Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática
2009.2

APLICANDO A FREQÜÊNCIA DE EPISÓDIOS NA CORRELAÇÃO E PREDIÇÃO DE ANOMALIAS

Trabalho de Graduação

LEONARDO HENRIQUE VILAÇA SILVA

Orientador: Prof. Djamel Fawzi Hadj Sadok (jamel@cin.ufpe.br)
Co-orientador: Prof. Eduardo Feitosa (elf@cin.ufpe.br)

Recife, dezembro de 2009

Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática
2009.2

APLICANDO A FREQÜÊNCIA DE EPISÓDIOS NA PREDIÇÃO DE ANOMALIAS

Trabalho de Graduação

LEONARDO HENRIQUE VILAÇA SILVA

Projeto de Graduação apresentado no Centro de Informática da Universidade Federal de Pernambuco por Leonardo Henrique Vilaça Silva, orientado pelo Prof. PhD. Djamel Fawzi Hadj Sadok, como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Djamel Fawzi Hadj Sadok (jamel@cin.ufpe.br)
Co-orientador: Prof. Eduardo Feitosa (elf@cin.ufpe.br)

Recife, dezembro de 2009

FOLHA DE APROVAÇÃO

APLICANDO A FREQÜÊNCIA DE EPISÓDIOS NA PREDIÇÃO DE ANOMALIAS

LEONARDO HENRIQUE VILAÇA SILVA

APROVADO EM 04 DE DEZEMBRO DE 2009

BANCA EXAMINADORA:

_____

Prof. Djamel Fawzi Hadj Sadok, PhD –
UFPE (Orientador)

_____

Prof. Ruy José Guerra Baretto de Queiroz, PhD –
UFPE (Avaliador)

"O dinheiro não traz felicidade, para quem não sabe o que fazer com ele."

Machado de Assis

"Não existe um caminho para a felicidade. A felicidade é o caminho."

Mahatma Gandhi

"O homem não voará nos próximos 1 000 anos."

Wilbur Wright

Agradecimentos

Estes agradecimentos são direcionados a todos que fizeram parte desses logos anos de graduação. Gostaria de deixar claro que os nomes citados não estão em ordem de preferência ou importância. Todos tiveram uma parcela de contribuição para que eu pudesse chegar a este final, esta é a oportunidade de agradecê-los.

Agradeço primeiramente aos meus pais, que sempre incentivaram minha educação escolar, não medindo esforços para me dar suporte nos momentos que precisei. As minhas irmãs Polliana e Julianna que aprendi a conviver e que sinto falta, quando por motivos geográficos, fico sem ver-las. Gigi, você veio para trazer alegria para nós. Agradeço a todos os familiares, tios, tias, primos, avôs, avós e cunhados.

Àqueles amigos da UFPE: 110, Alberto, Adelmo, Bei (manja muito), Cabeleira, Carol Cidão, Ciro (camarão), Du Brega (atual vestibulando), Elton Jonh, Urubú, Fumega, Calê, Flavinho, Maceió, Djorje, Perereca Albina, Cebola, Lucin, Márcio, Natália, Biu, Silvão, Styve Stallone, Tan2, Zé Fumaça, Inó, Vituxinho, Segurança, Guila, Vanessinho, Vanessão, Potter, Físico e Galego do Caldinho.

A todos meus amigos do tempo do colégio, que até hoje lembro dos momentos felizes que não voltam mais, de muita descontração. Agradeço em especial a: Carol, Berg, Bel, Belôto, Daniel, Flávia, Fred, Kong, Help, Rafaela, Kara Véa, Lucas, Pedrão, Peixe, Marcela, Michele, Rosana, Raône, Silvio Santos, Thiago e Xambinho.

Aos amigos do GPRT Nadia, Manu, Rafinha, Mestre, Arthur, Rover, Hashid, Rodrigo Germano, Josias, Digão, Fernandinho, Chico, Bate-Bate, Andréa, Paty, Pigmel, Cheiroso e Aly. Dentre os membros do GPRT não posso deixar de agradecer de uma forma especial a Professora Judith Kelner por todas as oportunidades, ensinamentos e puxões de orelha, o Professor Djamel Sadok, que além de orientador se mostrou um facilitador nestes anos de GPRT. Agradeço a Eduardo Feitosa (feitosa), meu co-orientador e amigo que tanto fez para a realização e conclusão deste trabalho, que me ajudou muito tirando dúvidas, me mostrando o caminho e dando conselhos por todas as etapas deste trabalho.

Agradeço a Mary, minha namorada, por me fazer tão feliz e pela a compreensão dos dias ausentes para fazer este trabalho. Você é parte fundamental neste processo.

Por fim, peço desculpas aos importantes amigos não citados (amigos de Lajedo, da Praia, Badiel, NSL, balada e Loro), vocês são sem sobra de dúvidas, muito importantes.

Resumo

Cada vez mais complexas, as técnicas de ataque a protocolos e proliferação de vírus e worms, entre outros, vêem causando diversas anomalias (desvios do funcionamento normal da rede) em redes e sistemas vulneráveis. Essas anomalias desperdiçam recursos, ameaçam a confidencialidade, privacidade e autenticidade dos dados. Diante da grande variedade de anomalias existentes no tráfego de rede e dos prejuízos causados por elas, percebe-se a importância de um sistema ou arquitetura eficiente na detecção de anomalias de tráfego. Este trabalho tem como objetivo a implementação de um sistema de detecção de anomalias, utilizando para isso algoritmos de freqüências de episódios, fazendo uma captura passiva do tráfego da rede. Ele será desenvolvido visando melhorar a segurança através da auditoria dos dados da rede, a partir de dispositivos de infra-estrutura existentes. Como o sistema irá utilizar freqüência de episódios em seqüência de eventos, será possível detectar anomalias conhecidas, como também anomalias desconhecidas e também eventos raros na rede automaticamente.

