEML; SCHMIDT, 2010). Além disso, um IEWS visa construir uma consciência situacional e gerar contra medidas para ameaças atuais com base nas informações adquiridas do ambiente monitorado (PETRI et al., 2013).

Segundo Bastke (2010), tem como objetivo detectar precocemente eventos que venham de alguma forma ameaçar a rede monitorada. Ainda, um IEWS é composto por diversos componentes como sensores, componente de detecção, base de conhecimento, componente de reação e gerenciamento de incidentes, componente de perpetuação de evidências e componente de distribuição das informações.

Sensores são utilizados para a geração da visão da atual situação do ambiente monitorado, criando a consciência situacional. Além disso, são responsáveis pela detecção dos eventos de segurança e identificação de novas ameaças. O componente de detecção é dividido em duas camadas: a camada de sinal, onde os dados da rede ou os logs são analisados por métodos de detecção por anomalia ou assinaturas, e a camada de eventos, na qual ocorre o relacionamento dos eventos da camada de sinal com eventos reportados por órgãos externos (FAN; JIHUA;

MIN, 2009).

2.2.1 Knowledge Base Attacks Monitoring – KBAM

Em um trabalho anterior do GTSEG-UFSM foi proposta uma base de conhecimento para ser utilizada em uma arquitetura IEWS. A base de conhecimento KBAM representa os dados de eventos de detecção de intrusão explorando o formato Intrusion *Detection Message Exchange Format* (IDMEF) para mensagens de detecção de intrusão e o formato *Intrusion Detection Response Exchange Format* (IDREF) para mensagens de respostas (PETRI, 2013; PETRI et al., 2012).

Os dados contidos na KBAM consideram os seguintes aspectos: dados de alertas gerados por sistemas de detecção de intrusão, informações sobre as medidas aplicadas em resposta a um alerta e a quantificação do tráfego da rede (PETRI, 2013). Como descrito na Figura 2.1, ao modelar os dados com base nos formatos padrões IDREF e IDMEF, a KBAM pode ser inserida em infraestruturas de rede que possuem IDS que utilizam esses padrões. Neste caso, a KBAM pode ser utilizada como um componente que armazena dados de diferentes aspectos da rede, que são essenciais para o monitoramento de ataques (PETRI, 2013).

Para o uso da KBAM em um IEWS, utiliza-se um sistema gerenciador de eventos de segurança denominado Prelude (PRELUDE, Acesso em 10/08/2013, Disponível em http://www.prelude-ids.com/index.php/uk/). Este sistema gerenciador de eventos é compatível com o formato IDMEF, permitindo que diferentes tipos de sensores criem alertas utilizando um único padrão de comunicação. Ou seja, conforme a Figura 2.1, a KBAM centraliza seus alertas a partir do Prelude. Nesta dissertação, essa base de conhecimento KBAM desenvolvida pelo GTSEG-UFSM é utilizada, porém sua arquitetura como IEWS é revista.

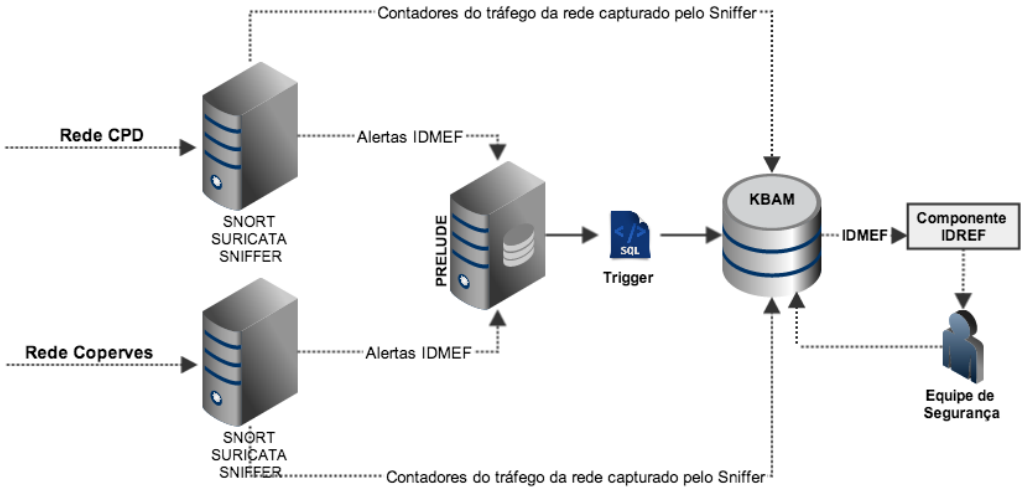

Figura 2.1 – Modelo de uma base de conhecimento KBAM descrito por Petri (2013).

2.3 Trabalhos Relacionados

Os principais trabalhos encontrados na literatura referem-se a proposta teórica de Arquiteturas de IEWS e IDS Distribuídos (que conseguem relacionar informações).

2.3.1 Arquiteturas de IEWS

A arquitetura teórica proposta por Apel (2010) apresenta um *Early Warning System* que trabalha a nível nacional, composto por quatro módulos básicos: *Collecting and Learning* (CL), *Threat Repository* (TR), *Detecting and Alerting* (DA), *Alert Repository* (AR). O primeiro módulo corresponde aos componentes para coleta de malwares e efetiva análise para geração de padrões apropriados. Já o módulo TR é usado para centralizar e gerenciar as informações dos malwares e detectar critérios entregues pelo módulo CL. O repositório TL fornece informações para a detecção de eventos para o módulo DA, que contém os componentes funcionais para detectar violações de segurança e gerar respectivos alertas. Os alertas gerados pelo módulo DA, bem como informações sobre *malwares* fornecidos pelo repositório de ameaças, possibilitam a construção de uma imagem da situação do ambiente monitorado e são gerenciados no módulo AR.

Na abordagem proposta por Bsufka (2006), é apresentado um sistema de alertas teórico baseado em sensores para infraestruturas críticas. O autor combina mecanismos de segurança existentes, como *firewall* e IDS, com novas abordagens, criando uma visão global para determi-

nar o atual estado de ameaça da rede monitorada.

Diferentemente das arquiteturas teóricas sobre IEWS encontradas na literatura, esta dissertação foca na proposta e implementação de uma estrutura inicial para correlação de alertas que compõem fundamentos de um IEWS.

2.3.2 IDS Distribuídos

Segundo Porras (1997), *Event Monitoring Enabling Responses to Anomalous Live Disturbances* (EMERALD) é um framework IDS criado pela *Stanford Research Institute* - SRI. Foi o primeiro a adotar o conceito de sensores distribuídos na detecção de intrusão. Segundo Valdes (2001), um módulo do EMERALD é uma abordagem em redes bayes para detecção em protocolo TCP sem remontagem, mas apenas com verificação de cabeçalhos. Por ser um sistema hierárquico, EMERALD foi escrito para detectar intrusões em redes distribuídas através de diversos sensores e responder a ameaças sobre alvos locais. Tem como característica ler os logs de atividades ou pacotes de rede passivamente, ou ativamente via sondagem que adiciona a coleta de eventos normais. Produz resultados analíticos e os distribuí assincronamente entre outros monitores clientes do EMERALD. Possui uma interface bem definida para compartilhar e receber eventos de dados e resultados e realizam detecção por anomalia baseada em perfil estatístico e analise de assinaturas. O EMERALD implementa uma análise para correlacionar os relatórios de atividades produzidos através do conjunto de domínios monitorados. Este sistema representou um avanço em relação às pesquisas anteriores e o desenvolvimento de detecção por assinatura e anomalia para tratar o monitoramento de grandes sistemas distribuídos e redes. Devido à capacidade de análise em tempo real de modo distribuído e aplicada onde for mais eficaz, em diferentes camadas de abstração, EMERALD apresenta vantagens significativas sobre as abordagens centralizadas, em termos da capacidade de detecção e resposta a eventos, detectando não apenas ataques locais, mas também ataques coordenados como negação de serviço distribuído ou padrões de ataque repetidos contra vários domínios.

Segundo Ramadas (2003), *The Integrated Network-Based Ohio University Network Detective Service – INBOUNDS,* é um IDS de rede, desenvolvido pela Universidade de Ohio. Tecnicamente é um *framework* que faz análise baseada em anomalias de temporizadores e o comprimento das sessões TCP. Segundo Vigna e Kemmerer (1998), o NetSTAT foi uma pesquisa do DARPA que apresentou um sistema de detecção de intrusão onde o analisador estende a abordagem STAT, a qual é utilizada para criar diagramas de transição de estados que repre-

sentam intrusões via rede de computadores.

Prelude-IDS é definido como um gerenciador de eventos de segurança da informação e permite a unificação de vários tipos de aplicações ou sensores, com código-fonte proprietário ou livre. Segundo Debar (2007) o Prelude-IDS permite a integração com IDS de código-fonte aberto ou fechado utilizando um modelo padrão de trocas de mensagens denominado *Intrusion Detection Message Exchange Format* (IDMEF), permitindo a integração entre diferentes tipos de sensores. Segundo Brandrão (2007), o Prelude é um IDS híbrido que agrega e correlaciona alertas gerados por sensores de diversos tipos e fabricantes, distribuídos em uma rede de computadores. No Prelude, é utilizado um modelo hierárquico de análise, sendo assim os dados dos sensores podem ser enviados a um ou mais gerenciadores.

Conforme visto nos trabalhos acima, IDS Distribuídos já encontram-se em um nível alto de maturidade. Diferentemente dos trabalhos descritos, a proposta dessa dissertação utiliza um modelo para correlação de alertas baseado em georreferenciamento, algo que não foi visto nos trabalhos descritos.

2.4 Resumo

Neste capítulo foi traçado, de forma cronológica, sobre a origem e desenvolvimento de Sistemas de Detecção de Intrusão. Foram discutidos alguns conceitos importantes quanto à detecção de intrusão e sua classificação. Mas a necessidade de cooperação entre estes IDS, impulsionado pela crescente oferta de serviços críticos na rede, tem exigido uma abordagem mais complexa como um *Internet Early Warning System*.

Assim, foram apresentados soluções existentes relacionadas ao tema como a base de conhecimento KBAM, desenvolvida pelo GTSEG-UFSM e que será utilizada neste trabalho. Foi evidenciado que os principais trabalhos relacionados da literatura podem ser classificados como Arquiteturas de IEWS (teóricas) e IDS Distribuídos (que realizam correlação de alertas vindos de diferentes sensores). Nesse contexto, o principal diferencial do trabalho nessa dissertação é a proposta de implementação, em um ambiente real (na UFSM), de um sistema de correlação de alertas como base para um IEWS, que utiliza a noção de georreferenciamento como parte integrante no cálculo de similaridade.

3 ARQUITETURA PARA CORRELAÇÃO DE ALERTAS

Este capítulo tem como objetivo propor uma arquitetura de *Internet Early Warning System* em que a correlação de alertas é um ponto fundamental para a obtenção de uma consciência sobre o estado atual de segurança da rede monitorada. A próxima seção descreve a noção de Consciência Situacional e sua aplicação no contexto de detecção de intrusão, fundamental para a proposta da arquitetura. A Seção 3.3 descreve a arquitetura de correlação proposta.

3.1 Consciência Situacional

Para Endsley (1995), cada ato de processamento de informação é mediado pelo sistema de categorias e conceitos, os quais constituem uma representação de mundo. A autora identifica então, que na execução de uma tarefa, a performance da ação é determinada por uma decisão, a qual, por sua vez, depende da adequada compreensão da situação. A compreensão das situações (ou Consciência Situacional) se processa cognitivamente em três níveis: primeiramente, tem-se a percepção dos elementos da situação atual; no segundo nível, os elementos percebidos são compreendidos pela ativação dos mecanismos de memória e associação direta ou indireta com os modelos mentais mais próximos da situação percebida; no terceiro nível, ocorre a manifestação dos mecanismos de antecipação dos status futuros da situação.

Os três níveis descritos precedem a tomada de decisão, a qual será influenciada e determinada pelo modo de controle cognitivo dos operadores para uma dada situação. McGuinness e Foy (2000) adicionaram um quarto nível ao modelo de Endsley, denominado resolução. O objetivo deste novo nível é tentar identificar o melhor caminho a seguir para resolver um determinado problema. A Figura 3.1 apresenta este modelo.

Figura 3.1 – Modelo proposto por Endsley com adição de McGuiness e Foy.