Sumário

Índice de Figuras

1 Introdução

A popularização dos computadores pessoais, a evolução das tecnologias utilizadas para a manipulação, armazenamento e apresentação das informações, assim como o fácil acesso à Internet, tornou possível nos últimos anos a formação de uma enorme rede de computadores interconectados. Pessoas e organizações de diversas partes do mundo agora trocam informações através de sistemas computacionais em um ritmo jamais visto na história.

Com essa evolução novas preocupações surgiram, entre elas, a com a segurança das redes de computadores. As informações devem ser armazenadas de maneira estratégica e confidencial. Senhas, cadastros, dados pessoais e tudo o que pode ser alvo de ataques, precisa estar de maneira apropriada e segura para garantir a integridade de corporações e das pessoas.

Técnicas de ataque a protocolos, vírus, worms, entre outros, com o passar do tempo vão se tornando cada vez mais complexas, causando anomalias (desvio do funcionamento normal da rede) em redes vulneráveis. Essas anomalias desperdiçam recursos, ameaçam a confidencialidade, privacidade e autenticidade dos dados.

Existem vários tipos de anomalias, uma das mais comuns é a que explora as falhas dos elementos de rede, protocolos, serviços e computadores de usuários finais. Uma outra conhecida, é quando usuários não autorizados tentam acessar recursos restritos. Outra anomalia que pode ocorrer é uma rede com uma invasão bem sucedida enviando dados para um atacante ou até mesmo atacantes manipulando protocolos de rede para ocultar suas ações realizadas.

Diante da grande variedade de anomalias existentes no tráfego de rede e dos prejuízos causados por elas, percebe-se a importância de um sistema ou arquitetura eficiente na detecção de anomalias de tráfego.

1.1 Motivação

O tráfego da internet, além deste enorme quantidade de anomalias, também possui uma inerente mudança de volume do seu tráfego, como mostram os conceitos de auto-similaridade, dependência de longa distância e fractalidade múltipla [16].

Existem vários pesquisas atuais, que buscam tratar e analisar o tráfego anômalo. Muitos destes trabalhos realizam cálculos estatísticos e medições na quantidade de dados que trafegam na rede. Os parâmetros analisados abrangem um parâmetro, ou correlacionam alguns parâmetros. Entre eles: endereços IP e porta, quantidade de pacotes trafegando na rede, tamanho dos pacotes, entre outros.

A análise do tráfego pode ser temporal ou espacial. A análise temporal correlaciona os parâmetros atuais da rede, com outros dados que ocorreram em outra faixa de tempo. Já a análise espacial, faz uma comparação entre dados que estão dispostos em pontos diversos na rede.

As áreas de predição e detecção de anomalias objetivam a criação, desenvolvimento e testes de técnicas que respectivamente, evitem a propagação das anomalias, mas se mesmo assim elas ocorrerem, que as mesmas, sejam identificadas e excluídas.

Muitas das redes são equipadas com softwares que lançam alertas quando determinados eventos anômalos são detectados. Porém, dependendo da quantidade de tráfego analisado, a quantidade de alertas gerados pode ser muito alta, não tendo condições de ser manipulável manualmente por administradores de redes.

Este trabalho propõe o desenvolvimento de um sistema que detecte as anomalias, gerando alerta. Posteriormente, é realizada uma analise esses alertas, tornando possível a obtenção de predições de futuros tráfegos anômalos em tempo real, a partir de uma técnica que correlacione os alertas. Esta técnica é chamada de freqüência de episódios, que é mostrada com detalhes no capítulo 3.

1.2 Objetivos

Este trabalho tem por objetivo de desenvolver um sistema de correlação e predição de anomalias em redes de computadores. Para tanto, a técnica de freqüência de episódios é utilizada para a construção de um protótipo, capaz de receber um conjunto de alertas, processá-los e gerar regras capaz de confirmar a identificação de ataques e anomalias sobre o tráfego da rede, além de permitir a predição de certos eventos.

1.3 Organização do Trabalho

O restante deste trabalho está dividido da seguinte forma.

O capítulo 2 descreve as técnicas de correlação e predição de alertas, apresentando suas definições, uma classificação e exemplos de trabalhos relacionados nesta área. O capítulo

3 descreve a técnica de episódios freqüentes. Uma explanação teórica detalhada é apresentada bem como uma breve descrição de alguns trabalhos relacionados. O capítulo 4 descreve a solução proposta, incluindo a arquitetura e desenvolvimento de um protótipo funcional. As avaliações e resultados são apresentados no capítulo 5. Por fim, o capítulo 6 apresenta as conclusões obtidas, as dificuldades encontradas e os trabalhos futuros.

2 Correlação e Predição de Alertas

2.1 Correlação de Alertas

A correlação de alertas é uma importante técnica para o gerenciamento de grandes volumes de alertas de intrusão e anomalias que são gerados por IDS, ADS e soluções colaborativas. A correlação de alertas é definida como um processo que contém diversos componentes com o propósito de analisar alertas e fornecer alto nível insight sobre o estado de segurança da rede sobre observação.

Um dos importantes usos da correlação de alertas é reconhecer estratégias ou planos de diferentes intrusões e inferir o objetivo dos ataques. A idéia é tentar identificar o próximo passo de uma intrusão ou seu objetivo através da comparação de padrões e, desta forma, prevenir-se da anomalia, minimizando seus efeitos. A correlação de alertas fornece meios para agrupar diferentes alertas interligados logicamente dentro de cenários de ataque, permitindo a análise das estratégias de ataque.

Inúmeras técnicas de correlação de alertas foram propostas ao longo dos anos. Geralmente, elas podem ser classificadas em quatro (4) categorias: correlação baseada em cenários, correlação baseada em regras, correlação baseada em estatística e por último, correlação baseada em tempo.

2.1.1 Correlação baseada em Similaridade

A correlação baseada em similaridade visa encontrar semelhanças entre os atributos dos alertas. A idéia é comparar um alerta com todos os outros que tenham atributos similares como, por exemplo, endereço IP de origem, endereço IP de destino, portas, *timestamp*, classe de ataque, e assim por diante. Alertas semelhantes tendem a ter causas semelhantes ou efeitos semelhantes sobre os recursos da rede. Desta forma, técnicas de similaridade pura e agregação são empregadas.

O trabalho desenvolvido por Valdés e Skiner [25] utiliza uma abordagem probabilística de correlação de alerta para o projeto EMERALD [32]. Foram implementadas três fases de correlação: *tópicos de ataque sintético*, onde os alertas são agrupados se alguma similaridade é encontrada; *incidentes de segurança*, utilizada para fundir o mesmo ataque relatado por múltiplos detectores; *relatórios de ataques correlacionados*, onde se fundem

alertas representando diferentes etapas de um ataque complexo. Em [14], Debar e Wespi propuseram um algoritmo de agregação e correlação de alertas de intrusão. Nesta abordagem, três passos são necessários para realizar a agregação de alertas e de correspondência: *processamento de alertas*, onde os alertas são traduzidos para um modelo de dados (os autores utilizaram a primeira discussão sobre IDMEF como modelo de dados); *correlação de relacionamento*, que extrai a correlação entre os alertas; *agregação de relacionamento*, onde a saída da segunda etapa (alertas) são agregadas em sete diferentes cenários (situações) de acordo com seus atributos.

Em relação às abordagens de agregação (*clustering*), os trabalhos de Julish [26] e Cuppens [27] são citados como exemplos clássicos na literatura. Já o trabalho de Zhu e Ghorbani [34] utiliza duas abordagens de redes neurais (*Multilayer Perceptron* e *Support Vector Machine* - SVM) para determinar a correlação entre os alertas e, conseqüentemente, estabelecer relações de causalidade. Para isso, introduziram a idéia de matriz de correlação de alerta (*Alert Correlation Matrix* - ACM) para armazenar a correlação média entre os alertas, que é calculada adaptativamente com base na análise estatística de entrada de alertas consecutivos. A característica de adaptação deste método torna possível começar com valores iniciais de probabilidade e ir aprender com o ambiente à medida que a operação continua, permitindo a extração de estratégias de ataque em alto nível.

2.1.2 Correlação baseada em Cenários de Ataque

Técnicas de cenário de ataque baseiam-se no fato de que ataques freqüentemente necessitam de várias etapas ou ações para alcançar seu objetivo. A idéia é que cada cenário de ataque tenha ou represente as correspondentes etapas necessárias para que o ataque seja bem sucedido. Desta forma, os alertas são comparados com os cenários de ataque conhecidos para serem correlacionados.

Tipicamente, os trabalhos nesta área têm sido focados em dois métodos: o uso de modelos formais definidos por especialistas humanos para especificar os cenários de ataque ou empregado aprendizado de máquina para criar tais cenários. Boris e Debar [35] propuseram um componente de correlação multi-alerta, baseado no formalismo de crônicas, para modelar cenários de ataque. O formalismo de crônicas, proposto por Dousson [36], é usado para construir blocos de correspondência e representar conjunto de padrões (cenários de ataque). Quando novos alertas são recebidos, eles são comparados com as crônicas. As crônicas são atualizadas sempre que uma correspondência ocorre ou quando ainda não foi construída.

Além disso, as pesquisas têm proposto vários tipos de correlação formal e diversas linguagens de definição para gerar cenários de ataque. Entre os mais conhecidos estão LAMBDA [37], STATL [38], ASL [39], JIGSAW [40] e ADeLe [41]. Em [42], os autores propuseram um esquema para fundir alertas em cenários de ataque pré-definidos. A idéia é usar um sistema de fusão para determinar para qual cenário de ataque um alerta pertence. Assim, sempre que um novo alerta é recebido, é comparada para determinar para qual cenário de ataque ele deve ser um membro. Os cenários são gerados usando duas abordagens, uma heurística e outra com mineração de dados.

2.1.3 Correlação baseada em Regras

Uma vez que ataques e suas variantes normalmente geram um grande número de cenários, a utilização de regras (pré-condição e pós-condições) tem sido empregada para resolver este problema, reduzindo o número de possíveis cenários ataques. Esta abordagem é conhecida como correlação baseada em regras, embora alguns autores a classifiquem como subclasse das técnicas de cenário de ataque.

O trabalho de Debar e Wespi [33], anteriormente descrito e classificado em correlação de similaridade, usa regras de conseqüência para definir cenários de ataque. Regras de conseqüência especificam que um evento (alerta) deve ser seguido por outro tipo de evento, permitindo assim que os alertas sejam correlacionados. Em [43], os autores propuseram uma abordagem para mapear as relações causais entre os alertas por meio de regras. Eles introduziram o conceito de hiper-alerta (*hyper alert*) para codificar a pré-condição e pós-condição de um alerta. Desta forma é possível extrair pré-requisitos e conseqüências dos hiper-alertas e gerar gráficos para determinar o objetivo do atacante.

2.1.4 Correlação baseada em Estatísticas

Apesar de efetivas, as técnicas de similaridade e cenário de ataque são apenas voltadas para correlacionar ataques e anomalias bem conhecidos. Visando preencher esta lacuna, técnicas estatísticas têm sido propostas para detectar ataques e anomalias desconhecidas.

Qui e Le [44] usaram GCT (*Granger Causality Test*), um método de análise de séries temporais, para correlacionar alertas com ênfase na análise de cenários de ataque. A idéia por trás dessa abordagem é usar a análise de causalidade para correlacionar alertas e gera cenários de ataque sem qualquer conhecimento pré-definido. Para isso, o método assume que cada passo do ataque irá gerar alertas que têm semelhanças estatísticas em seus atributos, e esses passos de ataque tem relação de causalidade [31].