Consciência situacional para Hollands (1999) é definido como sendo uma “capacidade de rapidamente se integrar na representação, àquelas características que mudam na situação”. Para Sarter (2000), consciência situacional refere-se ao acesso a uma representação coerente e explicativa da situação, continuamente renovada, baseando-se de acordo com resultados anteriores. É a partir da consciência situacional que decisões e ações de controle são escolhidas. Todos estes fatores interferem no tempo de resposta do operador a um determinado estímulo. Os fatores da situação são, também, influenciados pela capacidade do sistema, desenho da interface, carga de trabalho do operador, fatores de complexidade e automação.

3.1.1 No contexto de redes de computadores

Tadda (2010), utilizando-se do modelo proposto por Endsley (1995) e sua extensão por McGuinnes e Foy (2000), extrapolaram algumas de suas características, propondo um modelo de consciência situacional para domínio de cyber-segurança da seguinte maneira:

Nível 1 – Percepção: Refere-se ao conhecimento sobre os elementos de dentro da rede que analistas de segurança devem estar cientes, como alertas relatados por sistemas de detecção de intrusão (IDS), logs de *firewall*, varredura de portas, bem como o tempo no qual esses eventos de segurança ocorreram e quais controles específicos relataram tais alertas ou que geraram os logs;

Nível 2 – Compreensão: Refere-se a técnicas, metodologias, processos e procedimentos utilizados para analisar, sintetizar e correlacionar informações percebidas na rede a partir de elementos da própria rede;

Nível 3 – Projeção: Trata-se da capacidade de prever eventos futuros com base no conhecimento extraído da dinâmica dos elementos da rede e à compreensão da situação.

Nível 4 – Resolução: Refere-se à contra medida necessária para tratar os riscos inerentes na rede. Trata-se das medidas necessárias para enfrentar uma situação de rede quando ela ocorre.

Neste contexto, Onwubiko (2009) define o que devem ser atributos funcionais da Consciência Situacional no contexto de segurança de redes de computadores, dentre os atributos propostos, são relevantes no nosso contexto: *Realtime processing, Multisource Data Fusion, Heterogeneity, Security Visualisation e Forecasting and Prediction*, descritos a seguir:

Realtime processing: Para alcançar um maior nível de percepção da situação, a capacidade do dispositivo de computação utilizada para análise da Consciência Situacional deve

possuir a capacidade de fornecer o processamento em tempo real de dados e informação. Isto é análogo à cognitiva humana de uma pessoa ser capaz de realizar pensamentos instantâneos sobre uma situação percebida. A ausência de processamento em tempo real dificulta a possibilidade de se obter uma resposta rápida para ataques percebidos.

Multisource Data Fusion: Trata-se do processo realizado com dados de múltiplas fontes de informação para a detecção, associação, correlação e agregação. Dados originados no *Multisource Data Fusion (MSDF)* são combinados para obter uma melhor precisão, observação e inferências que aqueles obtidos de uma única fonte. Segundo Hall (2004), MSDF engloba a teoria, ferramentas e técnicas para explorar a sinergia nos dados que ajuda a entender melhor um determinado fenômeno.

Trabalhando no Nível 2 (compreensão), MSDF é essencial para criação de uma consciência situacional da rede. A técnica considera a premissa de que a evidência de várias fontes combinadas para detectar ataques proporciona uma melhor compreensão dos ataques do que uma única fonte. De acordo com Haines (2003):

> "Resultados anteriores indicam que não existe um único tipo de controle que, trabalhando de forma individual, seja capaz de detectar todos os tipos de ataques cibernéticos. A atenção deve ser voltada para sistemas de alto nível de correlação capaz de reunir e combinar evidências de diversos sistemas de detecção de intrusão.".

Heterogeneity: É a capacidade da utilização de diferentes fontes heterogêneas em observar, coletar e detectar mudanças dinâmicas na rede. Por exemplo, o uso de IDS, firewalls e antivírus com objetivo de agrupar informações de segurança observadas na rede. A ideia de utilizar sistemas heterogêneos auxilia a criar uma verdadeira consciência situacional, uma vez que nenhum controle único é capaz de identificar todas as ameaças à segurança.

Security Visualisation: Segundo Amico (2005), a visualização permite que a informação de rede seja exibida de tal forma que analistas de segurança tenham condições detectar padrões no tráfego de rede e observar uma grande quantidade de informação de forma concisa. Assim, a visualização tem-se provado uma ferramenta valiosa para trabalhar de forma mais eficaz com dados complexos e manter a consciência situacional.

Forecasting and Prediction: Uma verdadeira ferramenta de consciência situacional deve ser capaz de fazer previsões precisas sobre o estado futuro, independente do seu contexto. Segundo Amico (2005), o objetivo da previsão pode ser tanto para encontrar o estado provável futuro assumindo a progressão atual, quanto para determinar um futuro estado em particular com base em cursos possíveis de ação. É difícil realizar uma previsão correta de estados futuros se a situação atual do estado não pode ser determinada de forma satisfatória. A previsão é

muitas vezes obtida através da comparação de linhas de base, ou combinando o passado para estados atuais, desde que um modelo de confiança seja obtido.

3.2 Técnicas de correlação de alertas

Segundo Salah (2013), técnicas para correlação de alertas possuem suas principais características divididas quanto a fontes da informação e taxonomia de técnicas de correlação de alertas. A primeira refere-se a existência de uma ampla variedade de fontes de informações com o objetivo de alcançar correlação de alertas de forma eficaz e precisa, sendo definidos da seguinte forma:

Alerts database: Alertas são a fonte de informação para qualquer técnica de correlação, pois os mesmos são gerados por diferentes equipamentos e/ou sistemas de detecção no ambiente monitorado. Alertas podem ser gerados por diversos tipos de sistemas, deste sistemas que fornecem o gerenciamento da rede, através do protocolo SNMP, até Sistemas de Detecção de Intrusão como Snort ou Suricata, abordados em capítulo anterior.

Topology information: Tem como finalidade representar a localização e a conectividade entre sensores, fornecendo uma representação precisa sobre a topologia da rede monitorada. A informação de topologia deve conter detalhes da rede e equipamentos em sua infra-estrutura, como switches, roteadores e servidores. Trabalhos de (STANIFORD-CHEN et al., 1996) e (YU et al., 2004) utilizam informações de agentes da rede para coletar informações sobre a topologia da rede armazenando em banco de dados.

Vulnerabilites database: Utilizada em ambientes de detecção de intrusão, esta base armazena vulnerabilidades conhecidas e geralmente com uma contra-medida correspondente para o ataque. É construído através da analise de informações sobre configuração dos recursos monitorados, como sistema operacional ou serviços de rede potencialmente suscetíveis a serem explorados por atacantes. (STANIFORD-CHEN et al., 1996) utiliza a base pública de vulnerabilidades proposta pela *Common Vulnerability and Exposures* . A base CVE caracteriza-se como uma lista contendo vulnerabilidades de segurança que tem como objetivo fornecer uma nomenclatura comum para problemas conhecidos publicamente.

Ontology database Uma base de ontologias fornece construções poderosas e é uma ferramenta útil para lidar com o conhecimento diversificado, como alertas. Consiste em pares atributo-valor para diferentes tipos de nós que fornecem uma visão global necessária para o processo de construção do conhecimento. Em (LI; TIAN, 2009) é proposto um sistema de

correlação de alertas baseado em uma base de ontologia em conjunto com o formato IDMEF.

Cases database: Um caso é a descrição de um problema conhecido, assim como alertas associados e as soluções. Esta fonte de informação é utilizada principalmente na análise de mensagens de erro. Assim, cada caso é descrito com dois elementos diferentes: situação e solução. A situação descreve o contexto do caso e a solução fornece razões sobre por que o erro ocorreu, sugerindo uma solução para resolve-lo. Em (HOLUB et al., 2009) é proposto um mecanismo de correlação em tempo-real para analisar registros de log na qual é utilizada uma base de casos para fornecer um mecanismo para combinar problemas conhecidos em um ambiente de grande volume de informações.

Knowledge representation: Permite incorporar o conhecimento humano no processo de correlação de alertas ou de alguma forma replicar esse conhecimento através de regras ou modelos. Estes podem ser definidos por especialistas ou inferidos a partir da análise de alertas através de procedimentos de aprendizagem. Como exemplo de seu uso no campo da segurança, (KABIRI; GHORBANI, 2005) propôs um sistema de correlação de alertas baseado em regras utilizando *Inference Engine* com objetivo de derivar a correlação entre alertas.

Quanto a Taxonomia de técnicas de correlação de alertas podem ser descritas como:

Number of data sources: Técnicas de correlação de alertas podem ser classificadas com base no número de fontes de dados utilizados. Desta forma, podem aceitar dados a partir de uma única ou múltiplas fontes de dados.

Type of application: As técnicas de correlação de alertas usualmente existentes são implementadas no contexto de uma aplicação. Apesar de seu uso potencial em muitos outros campos, segundo Salah (2013) existem três campos principais onde foram propostas e avaliadas estas técnicas: sistemas de gerenciamento de rede, segurança de TI e controle de processos em sistemas de produção (sistemas SCADA).

Correlation method: Segundo Salah (2013), existem três principais categorias utilizadas quanto ao método de correlação: *similarity-based, sequential-based and case-based methods.*. Técnicas baseadas em similaridade tem como objetivo a redução do número total de alertas por agrupamento, agregando alertas por suas semelhanças. Cada alerta gerado tem vários atributos associados ou campos, tais como: números de porta origem e destino de endereços IP, protocolos, descrição do alerta e informações *timestamp*. Este tipo de técnica podem ser agrupados nas categorias baseadas em atributos semelhantes ou baseadas em informações temporais. Em *Sequencial-based methods* os alertas são correlacionados utilizando as relações de causalidade

entre eles. São definidos pré-condições como os requisitos necessários para que o ataque seja bem sucedido, assim como as consequências são definidas como os efeitos que ocorrem após um ataque específico. Por fim, *Case-based methods* caracteriza-se por depender da existência de uma base de conhecimento utilizada para representar cenários bem definidos. A partir desta informação, são projetados métodos de mineração procurando padrões específicos. Quando um problema é resolvido com sucesso, a solução é armazenada em uma base de conhecimento chamada base de casos. Quando um novo problema é gerado, o sistema busca na base os casos mais semelhantes com os sintomas similares ao problema informado.

3.3 Arquitetura Proposta para Correlação de Alertas

Correlação de alertas é definida como uma interpretação conceitual de vários alertas, de tal forma que novos significados podem ser atribuídos a eles (JAKOBSON; WEISSMAN, 1993). Para Gardner (1996), correlação de alertas é uma interpretação de múltiplos alarmes de modo a aumentar o conteúdo da informação semântica associada com um conjunto reduzido de mensagens. Alertas, neste contexto, também possuem referências na literatura como sendo alarmes ou simplesmente eventos. São mensagens curtas, com um formato de texto específico definido pelos fornecedores, e gerado como uma manifestação externa de uma falha potencial ou um defeito que ocorre em algum recurso da rede ou sistema. Normalmente, esses alertas contém informações sobre o dispositivo de emissão e o evento em si, ou seja, o tempo de criação e recepção, uma descrição da falha, a gravidade do alerta, etc. Além disso, os alertas podem fornecer informações com diferentes níveis de detalhes, como dados específicos sobre o estado dos dispositivos e suas configurações, ou detalhes de nível superior, com informações agregadas recolhidas a partir de vários alertas.

A maioria das técnicas de correlação de alertas existentes são colaborativas, o que significa que elas dependem de mais de uma fonte de informação a fim de proporcionar uma visão mais precisa e coerente sobre a rede monitorada (VALEUR; VIGNA, 2005). O custo de obtenção de melhores resultados quando múltiplas fontes de dados são usados adiciona uma maior complexidade em sistemas de correlação de alerta, principalmente à heterogeneidade das diferentes entradas. Além disso, eles precisam de uma quantidade extra de recursos, quando comparado com as técnicas de fonte de dados individuais.

Como o próprio nome representa, o principal objetivo de correlação de alertas é descobrir as relações entre alertas. Atacantes são propensos a lançarem uma série de ataques contra

seus alvos, IDS tradicionais possuem como característica principal gerar alertas isolados. Sendo assim, o componente de correlação de alertas deve ser utilizado para correlacionar alertas em seu contexto mais amplo, ou seja, correlacionar alertas oriundos de diferentes fontes de informação. Esta função tem como objetivo fornecer ao analista de segurança uma visão da Consciência Situacional da rede.

O componente de correlação de alertas tem outras funcionalidades, como por exemplo, realizar uma correlação de alertas entre diferentes tipos de sistemas de detecção, o mesmo deve ser capaz de confirmar a ocorrência de um determinado ataque. Por exemplo, um IDS baseado em rede detecta um ataque de estouro de *buffer* suspeito para obter acesso *shell* a um determinado servidor da rede. Mas, devido à sua limitação, o mesmo não sabe o que realmente está acontecendo dentro desse *host*. Enquanto isso, um sistema de IDS baseado em *host* implantado dentro do mesmo servidor é capaz de detectar um processo *shell* suspeito e gera um novo alerta. Portanto, correlacionando os alertas gerados pelos sistemas, o analista de segurança pode confirmar que um ataque de tentativa de acesso ao *shell* remoto está em andamento. Além disso, uma vez que cada tipo de sistema de segurança tem seus próprios pontos cegos, uma correlação pode ajudar a remover alertas falsos positivos.

Em trabalhos anteriores (PETRI et al., 2012, 2013; PETRI, 2013), foi descrita a KBAM como um modelo de dados de uma base de conhecimento para *Internet Early Warning System*. Como descrito na Subseção 2.2.1, a KBAM pode ser utilizada para a construção de uma Consciência Situacional do ambiente monitorado pois, ao utilizar-se do formato IDMEF em sua abordagem, a base de conhecimento pode ser inserida em qualquer infraestrutura de rede que possui algum sistema de segurança, desde que o mesmo tenha a capacidade de gerar alertas no formato IDMEF.

Nesta dissertação, é proposta uma arquitetura de correlação de alertas com o objetivo de relacionar e detectar intrusões, sendo esta uma extensão do modelo da base de conhecimento KBAM (PETRI, 2013). Como pode-se observar na Figura 3.2, a base de conhecimento KBAM continua sendo utilizada. Proposto anteriormente como receptor e centralizador de alertas IDMEF, a aplicação Prelude-Manager foi extinta, sendo substituída pela própria base de conhecimento KBAM. Além disso, por neste contexto de trabalho não ser relevante, extinguiu-se a possibilidade de armazenamento dos contadores dos pacotes do tráfego de rede, assim como a classe responsável pela resposta de incidentes, o IDREF, já que este trabalho tem como objetivo descrever a forma de armazenar, tratar e correlacionar de alertas.

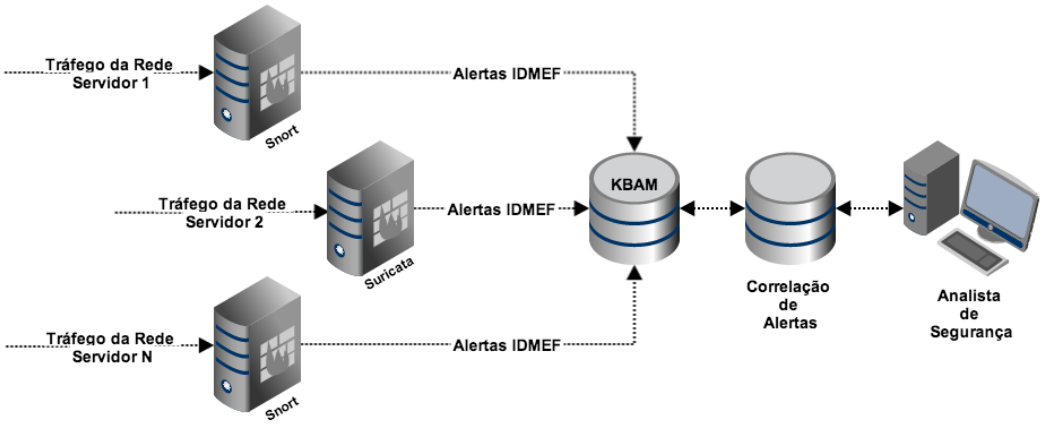

Figura 3.2 – Arquitetura de Correlação de Alertas para *Internet Early Warning System*.

3.4 Resumo

Este capítulo apresentou uma arquitetura para *Internet Early Warning System* baseada em Consciência Situacional. Como pode ser observado, Consciência Situacional e sua implementação no contexto de detecção de intrusão é pouco explorada na literatura. O modelo proposto por Endsley (1995), estendido por Golling (2011) e contextualizado para detecção de intrusão por Tadda (2010) e Onwubiko (2009) são fundamentais para a constituição de um *Internet Early Warning System*. Desta forma, uma visão geral da arquitetura de Correlação de Alertas foi apresentada, incluindo uma breve descrição de cada um dos seus componentes. Ao utilizar uma arquitetura baseada nos conceitos de Consciência Situacional, essa dissertação objetiva prover uma arquitetura para correlação de alertas em um *Internet Early Warning System* extensível, assim como demonstrar uma implementação da mesma utilizando *Case-Based Reasoning*.

4 CORRELAÇÃO DE ALERTAS EM UM IEWS

Enquanto IDSs são ferramentas eficientes no contexto tradicional de detecção de intrusão, elas não são eficazes o suficiente para a criação de uma Consciência Situacional do ambiente monitorado. Um atacante pode executar sua rotina de vigilância durante intervalos de tempo, podendo evitar a capacidade de detecção dos IDS, que tradicionalmente são incapazes de relacionar alertas. Além disso, um atacante inteligente pode criar um aplicativo que explore algum tipo de vulnerabilidade obtida através da Internet que não coincide com as assinaturas existentes ou comportamento detectável da rede.

Nesse contexto, apesar de um atacante muitas vezes deixar vários traços em diferentes pontos da rede durante uma tentativa de intrusão, a maioria dos IDSs tradicionais consideram cada ataque como uma tentativa de intrusão independente. Evidências de diferentes tipos de ataques contra a rede e seus recursos podem estar espalhados ao longo de vários servidores. Um sistema *Internet Early Warning System* deve ser capaz de coletar e relacionar informações de alertas em diferentes fontes e detectar situações de ataque. O processo de coleta e relacionar informações de alertas é chamado correlação de alertas.

Este capítulo esta organizado da seguinte maneira. A próxima seção descreve uma arquitetura para correlação de alertas que foi desenvolvida nesta dissertação e que se insere na arquitetura de *Internet Early Warning System* descrita no capítulo anterior. Para implementar esta arquitetura de correlação, um conjunto de pre-processamento para correlação é descrito na Seção 4.2. Na Seção 4.3 é apresentado o método de correlação proposto nesta dissertação.

4.1 Arquitetura de Correlação

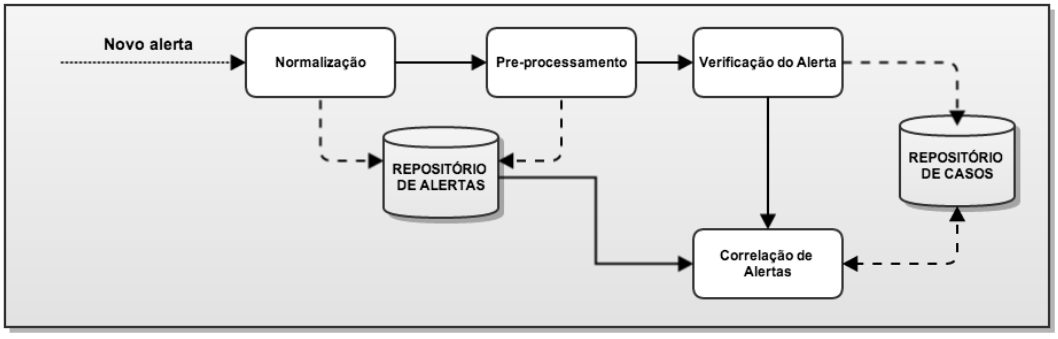

Figura 4.1 – Correlação de Alertas na arquitetura *Internet Early Warning System* proposta.

A Figura 4.1 apresenta o processo de correlação desenvolvido. As etapas de normalização e pré-processamento são aplicadas em todos alertas recebidos. A normalização tem como objetivo garantir que a padronização do formato IDMEF seja respeitada. O componente de pre-processamento tem como objetivo garantir que todos os atributos necessários para o cálculo de similaridade estejam presentes, ou seja, é utilizado para garantir que os atributos mais relevantes de cada alerta, informações como *timestamp*, tipo de ataque e endereços IP do atacante e seu alvo estejam disponíveis no alerta informado. O componente de verificação é utilizado para uma inspeção preliminar em algumas características contidas em cada novo alerta gerado, como classificação e tipo de ataque. Este componente também é utilizado para criação inicial da base de casos. A base de conhecimento KBAM neste contexto tem como propósito desempenhar um repositório de alertas no formato IDMEF. Já o repositório de casos é composto por alertas classificados como efetivamente um ataque real à infra-estrutura da rede. Inicialmente, para abastecer a base de casos, foram utilizados alertas classificados como *portscan*. O componente de correlação, responsável pelo cálculo de similaridade entre alertas e a base de casos é descrito a seguir.

4.2 Normalização e Pré-processamento

As subseções a seguir tem como objetivo descrever as etapas de padronização necessárias tanto no momento da geração do alerta no sensor IDS, quanto no recebimento e armazenamento do alerta no *Internet Early Warning System.*

4.2.1 Normalização

A etapa de normalização é utilizada para converter fontes heterogêneas de alertas em um formato padrão, que seja aceitável por outros módulos de correlação. O processo de correlação pode receber alertas de sensores diferentes, sendo assim, o objetivo do componente normalização de alertas é traduzir todos os atributos de cada sensor em um formato comum.

Com o objetivo de padronizar a representação dos alertas gerados por distintos IDS, utilizou-se o formato *Intrusion Detection Message Exchange Format*. Segundo Li (2009), o formato IDMEF é um exemplo relevante e amplamente utilizado, sendo um tipo de linguagem de comunicação que usa uma representação orientada a objetos para modelar os dados de alertas gerados por *IDS*. Foi implementado pelo *Internet Engineering Task Force* (IETF), em coopera-

ção com o grupo de trabalho *Intrusion Detection Working Group (IDWG).* Um dos principais objetivos do modelo IDMEF é ser capaz de expressar as relações entre alertas, assim como definir os procedimentos de troca e compartilhamento de informações que sejam de interesse para os sistemas de detecção e/ou prevenção de intrusão. O *Document Type Definition* (DTD) foi proposto para descrever o formato de dados do modelo IDMEF, sendo implementado em Extensible Markup Language (XML).

```
<?xml version="1.0" encoding="UTF-8"?>

<idmef:IDMEF-Message version="1.0"
            xmlns:idmef="http://iana.org/idmef">
  <idmef:Alert messageid="981012">
    <idmef:Analyzer analyzerid="2">
      <idmef:Node category="http">
   <idmef:location>CPD - UFSM</idmef:location>
   <idmef:name>snort-2</idmef:name>
      </idmef:Node>
    </idmef:Analyzer>
    <idmef:CreateTime ntpstamp="0xbc72b2b4.0x00000000">
      2013-07-06 23:17:58
    </idmef:CreateTime>
    <idmef:Source ident="abc01">
      <idmef:Node ident="abc01-01">
   <idmef:Address ident="abc01-02" category="ipv4-addr">
     <idmef:address>113.107.205.57</idmef:address>
   </idmef:Address>
      </idmef:Node>
    </idmef:Source>
    <idmef:Target ident="def01">
      <idmef:Node ident="def01-01" category="http">
   <idmef:name>sucuri.cpd.ufsm.br</idmef:name>
   <idmef:Address ident="def01-02" category="ipv4-addr">
     <idmef:address>200.18.33.57</idmef:address>
   </idmef:Address>
      </idmef:Node>
      <idmef:Service ident="def01-03">
   <idmef:portlist>21</idmef:portlist>
      </idmef:Service>
    </idmef:Target>
    <idmef:Classification text= portscan">
      <idmef:Reference origin="vendor-specific">
   <idmef:name>portscan</idmef:name>
   <idmef:url>http://www.vendor.com/portscan</idmef:url>
      </idmef:Reference>
    </idmef:Classification>
  </idmef:Alert>
</idmef:IDMEF-Message>
```

Figura 4.1 – Exemplo de um alerta no formato IDMEF.

Um exemplo da representação de um alerta enviado através do formato IDMEF é descrito na Figura 4.1. Como características deste ataque, pode-se destacar o seu identificador único *(Alert messageid)*, identificador único do sensor e respectivo nome *(Analyser analyse-*

rid) e (idmef:name), localização *(location)* física, tempo de detecção (idmef:CreateTime). O elemento *idmef:Address* tanto no atributo *idmef:Source* quanto *idmef:Target*, representam o endereço IP do atacante e da vítima, respectivamente. A classificação do alerta, assim como uma possível referência online da mesma são armazenadas nos elementos *idmef:Classification Text* e *idmef:url*, respectivamente.