Em outro trabalho, Qui [45] emprega uma rede Bayesiana para modelar a relação de causalidade entre os alertas, onde os alertas são nodos e as suas relações de causalidade são arestas. Neste modelo, alertas contínuos são divididos ao longo de intervalos de tempo iguais e o estado de cada nodo correspondente a um alerta é um valor binário que representa a presença do alerta no intervalo de tempo [22]. A idéia central deste trabalho é descobrir quais os tipos de alerta pode causar um alerta do tipo X e como a probabilidade condicional de X está relacionada com as suas causas (pais). Almgren et al. [46] também utiliza uma abordagem semelhante.

2.2 Predição de Alertas

A predição de alertas é uma técnica que tenta prever comportamentos divergentes de uma rede, comparando as características observadas no momento com padrões previamente observados e definidos. A idéia básica da predição é a após sofrer um determinado tipo de ataque ou anomalia, a seqüência de eventos que a geraram é armazenada. Se esse padrão se repetir ao logo do tempo, medidas podem ser tomadas para evitá-lo. Desta forma, são reduzidos significantemente os danos causados por ataques e anomalias a rede, pois procedimentos de defesa ou sinalizações ao administrador de rede são gerados antes do ataque ocorrer, evitando que a rede seja afetada.

Tipicamente, as abordagens para predição de alertas executam duas fases: o treinamento para identificação padrões de normalidade e a comparação para verificar se o padrão atual diverge dos padrões obtidos na fase de treino, seguido da geração do alerta. Os sistemas que implementam predição são chamados de IPS (*Intrusion Prevention System*), pois além da detecção (trabalho de IDS) podem prever ataques baseando-se no estado atual da rede.

Ao longo dos anos diversas abordagens têm sido utilizadas para realizar a predição de anomalias. Basicamente, elas são categorizadas de duas formas: abordagens de predição e abordagens de tomada de ação. Ye e Li [22] propuseram uma série de estudos probabilísticos para detecção de intrusos. Foram analisadas técnicas envolvendo árvores de decisão para relacionar atividades do usuário com predição de anomalia, além de cadeias de Markov, Teste de Hotelling T2 e teste multivariado chi-quadrado. Para cada uma destas técnicas foram gerados e analisados seus resultados com relação às anomalias.

Em [23], Ye e Chen utilizam cadeias de Markov para predizer ataques baseando-se em atividades anômalas dos usuários. A métrica de distância chi-quadrado foi utilizada para

medir o desvio das atividades observadas das atividades normais do usuário. Dois métodos de predição foram testados fazendo a auditoria de dados de computadores com atividades normais e atividades de intrusão.

No trabalho de Pikoulas e Buchanan [30] uma técnica de predição bayesiana para prever ações de usuários é apresentada. O sistema proposto neste trabalho consegue distinguir uma mudança corriqueira no comportamento do usuário, de uma mudança de padrão caracterizada como uma anomalia. Em outro trabalho, Hu [31] desenvolveu um sistema que faz a predição de ataques em dois estágios: classificação e predição. A classificação é baseada em SVM (*Support Vector Machine*) e a predição é baseada em SOM (*Self-Organizing Map*), um tipo específico de rede neural com aprendizagem não supervisionada.

Wang e Liu [26] propuseram uma técnica para correlação e predição de alertas, onde alertas individuais são correlacionados para detectar ataques gerados em várias etapas. O trabalho de Kannadiga e Zulkernine [29] resultou em um IPS chamado E-NIPS (*Event-based Network Intrusion Prediction System*). Utilizando uma abordagem *victim-end*, a idéia é particionar o cenário de ataque em diversas fases, dependendo das ações do atacante durante sua atuação. Ataques que possuem o mesmo objetivo são agrupados em classes para diminuir o processamento do módulo de predição. Quando as primeiras fases de um ataque são detectadas, isto é, quando as fases iniciais de um ataque correspondem a uma classe de ataque, alertas para administradores de redes são lançados. As seqüências de eventos dos ataques são representadas por regras, que são utilizadas para correlacionar classes de ataques detectadas nos cenários dos ataques.

O trabalho de Ramasubramanian [24] desenvolveu um framework para um sistema de predição de anomalias utilizando o modelo de predição de redes neurais Quickprop. O foco do trabalho é a detecção de modificações significantes na intensidade das transações dos usuários em uma instituição financeira.

3 Freqüência de Episódios

Freqüência de Episódios é uma técnica de descoberta de padrões temporais em dados seqüenciais. Mannila, Toivonen e Verkamo [17] propuseram um método desenvolvido para o estudo de padrões espaciais e temporais, baseado em eventos, chamado de descoberta de episódios freqüentes (do inglês *Frequent Episodes Discovery*). Este método baseia-se em dois conceitos chave: seqüência de eventos e episódio. O primeiro refere-se ao comportamento ou ações de usuários ou sistemas que podem ser coletados em diversos domínios. Episódio é uma coleção de eventos que ocorrem relativamente perto uns dos outros em uma ordem parcial.

O objetivo deste método é analisar em uma seqüência de eventos, quais episódios são freqüentes (o que justifica o nome freqüência de episódios). Essa técnica de mineração temporal pode ser utilizada em diversas aplicações. Primeiramente, ele foi utilizado para analisar alarmes oriundos de redes de telecomunicações [17], que eram dispostos temporalmente. Um gama de outros cenários, que representam uma seqüência de eventos, pode ser estudada utilizando esta abordagem: situação do clima de uma região ao longo dos anos [19], histórico de doenças de um determinado indivíduo [20], correlação entre falhas que ocorrem em uma linha de montagem de motores [21], entre outros. Em [18], é aplicada a técnica de freqüência de episódios no banco de dados de transações da empresa do setor de varejo Wal-Mart. Foram selecionadas e analisadas as movimentações financeiras de 135, com os dados relativos dos períodos de 1999 e 2000. Neste mesmo trabalho também foi aplicado freqüência de episódios em seqüências de cadeias de DNA, para o reconhecimento de regiões com mesmos padrões ao longo das cadeias. Neste trabalho será analisado o padrão anômalo em redes de computadores.

Este capítulo descreve em detalhes a técnica de episódios freqüentes. O subtópico 3.1.1 definirá formalmente o conceito de seqüência de eventos e de janela de tempo, o 3.1.2 fará o mesmo para episódios e sub-episódio. Posteriormente, em 3.1.3 e 3.1.4, serão mostradas como são calculadas a ocorrência e freqüência de um episódio para que em 3.1.5 possa ser mostrado como é realizada a descoberta dos episódios freqüentes.

3.1 Conceituação Básica

3.1.1 Seqüência de eventos

De acordo com Mannila et al. [17], dado um conjunto *E* de tipos de evento, um evento é um par (*A*, *t*), onde $A \in E$ é um tipo de evento e *t* é um inteiro, o tempo de ocorrência de um evento.

Uma seqüência de eventos **s** em *E* é um conjunto (s, T_s, T_e), onde

$$s = < (A_1, T_1), (A_2, T_2), \dots, (A_n, T_n) >$$

é uma seqüência de eventos ordenada tal que $A_i \in E$ para todo $i = 1, \dots, n$, e $t_i \leq t_{i+1}$ para todo $i = 1, \dots, n-1$. Further on, T_s e T_e são dois inteiros que representam o tempo de início e término, respectivamente, e $T_s \leq t_i < T_e$ para todo $i = 1, \dots, n$.

A figura 3.1 ilustra uma seqüência de eventos **s** = $(s, 60, 81)$ onde

$$s = < (C, 60), (B, 62), (M, 67), (B, 69), (C, 70), (K, 73), (G, 77), (B, 78), (M, 80), (C, 81) >$$

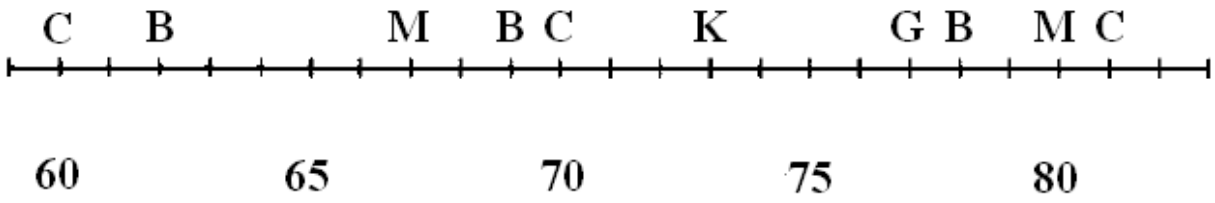

Figura 3.1: Representação gráfica de uma seqüência de eventos s.

Observando a seqüência de evento da figura 3.1, percebe-se que onde ela teve início no tempo 59 e término no tempo 83. Desta forma, pode-se afirmar que na seqüência **s** com 8 eventos, cada um deles ocorreu no intervalo de tempo [59; 83;.

Janela de Tempo

Uma vez que o objetivo da análise da freqüência de eventos é identificar todos os episódios freqüentes de uma classe de episódios, devem ser definidos intervalos de tempo, chamados de janela de tempo (*Time Window*), dentro do qual o evento pode ocorrer. Mannila et AL. [17] definem uma janela de tempo como uma fatia de uma seqüência de evento e consideram uma seqüência de evento com uma seqüência de janelas parcialmente sobrepostas. Em relação ao

tamanho da janela, cabe ao usuário definir seu tamanho de para que um episódio possa ser considerado freqüente.

Formalmente, uma janela de tempo em uma seqüência **s** = (s, T_s, T_e) é uma seqüência de eventos **w** = (w, t_s, t_e), onde $t_s < T_e$ e $t_e > T_s$, e *w* é formado um par (*A*, *t*) de **s** onde $t_s \leq t < t_e$. A diferença de tempo entre $t_s - t_e$ é chamada de tamanho da janela **w** e é representada por *width*(**w**). Desta foram, dado uma seqüência de evento **s** e um valor inteiro ***win***, o conjunto de todas as janelas **w** de tamanho *win* na seqüência **s** são denotadas por $W(s, win)$.

De acordo a definição, a primeira e a última janela em uma seqüência se entendem para fora da seqüência, tal que a primeira janela contenha somente o primeiro ponto de tempo da seqüência e a última janela contenha apenas o último ponto de tempo, permitindo que um evento seja observado igualitariamente em uma seqüência.

3.1.2 Episódios

Episódios podem ser definidos como uma coleção de eventos ordenados parcialmente de acordo com sua ocorrência. Formalmente, um episódio pode ser definido como $(V, \leq, m)$, onde *V* é o conjunto de todos os nós do episódio, ≤ é a ordem em que os eventos ocorrem no episódio, e *m* é a função que faz o mapeamento ($m : V \rightarrow Seq$) dos nós com seus respectivos tipos de eventos em uma seqüência de eventos.

Existem três tipos de episódios: paralelos, seriais e episódios não paralelos e não seriais. A figura 3.2 ilustra esses tipos e serve como auxilio de suas definições.

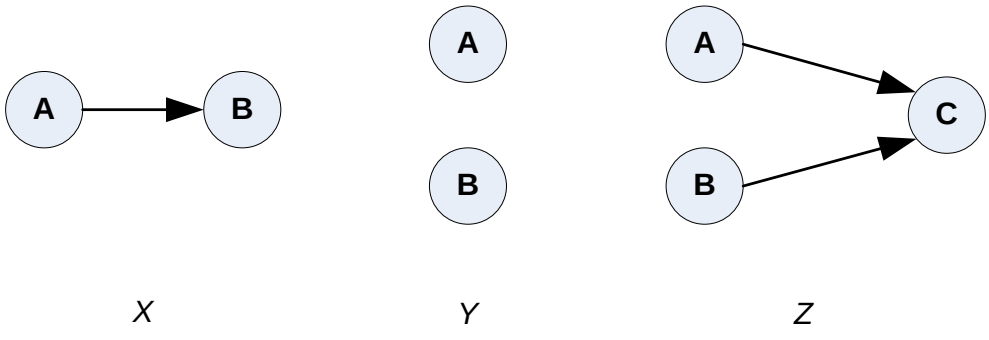

Figura 3.2: X, Y e Z representam respectivamente um episódio paralelo, um serial e um não serial e não paralelo.