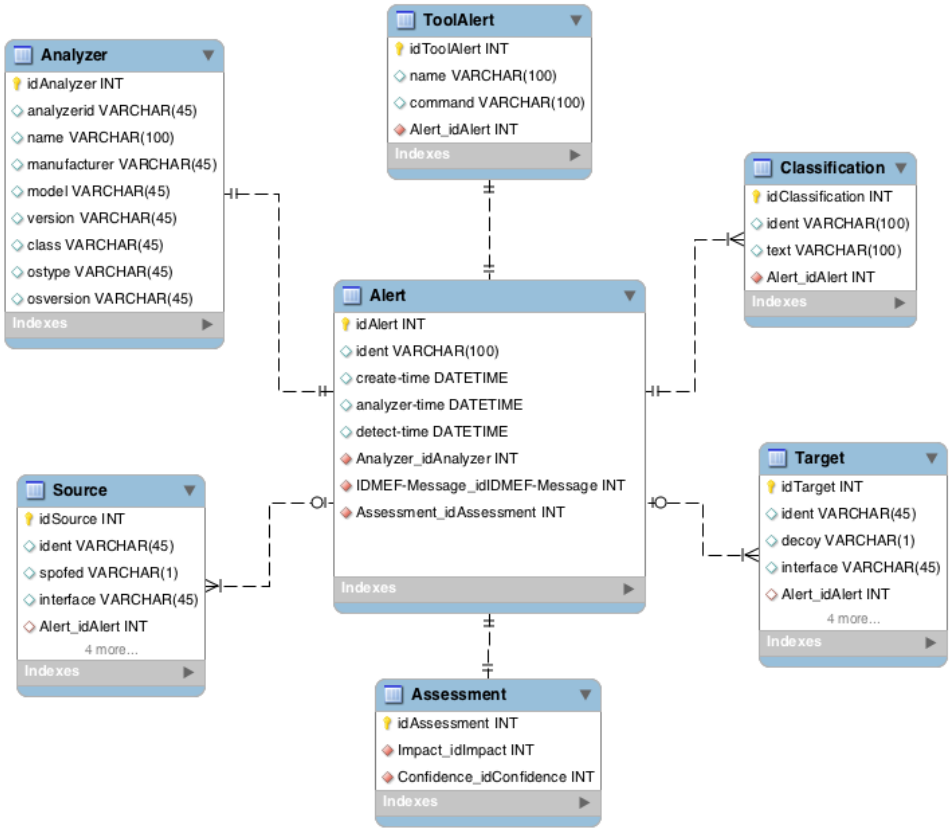

Figura 4.2 – Principais classes da KBAM.

Nesse sentido, a Figura 4.2 apresenta as principais classes utilizadas para armazenar alertas gerados por sistemas de detecção de intrusão na base de conhecimento KBAM. Segundo Petri (2013), pode-se descrever os atributos da classe *Alert* da seguinte maneira. O atributo *ident* armazena um identificador auto-incrementável para o alerta. *Create-time* representa o atributo referente ao instante da criação do alerta, o atributo *analyzer-time* armazena o momento em que o alerta foi disparado, já o instante em que o evento foi detectado está no atributo *detect-time*. Na arquitetura de correlação de alertas proposta neste trabalho, é relevante o relacionamento da classe *Alert* com as seguintes classes: *Assessment, Analyzer, Target, Source, ToolAlert e Classification*, descritas a seguir:

1. *Assessment:* Armazena as informações que permitem uma avaliação do evento causador

do alerta, como nível de impacto e de confiança do alerta.

2. *Analyzer:* Armazena informações referentes a identificação do IDS que gerou o alerta.

3. *Target:* Classe que armazena informações sobre o alvo do atacante.

4. *Source:* Classe que armazena informações sobre a possível origem do atacante.

5. *ToolAlert:* Contém informações referentes a alertas de ataques gerados por programas ou ferramentas. É quando o analisador consegue identificar qual a ferramenta ou programa foi o causador do ataque.

6. *Classification:* Representa uma possível classificação do tipo de ataque detectado.

Dessa forma, cada alerta recebido é traduzido e seus atributos são alimentados nos campos apropriados na base de conhecimento KBAM, conforme definido no diagrama de classes. A KBAM, proposta anteriormente por PETRI et al. (2012, 2013) utilizava-se de um framework chamado Prelude para coletar e armazenar alertas. O framework Prelude é um sistema gerenciador de eventos de segurança da informação que permite a unificação de vários tipos de aplicações e sensores em um sistema centralizado. A base de conhecimento KBAM possui muitas semelhanças no que tange à arquitetura de armazenamento de alertas utilizada no Prelude-Manager. Com o objetivo de evitar redundância de informações, desperdício de recursos de máquina e garantir uma flexibilidade no formato IDMEF em trabalhos futuros, optou-se pelo fim do intermédio do framework Prelude. Para isso, um *parser* de *log*, inicialmente compatível com os sistemas de detecção de intrusão *Snort* e *Suricata*, foi desenvolvido e instalado nos sensores responsáveis pela geração de alertas. Esse parser coleta e envia as informações para o módulo que insere os dados na base de conhecimento KBAM. Atualmente, esse software desenvolvido utiliza conexão TCP/IP não encriptada para comunicação.

4.2.2 Pré-processamento de Alertas

Como descrito na seção anterior, o processo de normalização deve ser eficaz na padronização de alertas. A realização do pré-processamento de alertas é necessário, pois em alguns casos, atributos como tempo, necessários para o processo de correlação, não são informados por alguns sensores. Assim, o objetivo do pré-processamento é preencher tais atributos necessários para a correta correlação de alertas.

Tradicionalmente um alerta no formato IDMEF pode conter três atributos do tipo *timestamp* distintos para a representação de tempo, são eles:

1. *AnalyserTime*: Representa o momento em que o IDS envia o alerta ao IEWS;
2. *DetectTime*: Representa o momento em que o IDS acredita que o evento aconteceu;
3. *CreateTime*: Representa o momento em que o alerta é criado pelo IDS.

Como descrito na Figura 4.3, o algoritmo é utilizado preferencialmente sobre o atributo *DetectTime*, seguido por *CreateTime* e então, *AnalyserTime*. Tem como objetivo de garantir uma maior precisão na correlação de alertas e de evitar inconsistências de *timestamp*. Recomenda-se a utilização de algum protocolo de sincronização de relógios NTP (NTP.BR, Acesso em 10/08/2013, Disponível em http://ntp.br) em todos sensores que de alguma forma fazem parte do *Internet Early Warning System.*

Input: ***alert***
Output: ***alert.time***

```
1:  if alert.Time is null: then
2:      if alert.DetectTime is not null: then
3:          alert.Time ← alert.DetectTime
4:      else if alert .CreateTime is not null: then
5:          alert.Time ← alert.CreateTime
6:      else if alert .AnalyzerTime is not null: then
7:          alert.Time ← alert.AnalyzerTime
8:      else
9:          alert.Time ← alert.ReceivedTime
10:     end if
11: end if
```

Figura 4.3 – Algoritmo para o pre-processamento de Alertas.

Na Tabela 4.1 é exibido o resultado do processo da etapa de normalização. Gerado a partir de fontes heterogêneas de sistemas de segurança, atributos como tempo, necessário para o correto armazenamento e consequentemente correlação de alertas, estão disponíveis de forma uniformizada.

4.2.3 Verificação

Construído especificamente para alimentar uma base com ataques reais para posterior comparação, a etapa de verificação é utilizada para realizar uma rotina que analisa o tipo de

Tabela 4.1 – Resultado do pre-processamento.

Alert ID	Analyser ID	Analyser Name	Alert Time	Recebido de
970690	2	snort-coralx-2	2013-07-06 15:10:41	**DetectTime**
981012	2	snort-coralx-2	2013-07-06 23:17:58	**CreateTime**
109602773	5	suricata-coralx-1	2013-12-04 23:26:17	**AnalyserTime**

classificação de cada alerta recebido. Pode-se utilizar diversos critérios para definir o que é considerado um ataque real, mas dificilmente com garantia de baixa quantidade de ataques falsos positivos. Nesse sentido, como descrito na Figura 4.4, o algoritmo de verificação utilizado neste trabalho considera que qualquer alerta classificado pelos sensores como *portscan* é o primeiro passo que um atacante realiza e por isso o mesmo é inserido automaticamente na base de casos para futura correlação com um novo alerta. Este comportamento será discutido em maiores detalhes na Seção 4.3.2 que trata sobre cálculo de similaridade.

Input: ***alert***
Output: ***cases***
1: **if** classification.text = '(portscan) TCP Portscan' **then**
2: $cases.add(alert)$
3: **end if**

Figura 4.4 – Algoritmo de verificação de alertas.

4.3 Correlação de Alertas

Técnicas baseadas em cálculo de similaridade, como CBR, visam reduzir o número total de alertas por agrupamento e agregá-los usando suas semelhanças. Cada alerta gerado neste trabalho possui atributos associados ou campos, tais como: endereços IP, números de porta, tipo de serviço e protocolos utilizados, descrição de alerta e informações de *timestamp*. A principal hipótese é que alertas semelhantes tendem a possuir as mesmas causas ou efeitos semelhantes no sistema monitorado. Definir pesos para o cálculo de similaridade é um problema crítico para esse tipo de técnica. Sendo assim, o objetivo é definir uma função de similaridade adequada para cada atributo, já que cada atributo em um alerta pode possuir diferentes pesos e por consequência, diferentes efeitos sobre o processo de correlação.

Segundo Salah (2013), técnicas que utilizam o cálculo de similaridade possuem muitas vantagens. Primeiro, elas geralmente são implementados com algoritmos leves e de menor complexidade, principalmente porque estes algoritmos se basearem em comparações lógicas

simples. Em segundo lugar, esta categoria provou a sua eficácia na redução do número total de alertas, que é um passo fundamental no processo de correlação, dado o grande número de alertas gerados.Técnicas para a cálculo de similaridade podem ser divididas em duas categorias: baseadas em atributos ou em informações temporais. Deste modo, optou-se por utilizar a técnica CBR para o módulo de correlação de alertas nessa dissertação. A seguir é apresentada a técnica CBR e logo após, é apresentada a aplicação de CBR para o contexto de *Internet Early Warning System.*

4.3.1 Case-Based Reasoning

Case-Based Reasoning – CBR é uma técnica que busca soluções para problemas atuais em soluções encontradas no passado, baseando-se em uma das principais características do ser humano, a memória. Segundo Wangenheim (2003), sistemas de CBR tem como objetivo resolver novos problemas utilizando e adaptando experiências anteriores contidas em um repositório de experiências concretas de soluções de problemas, denominada base de casos. Na forma mais simplificada, um caso é composto por três elementos: uma descrição do problema, uma solução e uma avaliação da solução. Em geral, o ciclo de CBR consiste em quatro etapas: recuperar (*retrieve*), reutilizar (*reuse*), revisar (*revise*) e reter (*retain*).

Descrito na Figura 4.5, o ciclo CBR proposto por Aamodt (1994) é considerado um formato completo que permite modelar os principais passos de um sistema CBR, sendo representado por um ciclo de raciocínio que pode ser contínuo. Este ciclo é composto pelas tarefas de recuperar, reutilizar, revisar e reter um caso. De acordo com o problema informado, ou novo caso é usado como consulta no sistema CBR, a base de casos é pesquisada para buscar problemas anteriormente resolvidos. Este processo de busca é realizado de acordo com o nível de similaridade entre atributos do novo problema e da base de casos (WANGENHEIM; WANGENHEIM, 2003). Em resumo, a partir da necessidade de resolver determinado problema, esta etapa de recuperação realiza uma busca na base de casos. Como resultado, a etapa de recuperação seleciona quais casos podem conter soluções relevantes (ou reusáveis) para a solução do novo problema, tomando como referência o nível de similaridade entre o problema atual e os casos da base de casos. Aamodt (1994) descreve que a tarefa de recuperação de casos inicia com a descrição de um problema e termina quando um caso mais similar é encontrado.