O episódio *X* é dito serial se na seqüência somente se eventos do tipo A e B ocorrerem na ordem, isto é, a ordem em que os elementos ocorrem na seqüência de eventos importa.

Sendo assim, formalmente, A e B $\in$ V, A $\leq$ B $\neq$ B $\leq$ A se A $\neq$ B para todo A e B pertencentes ao episódio *X*. Já um episódio *Y* é dito paralelo por não possuir restrição da ordem de acontecimento dos eventos, isto é, a ordem temporal em que eles ocorrem não importa. Formalmente, $\leq$ é tal que para todo A e B $\in$ V, sendo A $\neq$ B. Um episódio *Z* é nomeado não paralelo e não serial se dois eventos A e B precederem um evento C e não existir nenhuma restrição quanto à ordem de A e B.

Sub-episódios

O conceito de sub-episódio, bastante empregado em técnicas de descobertas de padrões em seqüência, baseia-se na premissa de que uma seqüência pode conter outra. Quando uma seqüência é parte do outra, ela é chamada de sub-episódio.

Usando a figura 3.2 como exemplo, pode-se afirmar que *Y* é um sub-episódio de *Z*, uma vez que existe um mapeamento m que interliga os nós A e B com outros, isto é, ambos os nós de *Y* tem nós correspondentes em *Z*. Formalmente, $Y = (V', \leq', m')$ é dito sub-episódio de $Z = (V, \leq, m)$, denotado por $Y \leq Z$, porque existe um mapeamento $f: V' \rightarrow V$ tal que $m'(v) = m(f(v))$ para todo $v \in V'$ e para todo $v, w \in V'$ com $v \leq' w$) também $f(v) \leq f(w)$.

3.1.3 Ocorrência de um episódio

Um episódio só ocorre em uma seqüência de eventos se todos os seus eventos ocorrem nessa seqüência de eventos e sua ordem parcial é respeitada.

Formalmente, um episódio $X = (V, \leq, m)$ ocorre em uma seqüência de eventos $s = < (A_1, T_1), (A_2, T_2), \ldots, (A_n, T_n) >$ se existir um mapeamento representado pela função $f: V \rightarrow \{1,2,3,\ldots,n\}$ de todos os nós de *X* para o evento *s* tal que $m(x) = A_{f(x)}$ para todo $x \in V$, e para todo $x, y \in V$ com $x \neq y$ e $x \leq y$, então tem-se que $t_{f(x)} < t_{f(y)}$.

3.1.4 Freqüência de um episódio

A freqüência de um episódio é definida por Mannila et al. [17] como sendo as frações de janelas na qual o episódio ocorre. Isto é, dado uma seqüência de evento *s* e uma janela de tamanho *win*, a freqüência de um episódio *E* em *s* é:

$$fr(E, s, win) = \frac{|\{\boldsymbol{w} \in W(\boldsymbol{s}, win) | E \ ocorre \ em \ \boldsymbol{w}\}|}{|W(\boldsymbol{s}, win)|}$$

Para determinar se um episódio *E* é ou não freqüente, um limiar de freqüência (*min_fr*) é utilizado. Sendo assim, E é dito freqüente se $fr(E, s, win) \geq min\_fr$. A representação do conjunto de episódios freqüentes em relação à s é dada por $\mathcal{F}(s, win, \min\_fr)$. É importante ressaltar que se um episódio é freqüente, então todos os seus sub-episódios também serão freqüentes. Tal premissa é bastante importante para a diminuição do custo do cálculo da geração dos candidatos a episódios freqüentes que será explicado adiante.

3.1.5 Descoberta da freqüência dos episódios

Uma vez que os episódios freqüentes são conhecidos, eles podem ser utilizados na obtenção de correlações entre os eventos da seqüência. Essas relações são chamadas de regras de episódio (*Episodes Rules*). Uma regra entre dois episódios *X* e *Y* é definida formalmente como $X \Rightarrow Y$, denominada R_{xy}, se *X* é sub-episódio de *Y*. Por exemplo, se os episódios $(A \rightarrow B)$ e $(A \rightarrow B \rightarrow C)$ são freqüentes, com freqüências f_1 e f_2 respectivamente, a regra resultante é $(A \rightarrow B) \Rightarrow (A \rightarrow B \rightarrow C)$ se a confidência estabelecida $(\frac{f_2}{f_1})$ ultrapassar um predefinido limite (*threshold*).

A confidência (*confidence*) de uma regra é uma fração entre a confidência de um sub-episódio por um episódio. Em outras palavras, representa a probabilidade condicional do episódio *Y* ocorrer por completo em uma janela, dado que o episódio *X* ocorreu na mesma.

Mannila et al. [17] propõem duas abordagens para o cálculo da freqüência dos episódios: uma baseada no número de janelas e outra em ocorrências mínimas.

4 Projeto e Implementação

Este capítulo descreve o funcionamento do protótipo implementado, bem como seus componentes e o processo de desenvolvimento. Na seção 4.1 será mostrada uma visão geral do protótipo bem como seu processo de funcionamento. Na subseção 4.2 cada um dos componentes (módulos) do protótipo será explicado detalhadamente.

4.1 Protótipo

O protótipo foi desenvolvido com o intuito de ser capaz de aumentar a precisão dos alertas e prever futuros alertas de acordo com o estado da rede. Na prática representa um software capaz de, baseado em saídas (alertas) geradas por ferramentas de detecção de anomalias, aumentar a eficiência da detecção, assim como prevê-las através da correlação dos alertas.

A idéia central é utilizar a técnica de freqüência de episódios para correlacionar os alertas gerados por atividades intrusivas e maliciosas, ordenando-os temporalmente, e confirmando a existência de anomalias ou ataques no tráfego da rede.