Em um primeiro momento, é realizado uma triagem em experiências passadas, e em seguida, as mesmas são armazenadas no sistema, considerando estas experiências passadas como

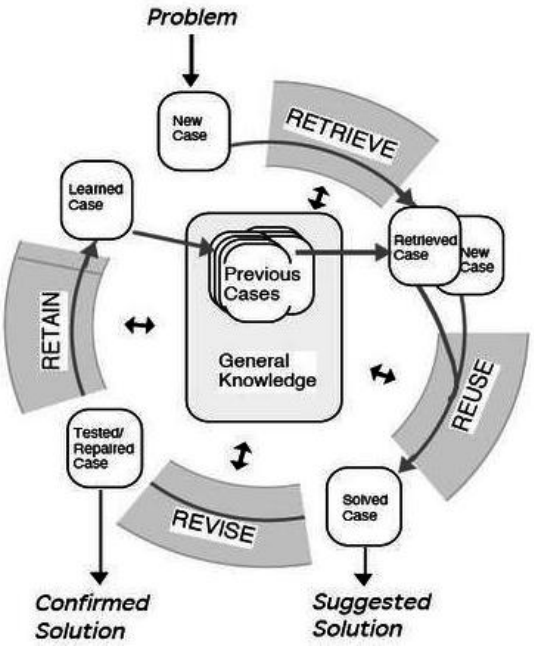

Figura 4.5 – Ciclo CBR descrito por Aamodt (1994).

casos. Durante o processo de raciocínio para a resolução de uma nova situação, cada novo problema informado é comparado aos casos armazenados na base de casos. Os casos mais similares são utilizados para propor uma solução ao problema informado.

Técnicas de *CBR* tem diversas vantagens sobre outros paradigmas de raciocínio. Uma delas diz respeito à facilidade na aquisição do conhecimento, que é realizada buscando experiências reais de situações passadas (ESMAILI, 1996). Outra vantagem é a possibilidade de se obter uma relação parcial entre a nova situação e os casos, permitindo maior flexibilidade em domínios na qual os atributos e as condições do problema podem ter pequenas variações ao ocorrerem em situações reais.

No processo de recuperação de casos, uma métrica de similaridade é uma função que permite avaliar analiticamente os graus de similaridade entre dois casos. Usualmente, são atribuídos pesos diferentes a cada uma das características de um caso. No intuito de combinar as similaridades medidas entre cada um dos atributos representados nos casos, ou similaridades locais, e métodos de agregação como a média ponderada aplicada a valores de similaridades locais são utilizados para gerar um valor global de similaridade entre dois casos. Neste processo, o valor de cada peso é diretamente proporcional à importância de cada atributo definido na estrutura de um caso. A determinação numérica de pesos é geralmente definida como resultado de um processo gradual de ajuste de pesos e consequente avaliação da performance do sistema. Em geral, este processo é caracterizado como um processo de tentativa e erro orientado por

resultados de *precision and recall*.

Um caso recuperado pode ser útil para a solução de determinado problema quando a métrica de similaridade entre o novo problema informado e o caso recuperado da base é alta. Um ou vários casos podem ser recuperados, cabendo ao algoritmo de CBR determinar a melhor solução.

Diante do problema referido, é proposta uma abordagem que explora a técnica de CBR para identificar de forma automática, ataques reais dentre todos alertas gerados. A abordagem fornece condições para que cenários de intrusão possam ser modelados como casos e assim, sempre que características semelhantes se repetirem em novos alertas, o algoritmo deve ser capaz de identificá-los e notificar o analista de segurança.

Devido a grande quantidade de alertas gerados por IDS tradicionais, em sua grande maioria das vezes ataques falsos-positivos, ataques legítimos passam despercebidos pelo analista de segurança. A abordagem neste capítulo propõe a utilização do paradigma CBR para identificar cenários de intrusão de forma automática, correlacionando alertas gerados por fontes heterogêneas. No contexto de detecção de intrusão, um caso utiliza como base os valores definidos no modelo KBAM, utilizando-se do formato IDMEF.

4.3.2 Cálculo de Similaridade entre Atributos dos Alertas

O cálculo de similaridade entre um novo alerta e um caso armazenado na base de casos é feita em duas formas: (a) similaridade local, calculando a similaridade entre cada atributo do novo alerta e cada atributo de todos casos da base de casos e; (b) similaridade global, combinando as médias de similaridade locais do cálculo anterior, compara-se a média de similaridade entre o novo caso e todos casos da base de casos.

Com objetivo de criar um repositório inicial de casos com ameaças reais dentro da base de conhecimento KBAM, a ser utilizado como parâmetro de comparação, utilizou-se todos os alertas classificados pelo sistema de detecção de intrusão como *portscan*, por considerar que tal técnica é uma das possíveis formas utilizadas por atacantes reais de iniciar um ataque a infraestrutura da rede e que são facilmente detectados por IDS tradicionais. *Portscan* tem como objetivo testar as portas lógicas de determinado computador remoto. Neste teste, o atacante verifica o status das portas, se estão fechadas, escutando ou abertas. Técnicas de *portscan* são utilizadas por atacantes com objetivo de verificar a possibilidade de explorar alguma vulnerabilidade em ataque posterior, que se realizado e um novo alerta disparado, será utilizado para o

cálculo de similaridade e destacado em meio de falsos-positivos, como descrito na Figura 4.6.

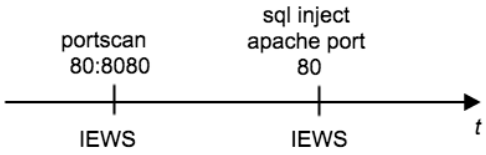

Figura 4.6 – Ataque em dois estágios.

Entre todos atributos contidos em um alerta que podem empregados no cálculo de similaridade e que são armazenados no formato IDMEF dentro da base de conhecimento KBAM, foram considerados como atributos relevantes para a estrutura de um caso:

Analyser ID: Identificação do sensor que reportou o alerta;

DetectTime: Instante da detecção do alerta;

Classification: Classificação do ataque;

Source IP: Possível Endereço IP do atacante;

Target IP: Endereço IP da vítima;

Target Port: Porta lógica do serviço afetado;

Service Protocol: Tipo de protocolo de serviço utilizado.

Um caso pode conter um ou mais atributos com diferentes níveis de importância, conforme as características do cenário de intrusão ou da atividade suspeita, ou seja, atributos podem possuir pesos diferentes, conforme descrito na Tabela 4.2.

Tabela 4.2 – Pesos dos atributos.

Atributo	**Peso**
Analyser ID	2
Detection time	3
Source IP Address	6
Target IP Address	4
Target Port	4
Service Protocol	2
Classification	3
Classification Type	3

Um exemplo de um ataque do tipo *portscan* armazenado como caso é apresentado na Tabela 4.3. O caso descreve cada atributo utilizado para calcular a similaridade entre novos alertas recebidos.

Tabela 4.3 – Representação de um Caso.

Atributo	**Caso A**
Alert ID	970690
Analyser ID	snort-coralx (1)
Detection time	2013-07-12 11:58:47
Source IP Address	113.107.205.57
Target IP Address	200.18.33.52
Target Port	21:80
Service Protocol	TCP
Classification	(portscan) TCP Portscan
Classification Type	13

Quando um novo alerta é gerado, o mesmo é correlacionado medindo-se a similaridade local entre cada um dos atributos do novo alerta e os atributos de cada caso em toda a base de casos. O resultado do cálculo de similaridade local é exemplificado na Tabela 4.4.

Tabela 4.4 – Cálculo de similaridade local de cada atributo entre um caso e alerta.

Atributo	**Caso A**	**Novo Alerta**	*Sim*
Alert ID	970690	981012	-
Analyser ID	snort-coralx (1)	snort-coralx (1)	1
Detection time	2013-07-06 15:10:41	2013-07-06 23:17:58	0.6
Source IP Address	113.107.205.57	113.107.205.57	1
Destination IP Address	200.18.33.57	200.18.33.57	1
Destination Port	21:80	21	1
Service Protocol	TCP	TCP	1
Classification	(portscan) TCP Portscan	Invalid FTP Command	0
Classification Type	13	15	0

Com o objetivo de combinar medidas locais de similaridade para o cálculo de similaridade global, é realizado uma soma ponderada (pelos pesos) de similaridades locais para todos os atributos, descrito na Equação 4.1.

$$Sim(f^l, f^r) = \frac{\sum_{i=1}^{n} w_i \times sim_i(f_i^l f_i^r)}{\sum_{i=1}^{n} w_i} \tag{4.1}$$

Onde w_i representa o peso do atributo i, f_i^l é o valor do atributo i para o caso da base, f_i^r é o valor do atributo i para o alerta em análise e $sim_i(f_i^l, f_i^r)$ é a função de similaridade do atributo i.

Por fim, é realizado um laço de repetição entre cada novo alerta e todos os casos da base de casos, como descrito na Figura 4.7.

Input: ***alerta, casos***
Output: ***casos, proximo***

```
1: for i in casos do
2:     if sim(alerta, casos(i)) > 90% then
3:         add casos_similares(i)
4:     end if
5: end for
```

Figura 4.7 – Algoritmo que cria um laço de repetição entre novos alertas e toda base de casos

Assim, ordena-se os resultados encontrados por ordem decrescente de similaridade. Obteve-se melhores resultados, possibilidade de um ataque real entre os casos que retornaram como similares, quando o resultado do cálculo de similaridade global entre o novo alerta e todos os casos for igual ou superior a 0.9. Ou seja, obteve-se os melhores resultados na performance do sistema o quando o valor de *threshold* foi igual ou maior que 90%. Assim utilizou-se essa métrica em todos experimentos. Na sequência, será descrito como é realizado o cálculo de similaridade local entre cada atributo do novo alerta e um caso da base de casos.

4.3.3 Cálculo de Similaridade entre Endereços IP

É utilizado para calcular a similaridade dos atributos *SourceIP* e *TargetIP* entre um novo alerta e todos os casos da base de casos. *SourceIP* e *TargetIP*, por conter o identificador tanto do possível atacante quanto do servidor atingido , são considerados um dos principais atributos disponíveis para o calculo de similaridade, e por isso, seu peso diferenciado.

Para quantificar a proximidade entre dois Endereços IP, quando existe um casamento entre os atributos, ou seja, quando o endereço IP do atributo *SourceIP* forem iguais tanto no novo alerta quanto no caso comparado, o valor de similaridade é 1. Quando não existe casamento de endereço IP é proposto o uso de Georreferenciamento de Endereços IP, descrito a seguir.

4.3.3.1 Georreferenciamento de Endereços IP

Georreferenciamento de Endereços IP (GeoIP) é método utilizado para determinar a localização no planeta de determinado endereço IP na rede. Existe uma série de base de dados GeoIP disponíveis, tanto gratuitas quanto comerciais. Em boa parte das situações consegue-se extrair informações como país, estado, cidade e provedor de acesso a Internet de determinado

endereço IP. Atualmente este recurso é muito utilizado em publicidade direcionada na Internet.

No método *diff* que é amplamente utilizado na literatura (ZHIHONG et al., 2008; ZHUANG et al., 2008; FAN; JIHUA; MIN, 2009) no cálculo de similaridade entre endereços IP, calcula-se a diferença bit-a-bit entre dois Endereços IP e divide-se por 32, ou seja, o número máximo de 1 bit em uma máscara de sub-rede IPv4. Nesse sentido, o GeoIP apresenta vantagens, sendo a principal delas em relação do método *diff* a não restrição do cálculo de distância, a apenas o escopo da máscara de rede. Assim, pode-se estender o cálculo de distância e dentre as informações disponíveis em uma base GeoIP, pode-se relacionar atacantes a níveis de país, cidade e até provedor de acesso. Um exemplo dos atributos que podem ser extraídos de uma base GeoIP e serem utilizados para o cálculo de distância são descritos na Tabela 4.5.

Tabela 4.5 – Exemplo do uso do GeoIP.

Endereço IP	67.202.95.200	113.107.205.57
Código	US	CN
Cidade	Chicago	Cantão
Estado	Illinois	Guangdong
País	Estados Unidos	China
Localização	América do Norte	Asia
Código Postal	60607	
Coordenadas	41.8745, -87.6503	23.1167,113.25
Provedor	Steadfast Networks	China Telecom Guangdong
Organização	Steadfast Networks	China Telecom Guangdong
Domínio	steadfastdns.net	
Código Metropolitano	602	

Na abordagem proposta, o cálculo de similaridade ocorre entre o atributo *address* da classe *kbam.address* e o atributo *address* da classe *kbam.cases*. O algoritmo utilizado para a definição dos pesos desta etapa é descrito na Figura 4.8. Limitou-se a utilização dos atributos GeoIP a nível de cidade, região, país e provedor de acesso.

Um exemplo onde são apresentados os resultados do cálculo de similaridade entre os atributos *address* de um novo alerta e dois casos é descrito nas Tabelas 4.6 e 4.7.

Input: $x \leftarrow kbam.address$
Input: $y \leftarrow kbam.cases$
Output: $Sim_{ip}(x, y)$
1: **if** $X.ip = Y.ip$ **then**
2: $Sim = 1$
3: **else if** $X.city = Y.city$ **and** $X.country = Y.country$ **and** $X.provider = Y.provider$ **then**
4: $Sim = 0.7$
5: **else if** $X.city = Y.city$ **or** $X.region = Y.region$ **then**
6: $Sim = 0.5$
7: **else if** $X.country = Y.country$ **then**
8: $Sim = 0.45$
9: **else if** $X.provider = Y.provider$ **then**
10: $Sim = 0.4$
11: **else**
12: $Sim = 0$
13: **end if**

Figura 4.8 – Algoritmo utilizado para o cálculo de similaridade entre Endereços IP.

Tabela 4.6 – Similaridade Local do atributo *address* ao receber Endereços IP idênticos.

Atributo	**Novo Alerta** (x)	**Caso 1** (y)	$Sim(x, y)$
Endereço IP	113.107.205.57	113.107.205.57	1
Cidade	Cantão		
Estado	Guangdong		
Pais	China		
Provedor	China Telecom Guangdong		

Tabela 4.7 – Similaridade Local do atributo *address* ao receber Endereços IP distintos.

Atributo	**Novo Alerta** (x)	**Caso 2** (y)	$Sim(x, y)$
Endereço IP	113.107.205.57	119.29.255.255	0
Cidade	Cantão	Pequim	0
Estado	Guangdong	Beijing Shi	0
Pais	China	China	0.45
Provedor	China Telecom Guangdong	Beijing SHUXUNDA	0

4.3.4 Cálculo de Similaridade entre *Timestamp*

Para realizar o cálculo de distância entre atributos de um novo alerta (t_1) e um caso (t_2) que referenciam uma unidade de tempo foi utilizado o algoritmo descrito na Figura 4.9. Converte-se do formato *timestamp* para *unixtime* e então calcula-se a distância euclidiana entre os valores. Para fins de referência, sendo a base de casos composta somente por alertas classi-

ficados como *portscan*, quando a distância entre os atributos for menor que 60 minutos (T_{min}) e não maior que (T_{max}) 24 horas, maior a precisão de acerto nos resultados obtidos.

Input: $t_1 \leftarrow kbam.alert$
Input: $t_2 \leftarrow kbam.cases$
Output: $Sim(t_1, t_2)$

1: **if** $|t_1 - t_2| < T_{min}$ **then**
2: $\quad Sim(t_1, t_2) = 1$
3: **else if** $|t_1 - t_2| > T_{max}$ **then**
4: $\quad Sim(t_1, t_2) = 0$
5: **else**
6: $\quad Sim(t_1, t_2) = 1 - (\frac{t_1 - t_2}{t_1})$
7: **end if**

Figura 4.9 – Cálculo de Similaridade entre *Timestamp*.

4.3.5 Cálculo de Similaridade entre Portas Lógicas

Representando uma ou mais portas lógicas no qual o atacante pode vir a analisar no servidor remoto, o atributo *port* integrante da classe *kbam.service* pode incluir várias portas em seu conteúdo. Dessa forma, o resultado do cálculo de similaridade pode depender da sobreposição de dois valores. Suponha-se que um novo alerta (x) a ser comparado com cada caso (y) contido na base de casos, a Equação 4.2 demonstra o calculo de similaridade adotado.

$$Sim_{port}(x, y) = \begin{cases} 1 & \text{if } x.port \subset y.port \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Um exemplo onde é apresentado os resultados do cálculo de similaridade comparando o atributo *port* entre um novo alerta (x) e dois casos (y_1 e y_2) é apresentado na tabela 4.8.

Tabela 4.8 – Cálculo de Similaridade Local do atributo *port* entre um novo alerta e dois casos.

Novo Alerta (x)	**Caso 1** (y_1)	$Sim(x, y_1)$	**Caso 2** (y_2)	$Sim(x, y_2)$
21	21-80	1	80-8080	0

4.3.6 Cálculo de Similaridade entre Sensores

O atributo *analyserid* representa a identificação do sensor que enviou o alerta, o cálculo é realizado entre a classe *kbam.analyserid*, atributo *analyserid*, tipo inteiro, e a classe *kbam.cases*, atributo *analyserid*, tipo inteiro. Conforme descrito na Equação 4.3, quando um novo alerta é

originado a partir do mesmo sensor (IDS) que alimentou o caso ao qual é comparado, o mesmo recebe o valor Similaridade Local neste atributo igual a 1, caso contrário igual a 0.

$$Sim_{sensor}(x, y) = \begin{cases} 1 & if\ x.analyserid = y.analyserid \\ 0 & otherwise \end{cases} \tag{4.3}$$

Um exemplo onde é apresentado os resultados do cálculo de similaridade comparando o atributo *analyserid* entre um novo alerta (x) e dois casos (y_1 e y_2) é apresentado na tabela 4.9.

Tabela 4.9 – Cálculo de Similaridade Local do atributo *analyserid* entre um novo alerta e dois casos.

Novo Alerta (x)	**Caso 1** (y_1)	$Sim(x, y_1)$	**Caso 2** (y_2)	$Sim(x, y_2)$
2	2	1	3	0

4.3.7 Cálculo de Similaridade entre Service Protocol

O atributo *Service Protocol* representa o protocolo de transporte utilizado pelo serviço que representa o incidente de segurança. O cálculo é realizado entre a classe *kbam.service*, atributo *protocol*, tipo inteiro, e a classe *kbam.cases*, atributo *protocol*, tipo inteiro. Conforme descrito na Equação 4.4, quando um novo alerta utilizou o mesmo protocolo de transporte do caso ao qual é comparado, o mesmo recebe o valor de Similaridade Local neste atributo igual a 1, caso contrário igual a 0.

$$Sim_{sensor}(x, y) = \begin{cases} 1 & if\ x.procotol = y.procotol \\ 0 & otherwise \end{cases} \tag{4.4}$$

Um exemplo onde é apresentado os resultados do cálculo de similaridade comparando o atributo *analyserid* entre um novo alerta (x) e dois casos (y_1 e y_2) é apresentado na tabela 4.10.

Tabela 4.10 – Cálculo de Similaridade Local do atributo *protocol* entre um novo alerta e dois casos.

Novo Alerta (x)	**Caso 1** (y_1)	$Sim(x, y_1)$	**Caso 2** (y_2)	$Sim(x, y_2)$
TCP	TCP	1	UDP	0

4.4 Resumo

Apesar de um atacante muitas vezes deixar vários vestígios em diferentes pontos da rede durante uma tentativa de intrusão, a maioria dos IDS consideram cada ataque de forma independente. Nesse sentido, um sistema *Internet Early Warning System* deve possuir a capacidade de centralizar diferentes fontes de alertas e detectar situações de riscos. Assim, este capítulo apresentou e descreveu os componentes necessários em uma arquitetura para correlação de alertas em um *Internet Early Warning System*. Dividido em duas etapas principais, pre-processamento e correlação, sendo que a primeira etapa tem como função normalizar os alertas recebidos em um formato padrão e completo. Já a etapa de correlação descreve passo-a-passo o cálculo de similaridade em cada atributo utilizando-se de recursos como a técnica *Case-Based Reasoning* e Georreferenciamento de IP para que, através do resultado da Correlação de Alertas entre sensores distribuídos, obtenha-se uma visão da Consciência Situacional da rede monitorada como um todo.

5 AVALIAÇÃO E RESULTADOS

Neste capítulo são apresentados os experimentos realizados para a validação da proposta de correlação de alertas apresentada no Capítulo 4. A avaliação é realizada através da implementação de uma arquitetura de correlação de alertas em um *Internet Early Warning System* dentro da rede da Universidade Federal de Santa Maria (UFSM), utilizando recursos como a técnica de *Case-Based Reasoning* e Georreferenciamento IP para o cálculo de similaridade. A Seção 5.1 descreve a configuração do ambiente utilizado na geração, processamento e armazenamento de alertas. Na Seção 5.2 é descrita a obtenção dos níveis de percepção e compreensão da Consciência Situacional do ambiente monitorado através da discussão dos experimentos realizados.

5.1 Configuração do Sistema

O processo de geração de alertas é realizado a partir de pontos estratégicos que fazem parte da infraestrutura de rede da Universidade Federal de Santa Maria (UFSM). Como exposto na Figura 5.1, os servidores escolhidos foram definidos a partir de critérios como a importância dos recursos que determinado servidor oferece, como portal do aluno ou biblioteca, e pelo seu possível caráter confidencial, como por exemplo processos seletivos e informações hospitalares.

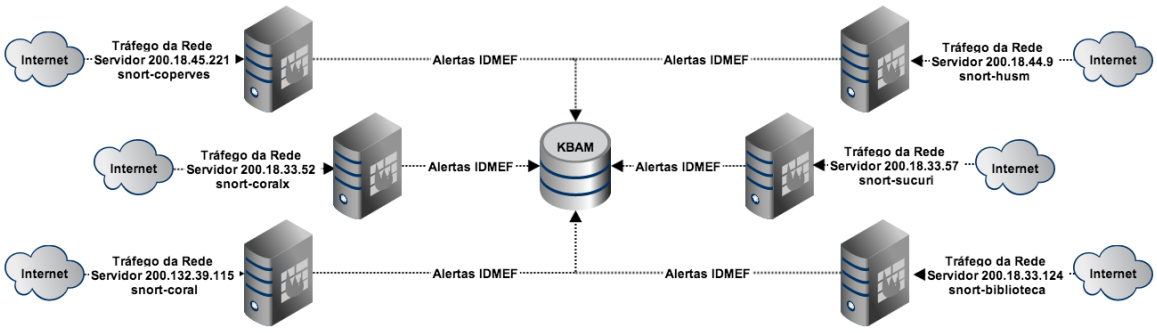

Figura 5.1 – Arquitetura para centralização de alertas na Base de Conhecimento KBAM.

Para a construção do IEWS, optou-se pela utilização do IDS Snort[1] para a geração de alertas analisando o tráfego de rede em cada um dos sensores anteriores. Sendo considerado um tradicional sistema de detecção de intrusão de código aberto que utiliza a biblioteca PCAP[2]

[1] Disponível em http://www.snort.org
[2] Disponível em http://www.tcpdump.org

para capturar e filtrar facotes da rede, o Snort realiza a inspeção de pacotes utilizando regras e assinaturas de ataques já conhecidos. A geração dos alertas utilizou o formato padronizado IDMEF e sua transmissão (do alerta) realizada através de um parser que armazena remotamente na base de conhecimento KBAM. A KBAM foi apresentada em trabalhos anteriores (PETRI et al., 2012, 2013; PETRI, 2013) e descrita como um modelo de dados de uma base de conhecimento para *Internet Early Warning System*.

5.1.1 Sensores Embarcados

Sensores são parte fundamental na estrutura de um *Internet Early Warning System* pois, como apresentado, a quantidade de sensores é diretamente proporcional ao nível de percepção da rede monitorada. Nesse sentido, com o objetivo de expandir a Consciência Situacional da rede, experimentou-se a utilização de dispositivos embarcados de baixíssimo custo como sensor IEWS.

Sensores embarcados tem como grande vantagem a sua relação de baixo custo pelo desempenho apresentado. Testou-se duas placas: O primeiro foi Raspberry Pi[3] com processador ARM11 de 700MHz, 512MB de RAM (*Random Access Memory*) e Sistema Operacional Raspian. A segunda placa foi o modelo CubieBoard AllWinner A20[4] com processador dual-core ARM Cortex A7 de 1GHz, 1GB de memória RAM e Sistema Operacional Cubian. Tanto Raspian quanto Cubian são baseados na distribuição GNU/Linux e ambos possuem entrada para cartão de memória e rede ethernet 10/100.