Em relação a arquitetura, o protótipo é modular, o que facilita a manutenção, diminui a repetição de código e melhora sua legibilidade. A figura 4.1 ilustra a arquitetura do protótipo.

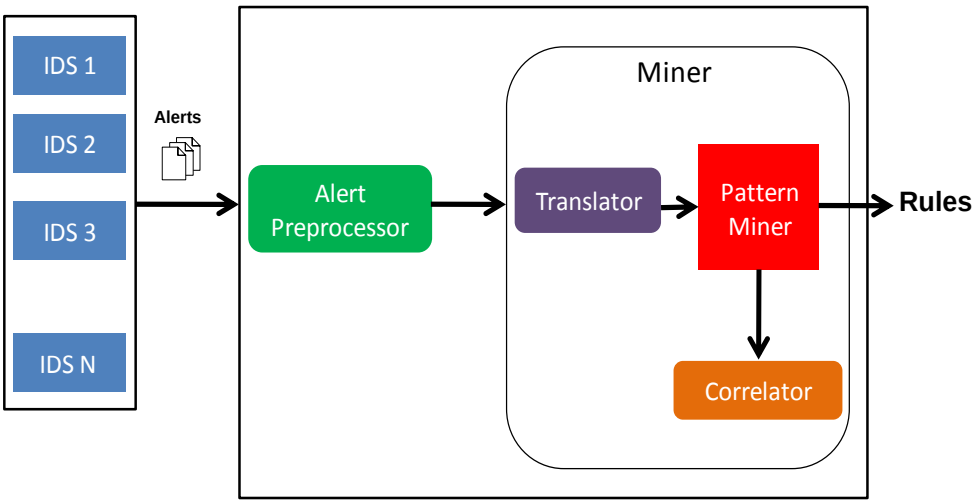

Figura 4.1: Arquitetura do Protótipo.

As subseções seguintes fazem uma análise da arquitetura do protótipo.

4.1.1 Detectores de anomalia

Apesar de não ser parte implementada da arquitetura, os detectores de anomalia são responsáveis pela análise do tráfego da rede e geração de alertas, caso alguma anomalia seja encontrada. Uma vez que existem diferentes tipos de ferramentas, também existem vários tipos de saída. Por este motivo, se faz necessária a padronização do formato dos alertas.

O padrão escolhido foi o IDMEF (*Intrusion Detection Message Exchange Format*) [32]. Criado pelo grupo IDWG (*Intrusion Detection Exchange Format Working Group*), pertencente ao IETF (*Internet Engineering Task Force*), o IDMEF é uma linguagem de marcação XML para representar os dados de alertas que são organizados de forma hierárquica, com um formato independente das plataformas de software, hardware ou de um banco de dados.

A figura 4.2 exemplifica uma alerta em IDMEF. O ataque representado na figura ocorreu em 16 de abril de 2000 (linha 10), tem como atacante o IP 202.77.162.213 (linha 15), o alvo 172.16.115.20 (linha 22) e classificado como *buffer-overflow* (linha 29).

```xml
<?xml version="1.0"?>
<!DOCTYPE IDMEF-Message PUBLIC"/usr/local/share/idmef-message.dtd">
<IDMEF-Message version="1.0">
  <Alert messageid="3004">
    <Analyzer analyzerid="IDS1" manufacturer="Joe McAlerney" version="2.8.3.2" >
      <Node category="unknown">
        <name>teste</name>
      </Node>
    </Analyzer>
    <CreateTime ntpstamp="0xce5110b5.0xe1117b52">2000-04-16T16:17:47Z</CreateTime>
    <AnalyzerTime ntpstamp="0xce5110b5.0xe1e54b48">2000-04-16T16:17:47Z</AnalyzerTime>
    <Source>
      <Node category="unknown">
        <Address category="ipv4-addr">
          <address>202.77.162.213:823</address>
        </Address>
      </Node>
    </Source>
    <Target>
      <Node category="unknown">
        <Address category="ipv4-addr">
          <address>172.16.115.20:33895</address>
        </Address>
      </Node>
      <Service ip_version="4">
        <portlist></portlist>
      </Service>
    </Target>
    <Classification text=" RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt">
      <Reference origin="vendor-specific" meaning="sfportscan">
        <name> RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt</name>
        <url>http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0977][Xref</url>
      </Reference>
    </Classification>
    <Assessment>
      <Impact type="recon" severity="low"> Attempted Administrator Privilege Gain]</Impact>
    </Assessment>
  </Alert>
</IDMEF-Message>
```

Figura 4.2: Exemplo de alerta IDMEF.

4.1.2 Módulo de Tradução

O módulo de tradução é responsável pela tradução, ou adequação, dos alertas recebidos em ordem cronológica para o formato de análise pelo módulo de mineração de padrões.

Para tanto, cada alerta recebido é correlacionado com um tipo de evento (*event type*) para o cálculo dos episódios freqüentes. Os atributos escolhidos para identificar um alerta são: horário em que o alerta ocorreu, endereço (IP e porta) do nó que gerou o alerta, endereço (IP e porta) do nó alvo e o tipo do alerta. Desta forma, são criadas estruturas que possuem como atributo uma lista de eventos que somente contem os alertas que ocorreram no mesmo intervalo de tempo (segundo). Essas estruturas são colocadas em uma lista, ordenada de forma crescente em relação ao horário em que os tipos de eventos ocorreram.

Uma representação simbólica da lista é mostrada na figura 4.3. Posteriormente, a lista é enviada para o módulo de mineração de padrões, juntamente com a tabela de correlação entre tipos de evento e alertas.

```
1 A I
2 C T F C
3 B O
7 A X K M
8 A O
9 B
10 C K P D
11 B O
12 C Q
13 C M O V
14 A Y U
15 A L P J
16 A
18 A M
19 A S H U L R J I
25 B
26 D
27 A
28 C G
```

Figura 4.3: Representação da seqüência de eventos realizada pelo módulo de tradução.