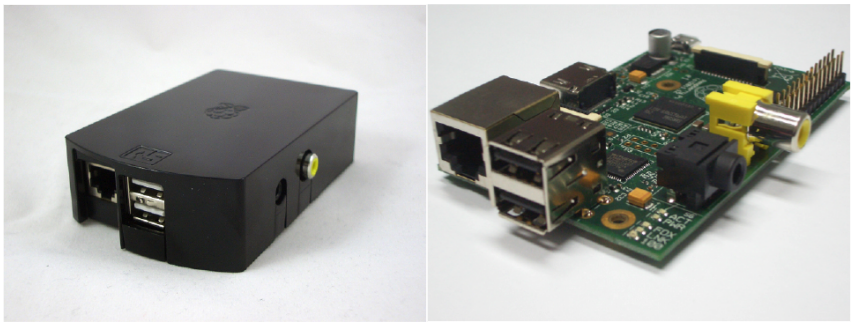

Figura 5.2 – Sensores Embarcados - Raspberry Pi.

Nos testes realizados, optou-se pela execução do sistema de detecção de intrusão Snort em sua versão 2.9, disponível de forma pré-compilada no repositório oficial de sua respectiva distribuição. Resultados: Constatou-se que em ambos equipamentos o Snort em sua configu-

[3] Disponível em http://www.raspberrypi.org/
[4] Disponível em http://cubieboard.org/

ração padrão perdeu pacotes em uma rede com tráfego de informações superior a 20Mbps e, com ajustes finos na configuração da aplicação, obteve-se cerca de 50Mbps sem perda de pacotes ao desabilitar o salvamento de logs da aplicação em disco. O desempenho semelhante entre as duas placas possivelmente aconteceu pelo fato a aplicação Snort não aproveitar de processadores multi-thread. Através de tais resultados, constatou-se que equipamentos deste tipo pelo seu baixo preço, podem ser utilizados em grande quantidade para monitorar a rede e abastecer a base de conhecimento do IEWS com uma grande quantidade de sensores espalhados geograficamente em qualquer lugar da infraestrutura da rede.

5.2 Construindo a Consciência Situacional do Ambiente Monitorado

Os três níveis de Consciência Situacional apresentado por Endsley (1995) são referência no assunto e, desde então, vem sendo aplicado para diversas áreas da ciência. Tadda (2010), utilizando o modelo proposto e de sua extensão por McGuinness e Foy (2000), extrapolaram algumas de suas características, propondo um modelo de Consciência Situacional para domínio de cyber-segurança. Este trabalho tem como objetivo atingir o primeiro e segundo níveis da consciência situacional, ou seja, perceber e compreender o ambiente da rede monitorada.

5.2.1 Percepção

O primeiro nível da consciência situacional refere-se a percepção do ambiente, ou seja, ao conhecimento sobre todos os elementos de dentro da rede que analistas de segurança devem estar cientes, como por exemplo, alertas relatados por sistemas de detecção de intrusão (IDS). O primeiro nível de consciência situacional foi atingido com a utilização de diversos sensores, cada qual analisando tráfego da rede em pontos geograficamente diferentes e gerando alertas de tentativas de intrusão em um formato padronizado, sendo os alertas de todos sensores armazenado em uma única na base de conhecimento KBAM.

5.2.2 Compreensão Básica

Segundo Tadda (2010), o nível de Compreensão refere-se a técnicas, metodologias, processos e procedimentos utilizados para analisar, sintetizar e correlacionar informações percebidas na rede a partir de elementos da própria rede. Uma grande quantidade de dados são armazenados na base de conhecimento KBAM e, quando combinados podem gerar novas infor-

mações. Nesse sentido, perguntas como quais os principais servidores são os alvos de atacantes ou quais os principais tipos de classificação de ataques são realizados contra a infraestrutura da rede podem ser respondidas com facilidade, conforme exposto nas Tabelas 5.1 e 5.2. Com isso, alguns requisitos necessários para segundo nível da consciência situacional são preenchidos.

Tabela 5.1 – Principais alvos na rede monitorada.

Sensor	**Alertas**
coralx	780409
sucuri	308394
coperves	272312
biblioteca	224343
coral	194516
husm	48643

Tabela 5.2 – Principais alertas reportados na rede monitorada.

Classificação do Ataque	**Alertas**
COMMUNITY WEB-MISC mod_jrun overflow attempt	1900686
SHELLCODE x86 NOOP	662558
NETBIOS SMB-DS Session Setup username overflow attempt	341188
WEB-PHP remote include path	97253
NETBIOS SMB-DS repeated logon failure	65885
WEB-ATTACKS rm command attempt	22773
WEB-MISC weblogic/tomcat .jsp view source attempt	21607

Em contrapartida, ter compreensão sobre a rede monitorada também significa descobrir as possíveis relações existentes entre alertas recebidos. Atacantes são propensos a lançarem uma série de ataques contra seus alvos em períodos de tempo diferentes e IDS por padrão geram alertas de forma isolada. A seguir, a noção de compreensão através de correlação de alertas, conforme apresentado no capítulo anterior é apresentada.

5.2.3 Compreensão através da Correlação de Alertas

Com o objetivo de prover um nível de compreensão mais expressivo, além da compreensão básica que passa somente por filtros de dados para a obtenção de alvos e classificações de alertas, agora serão realizados experimentos demonstrando a aplicabilidade do modelo de correlação de dados discutido no Capítulo 4.

Conforme apresentado no capítulo anterior, para realizar a correlação de alertas utilizando a técnica de CBR, um conjunto de pesos devem ser definidos aos atributos utilizados no

cálculo da similaridade. Nesse sentido, nove atributos obtidos a partir de alertas (e casos) são utilizados nesse cálculo específico. É importante salientar que, para obter um nível de correlação próximo do ideal, i.e. eliminando alertas do tipo falso positivo e apresentando alertas reais, os pesos no cálculo da similaridade devem ser atribuídos da melhor forma possível. Porém, para chegar nesses pesos, é necessário utilizar um processo de tentativa e erro orientado por resultados, onde certos atributos inferem mudanças maiores na precisão dos resultados obtido no cálculo de similaridade.

Durante os testes realizados nesse trabalho, identificou-se que três atributos (*Source IP Address*, *Target IP Address* e *Detection Time*) influenciam de forma mais evidente nos resultados do cálculo de similaridade, sendo que os outros atributos passam por um ajuste mais fino de peso, não impactando da mesma forma. Assim, os experimentos apresentados a seguir tem como objetivo evidenciar a metodologia utilizada para atribuir os pesos considerados ideais para a correlação de alertas, que foi descrita na Tabela 4.2 no Capítulo 4.

Todos os experimentos realizados a seguir utilizaram-se da configuração do sistema descrita na Seção 5.1. Os dados utilizados do sistema compreendem um total de 15 dias de coleta de dados, gerando uma base total de 23GB e 33.369.959 alertas obtidos dos seis sensores distribuídos que compõem a configuração do sistema. Foram utilizados 3 percentuais de na precisão da similaridade, variando acima de 70%, 80% e 90%, respectivamente. Os experimentos foram executados em uma máquina com processador Intel Core i7 3.9GHz, 32GB de memória RAM (*Random Access Memory*) e 128 GB SSD (*Solid State Drive*) executando o Sistema Operacional MAC OSX 10.9.1 e o banco de dados PostgreSQL 9.3.1. A base de casos foi alimentada com 5000 alertas classificados como *portscan*. Além disso, também foram selecionados aleatoriamente da base de conhecimento KBAM um total de 9500 alertas, sendo utilizados em todos experimentos e classificados pelo tipo de ataque como descrito na Tabela 5.3.

Tabela 5.3 – Alertas selecionados.

Classificação do alerta	**Quantidade**
COMMUNITY WEB-MISC mod_jrun overflow attempt	5046
COMMUNITY WEB-PHP thinkWMS index.php SQL injection attempt	159
NETBIOS SMB-DS Session unicode username overflow attempt	982
WEB-ATTACKS cc command attempt	1363
WEB-ATTACKS id command attempt	1936
WEB-ATTACKS mail command attempt	184

5.2.3.1 Experimento 1

Esse experimento tem como objetivo definir uma referência básica para os próximos. Assim, para esse experimento utilizou-se o peso de valor dois, de forma uniforme sobre todos os atributos que compõem o cálculo de similaridade. Esses valores são apresentados na Tabela 5.4.

Tabela 5.4 – Peso dos Atributos - Experimento 1.

Atributo	**Peso**
Analyser ID	2
Detection time	2
Source IP Address	2
Source Port	2
Target IP Address	2
Target Port	2
Service Protocol	2
Classification	2
Classification Type	2

Conforme pode ser observado na Tabela 5.5, o conjunto de alertas que apresentaram um grau de similaridade superior a de 90% em relação a base de casos foi bastante alto para alguns tipos de ataque. Mesmo assim, percebe-se que o algoritmo foi eficiente o suficiente para reduzir pela metade os alertas falsos-positivos em em alguns tipos de ataque, considerando a utilização dos resultados com precisão acima de 90%. A consulta realizada para calcular a similaridade de todos os alertas e toda base de casos levou 39.1 minutos para ser concluída, sendo uma média de 0.24 segundos por alerta.

Tabela 5.5 – Resultados para o Experimento 1.

Classificação do Ataque	**Precisão da Similaridade**		
	$\geq$ 70% e < 80%	$\geq$ 80% e < 90%	$\geq$ 90%
COMMUNITY WEB-MISC mod_jrun overflow attempt	6	2554	2313
COMMUNITY WEB-PHP thinkWMS index.php SQL injection	0	33	126
NETBIOS SMB-DS Session unicode username overflow attempt	74	908	0
WEB-ATTACKS cc command attempt	0	563	797
WEB-ATTACKS id command attempt	4	1410	518
WEB-ATTACKS mail command attempt	0	157	27

5.2.3.2 Experimento 2

Levando em consideração o experimento anterior, um grande conjunto de falsos positivos continua a ser reportado. Isso ocorre porque Endereços de IP, tanto de origem como destino, assim como o tempo de detecção não são evidenciados no cálculo de similaridade do experimento anterior, uma vez que possuem o mesmo valor que os outros campos. Logo, com o objetivo de deixar o cálculo mais preciso, definiu-se valores descritos na Tabela 5.6 para os pesos. O atributo de *Source IP Address* apresenta um peso 6, sendo que o atributo *Target IP Address* recebe um peso 4 e o atributo *Detection Time* peso 3.

Tabela 5.6 – Peso dos Atributos - Experimento 2.

Atributo	**Peso**
Analyser ID	2
Detection time	3
Source IP Address	6
Source Port	2
Target IP Address	4
Target Port	2
Service Protocol	2
Classification	2
Classification Type	2

Conforme descrito na Tabela 5.7, o conjunto de alertas com similaridade maior que 90% reduziram-se consideravelmente, confirmando que evidenciando-se o peso em determinados atributos é relevante na correlação de alertas e na redução de falsos-positivos. Além disso, percebe-se que ataques classificados como *WEB-ATTACKS id command attempt* apresentaram um maior número de alertas do que experimento anterior, evidenciando assim a necessidade de um estudo aprofundado da relação entre o tipo de ataque e o peso utilizado.

Tabela 5.7 – Resultados para o Experimento 2.

Classificação do Ataque	**Precisão da Similaridade**		
	$\geq$ 70% e < 80%	$\geq$ 80% e < 90%	$\geq$ 90%
COMMUNITY WEB-MISC mod_jrun overflow attempt	154	3543	1165
COMMUNITY WEB-PHP thinkWMS index.php SQL injection	7	143	6
NETBIOS SMB-DS Session unicode username overflow attempt	95	865	0
WEB-ATTACKS cc command attempt	13	627	711
WEB-ATTACKS id command attempt	129	387	868
WEB-ATTACKS mail command attempt	1	174	9

5.2.3.3 Experimento 3

Ao utilizar uma base de casos que consideram alertas do tipo *portscan*, observou-se que os principais pesos necessários para uma eficiente correlação foram contemplados na análise descrita no experimento anterior. Ou seja, os outros campos utilizados na correlação passam mais por um ajuste fino de teste e erro, que não se pronunciam da mesma forma que os campos de *Source IP Address*, *Target IP Address* e *Detection Time*. Durante os testes realizados neste trabalho, os valores descritos na Tabela 5.8 apresentaram as melhores taxas de correlação na redução de falsos-positivos.

Tabela 5.8 – Peso dos Atributos - Experimento 3.

Atributo	**Peso**
Analyser ID	2
Detection time	3
Source IP Address	6
Source Port	4
Target IP Address	4
Target Port	4
Service Protocol	2
Classification	3
Classification Type	3

Conforme apresentado na Tabela 5.9, o conjunto de alertas que apresentaram um grau de precisão de similaridade maior ou igual à 90% é consideravelmente menor que em relação ao experimento 1 ou ao total de 9500 alertas inicias. A consulta realizada para calcular esse experimento levou 38.9 minutos para ser realizada, sendo uma média de 0.24 segundos por alerta.

Tabela 5.9 – Resultados para o Experimento 3.

Classificação do Ataque	**Precisão da Similaridade**		
	$\geq$ 70% e < 80%	$\geq$ 80% e < 90%	$\geq$ 90%
COMMUNITY WEB-MISC mod_jrun overflow attempt	2046	1412	734
COMMUNITY WEB-PHP thinkWMS index.php SQL injection	30	120	4
NETBIOS SMB-DS Session unicode username overflow attempt	953	0	0
WEB-ATTACKS cc command attempt	432	422	497
WEB-ATTACKS id command attempt	450	589	387
WEB-ATTACKS mail command attempt	157	20	7

5.3 Resumo

Este capítulo apresentou os experimentos realizados para a validação da proposta de correlação de alertas. Para isso, utilizou-se experimentos realizados dentro da UFSM executando Sistemas de Detecção de Instrusão distribuídos, tendo seus alertas centralizados na base de conhecimento KBAM. Para a correlação de alertas utilizou-se a arquitetura de correlações de alertas em um IEWS apresentado anteriormente e através técnica de *Case-Based Reasoning* em conjunto com o georreferenciamento de IP evidenciou-se a possibilidade da redução de alertas definidos como falso-positivos.

A utilização da arquitetura de correlação de alertas nos experimentos apresentados reduziu a quantidade de alertas falsos-positivos em 82%, totalizando uma redução de 7871 dos 9500 alertas falsos-positivos iniciais. Percebeu-se também que evidenciado-se o peso em determinados atributos impactaram diretamente nos resultados obtidos, uma vez que a diferença do total de alertas com similaridade superior a 90% entre o primeiro e terceiro experimento foi de 2152 alertas. Além disso, o tempo de consulta para comparar um único alerta com toda a base de casos foi de 0.24 segundos, evidenciando assim a possibilidade de executar em tempo real a proposta desta dissertação.

6 CONCLUSÕES

Medidas tradicionais de detecção de intrusão não são o suficientes para garantir a segurança das infraestruturas de rede, uma vez que a complexidade dos ataques tendem a aumentar. Nesse sentido, a necessidade de cooperação entre Sistemas de Detecção de Intrusão, impulsionado pela crescente oferta de serviços críticos na rede tem exigido uma abordagem mais complexa como um *Internet Early Warning System* (IEWS).

Como base de conhecimento do IEWS utilizou-se a KBAM, um modelo desenvolvido pelo GTSEG-UFSM. A base de conhecimento KBAM representa os dados de eventos de detecção de intrusão explorando o formato *Intrusion Detection Message Exchange Format* (IDMEF). Para comportar uma arquitetura com objetivo de relacionar e detectar intrusões, esta base de conhecimento foi estendida para compor um arquitetura de correlação de alertas.

Foi apresentada uma arquitetura extensível para *Internet Early Warning System* baseada em noções de Consciência Situacional para a correlação de alertas, assim como os componentes necessários. Composta das etapas de pré-processamento e correlação, a primeira tem como objetivo normalizar os alertas recebidos em um formato padrão e completo. Já a segunda descreve o cálculo de similaridade em cada atributo, utilizando recursos como a técnica *Case-Based Reasoning* e Georreferenciamento de IP. Com isso, obteve-se uma visão da Consciência Situacional da rede monitorada através do resultado da correlação de alertas entre sensores distribuídos.

A realização de experimentos na infraestrutura de rede da Universidade Federal de Santa Maria permitiu demonstrar a aplicabilidade da arquitetura de correlação de alertas, diminuindo falso-positivos em 82%. O principal diferencial deste trabalho é a implementação, em um ambiente real (na UFSM), de um sistema de correlação de alertas como base para um IEWS, que utiliza a noção de georreferenciamento.

6.1 Trabalhos Futuros

Contemplar os requisitos de Consciência Situacional no contexto de um *Internet Early Warning System* exige uma série de esforços adicionais nos dois primeiros níveis explorados neste trabalho. No primeiro, percepção, sugere-se explorar a utilização de sensores em *hardware* embarcados, tendo sua implementação em outros segmentos de rede. No segundo nível, compreensão, como a arquitetura de correlação de alertas apresentada é extensível, espera-se que com a utilização de outras técnicas de detecção de intrusão e que, utilizadas em conjunto

com listas públicas de agentes e ataques maliciosos, disponíveis na internet, seja possível alimentar base de casos com outros tipos de ataques além dos classificados como *portscan*.

Como observado no decorrer do trabalho, a definição dos pesos para cada atributo foi um dos principais obstáculos encontrados neste trabalho, nesse sentido sugere-se a utilização de algoritmos capazes de selecionar os principais atributos e quantificar o valor de seu peso. Além disso, espera-se que a utilização do recurso de Georeferenciamento de Endereços IP continue a ser explorado, como por exemplo na visualização de incidentes de segurança.

REFERÊNCIAS

AAMODT, A.; PLAZA, E. Case-based reasoning: foundational issues, methodological variations, and system approaches. **AI communications**, [S.l.], v.7, p.39–59, 1994.

APEL, M. et al. Early Warning System on a National Level. **1st European Workshop on Internet Early Warning and Network Intelligence (EWNI)**, [S.l.], 2010.

BACE, R.; MELL, P. **NIST Special Publication on Intrusion Detection Systems**. [S.l.: s.n.], 2001.

BASTKE, S.; DEML, M.; SCHMIDT, S. **Internet Early Warning Systems - Overview and Architecture**. [S.l.: s.n.], 2010.

BRANDÃO, J. E. M. d. S. **Composições de IDSs**: viabilizando o monitoramento de segurança em ambientes de larga escala. 2007. 147p. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Santa Catarina.

BSUFKA, K.; KROLL-PETERS, O.; ALBAYRAK, S. Intelligent network-based early warning systems. **Critical Information Infrastructures**, [S.l.], 2006.

D'AMICO, A.; KOCKA, M. Information assurance visualizations for specific stages of situational awareness and intended uses: lessons learned. **Visualization for Computer Security, 2005 . . .**, [S.l.], p.107–112, 2005.

DEBAR, H.; CURRY, D.; FEINSTEIN, B. **The intrusion detection message exchange format (IDMEF)**. [S.l.: s.n.], 2007.

ENDSLEY, M. R. Toward a Theory of Situation Awareness in Dynamic Systems. **Human Factors: The Journal of the Human Factors and Ergonomics Society**, [S.l.], v.37, n.1, p.32–64, March 1995.

ESMAILI, M. Case-based reasoning for intrusion detection. **Computer Security Applications Conference**, [S.l.], p.214–223, 1996.

FAN, G.; JIHUA, Y.; MIN, Y. Design and implementation of a distributed IDS alert aggregation model. **4th International Conference on Computer Science & Education, 2009. ICCSE '09.**, [S.l.], p.975–980, 2009.

GARDNER, R.; HARLE, D. Methods and systems for alarm correlation. **Global Telecommunications Conference**, [S.l.], p.136–140, 1996.

GOLLING, M.; STELTE, B. Requirements for a future EWS-Cyber Defence in the internet of the future. **2011 3rd International Conference on Cyber Conflict (ICCC)**, [S.l.], p.1–16, 2011.

HAINES, J.; RYDER, D. K. Validation of sensor alert correlators. **Security and Privacy**, [S.l.], 2003.

HALL, D.; MCMULLEN, S. **Mathematical techniques in multisensor data fusion**. [S.l.: s.n.], 2004.

HOLLANDS, J.; WICKENS, C. **Engineering psychology and human performance**. [S.l.: s.n.], 1999.

HOLUB, V. et al. Run-time correlation engine for system monitoring and testing. **Proceedings of the 6th international conference on Autonomic computing - ICAC '09**, New York, USA, p.43, 2009.

INNELLA, P. **The Evolution of Intrusion Detection Systems**. 2010.

JAKOBSON, G.; WEISSMAN, M. D. Alam Correlation. **IEEE Network**, [S.l.], 1993.

KABIRI, P.; GHORBANI, A. A. **A Rule-Based Temporal Alert Correlation System**. 2005.

LI, W.; TIAN, S. Preprocessor of intrusion alerts correlation based on ontology. **International Conference on Communications and Mobile Computing**, [S.l.], v.3, p.460–464, January 2009.

MCGUINNESS, B.; FOY, L. A subjective measure of SA. **First human performance, situation awareness, and automation conference, Savannah, Georgia**, [S.l.], 2000.

NTP.BR. **NTP.br - Hora Certa via Internet**. Acesso em 10/08/2013, Disponível em http://ntp.br.

ONWUBIKO, C. Functional requirements of situational awareness in computer network security. **Intelligence and Security Informatics, 2009. ISI**, [S.l.], p.209–213, 2009.

PEREIRA, H. **Sistema de Detecção de Intrusao para Serviços Web baseado em Anomalias**. 2011. 103p. Tese (Doutorado em Ciência da Computação) — Pontifícia Universidade Católica do Paraná.

PETRI, G. **Modelo de Dados de uma Base de Conhecimento para Internet Early Warning Systems**. 2013. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Santa Maria.

PETRI, G. et al. Modelagem de uma Base de Conhecimento para o Monitoramento de Ataques. **Escola Regional de Redes de Computadores**, [S.l.], 2012.

PETRI, G. et al. Building Situation Awareness to Monitor Critical Infrastructures. **Latin-American Symposium on Dependable Computing (LADC)**, [S.l.], 2013.

PORRAS, P.; NEUMANN, P. EMERALD: event monitoring enabling response to anomalous live disturbances. **Proceedings of the 20th national information**, [S.l.], 1997.

PRELUDE. **Prelude SIEM**. Acesso em 10/08/2013, Disponível em http://www.prelude-ids.com/index.php/uk/.

RAMADAS, M.; OSTERMANN, S.; TJADEN, B. Detecting anomalous network traffic with self-organizing maps. **Recent Advances in Intrusion Detection**, [S.l.], p.36–54, 2003.

SALAH, S.; MACIA-FERNANDEZ, G.; DIAZ-VERDEJO, J. A model-based survey of alert correlation techniques. **Computer Networks**, [S.l.], v.57, n.5, p.1289–1317, April 2013.

SARTER, N. B.; WOODS, D. D. Team Play with a Powerful and Independent Agent: a full-mission simulation study. **Human Factors: The Journal of the Human Factors and Ergonomics Society**, [S.l.], v.42, n.3, p.390–402, September 2000.

SCARFONE, K.; MELL, P. Guide to Intrusion Detection and Prevention Systems (IDPS). **NIST Special Publication**, [S.l.], 2007.

SILVA, L. d. S. **Uma metodologia para detecção de ataques no tráfego de redes baseada em redes neurais**. 2007. Tese (Doutorado em Ciência da Computação) — Instituto Nacional de Pesquisas Espaciais.

SNORT. **Snort Network Intrusion Detection System web site**. Acesso em 10/08/2013, Disponível em http://www.snort.org/.

STANIFORD-CHEN, S. et al. GrIDS - A Graph Based Intrusion Detection System For Large Networks. **National Information Systems Security Conference**, [S.l.], p.361–370, 1996.

TADDA, G. Cyber Situational Awareness. **Advances in Information Security**, Boston, MA, 2010.

VALDES, A.; SKINNER, K. Probabilistic alert correlation. **Recent Advances in Intrusion Detection**, [S.l.], p.54–68, 2001.

VALEUR, F.; VIGNA, G. **Intrusion detection and correlation**: challenges and solutions. [S.l.: s.n.], 2005.

WANGENHEIM, A. von; WANGENHEIM, C. G. von. **Raciocínio Baseado em Casos**. [S.l.: s.n.], 2003. 300p.

YU, J. et al. TRINETR: an intrusion detection alert management systems. **Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004. 13th IEEE International Workshops on**, [S.l.], p.235–240, 2004.

ZHIHONG, T. et al. Alertclu: a realtime alert aggregation and correlation system. **2008 International Conference on Cyberworlds**, [S.l.], p.778–781, September 2008.

ZHUANG, X. et al. Applying Data Fusion in Collaborative Alerts Correlation. **2008 International Symposium on Computer Science and Computational Technology**, [S.l.], p.124–127, 2008.