Além desta função, este módulo gera uma tabela de correspondência, onde os tipos de evento e os atributos dos alertas correspondentes a eles são relacionados. Os dados contidos na tabela de correspondência são representados graficamente na figura 4.4.

```
B ( ICMP redirect host 172.16.114.10 ---> 172.16.114.1) Has 32 Alerts.
C ( ATTACK-RESPONSES directory listing 196.37.75.158:3841 ---> 172.16.112.100:23) Has 16 Alerts
D ( WEB-MISC RBS ISP /newuser access 134.205.131.13:80 ---> 172.16.112.194:65236) Has 8 Alerts.
E ( WEB-CGI redirect access 209.185.123.57:80 ---> 172.16.113.207:63824) Has 2 Alerts.
F ( RPC portmap sadmind request UDP 172.16.115.20:111 ---> 202.77.162.213:822) Has 2 Alerts.
G ( RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt 172.16.115.20:33911 --->
```

Figura 4.4: Representação gráfica da tabela de correspondência.

4.1.3 Módulo de Mineração de Padrões

O módulo de mineração de padrões é responsável pelo cálculo dos episódios freqüentes e da geração das regras. Em linhas gerais, recebe uma lista contendo os tipos de eventos e calcula os episódios freqüentes de acordo com o tamanho da janela estipulado e do limiar estabelecido. Para tanto, utiliza quatro funções:

- **Coletor de Eventos**: cuja função é percorrer a estrutura que contêm os tipos de evento, devolvendo os eventos freqüentes;

- **Gerador de Candidatos**: cuja função é receber os episódios freqüentes de tamanho X e gerar os candidatos de tamanho X+1;
- **Gerador de Episódios Freqüentes**: cuja função é calcular todos os episódios freqüentes, utilizando as funções do gerador de candidatos e coletor de eventos e retornando os episódios freqüentes;
- **Gerador de Regras**: tem a função de gerar regras baseado nos episódios freqüentes enviados pelo gerador de episódios freqüentes.

5 Avaliação e Resultados

Este capítulo descreve o processo de avaliação de resultados utilizado neste trabalho, assim como as métricas de avaliação, descrição do ambiente de teste e ferramentas utilizadas na geração do tráfego analisado.

Para tanto, a avaliação foi dividida em dois experimentos. No primeiro foram utilizadas as bases de dados do *DARPA 2000 Intrusion Detection Scenario Specific Data Set* [19], uma base de tráfego bastante conhecida na literatura e empregada em dezenas de trabalhos. A idéia é avaliar a precisão de detecção das anomalias, correlacionando os alertas gerados pelo Snort com as anomalias existentes no cenário que a base apresenta, detectando verdadeiros e falsos alertas.

No segundo experimento foi utilizado tráfego real capturado na rede do Grupo de Pesquisa em Redes de Computadores (GPRT) do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE). O objetivo deste experimento é validar a eficiência e precisão em ambiente real.

5.1 DARPA 2000 dataset

O DARPA 2000 dataset [17] é uma conhecida base de informação para avaliação de IDSs. Criada pelo *MIT Lincoln Laboratory*, contém dois cenários: LLDOS 1.0 e LLDOS, onde, em ambos, o tráfego coletado pertence tanto a uma rede externa, no caso uma zona desmilitarizada (do inglês *DeMilitarized Zone* - DMZ), quanto a uma rede interna.

5.1.1 LLDOS 1.0

O LLDOS 1.0 é dividido em 5 fases:

- **Fase 1**: o atacante apenas envia mensagens do tipo ICMP para tentar descobrir quais hosts da sub-rede estão realmente ativos. Os pacotes são enviados as sub-redes 172.16.115.0/24, 172.16.114.0/24, 172.16.113.0/24, 172.16.112.0/24.
- **Fase 2**: de posse dos hosts ativos nas sub-redes vasculhadas na fase anterior, o atacante executa um ferramenta tipo *exploit* para determinar se o serviço *sadmind* está executando.
- **Fase 3**: uma vez que o atacante conhece os hosts ativos executando o serviço *sadmind*, o atacante inicia várias tentativas para ter acesso de super usuário (root)

nestes hosts, empregando, em cada tentativa, parâmetros diferentes, executando um ataque do tipo buffer-overflow.

- **Fase 4**: uma vez que consegue os privilégios de *root*, o atacante executa os comandos *Telnet* e *rpc* para permitir a realização de ataques DDoS a partir desses hosts. Um arquivo do tipo ".rhosts" e outro software chamado de *master-sol* são copiados e instalados nos hosts.
- **Fase 5**: com controle absoluto de 3 hosts (172.16.115.20, 172.16.112.10, 172.16.112.50), o atacante inicia um ataque DDoS apenas executando o comando "mstream 131.84.1.31 5", que obriga as máquinas a enviarem ao mesmo tempo uma grande quantidade de pacotes para o alvo 131.84.1.31 durante 5 segundos, com endereços IP de origem com valores aleatórios.

5.1.2 LLDOS 2.0.2

Similar ao LLDSO 1.0, o cenário LLDOS 2.0.2 também é dividido em 5 fases, com o objetivo de realizar um ataque DDoS. A grande diferença deste cenário é a utilização de consultas do tipo HINFO a um servidor DNS. Com essas consultas, o atacante obtém informações de *host* gravadas no servidor DNS, entre elas: plataforma, sistema operacional e endereço.

- **Fase 1**: o atacante faz consultas do tipo HINFO[1] ao servidor DNS da rede (172.16.1145.20) informações como a plataforma e sistema operacional de possíveis vítimas. Desta forma, o atacante pode escolher a melhor técnica de ataque baseado nas configurações de cada vítima.
- **Fase 2**: o atacante consegue invadir o servidor DNS explorando a vulnerabilidade do serviço *sadmind*.
- **Fase 3**: através de uma conexão FTP, o atacante injeta no servidor DNS o programa *mstream para* realizar ataques DDoS.
- **Fase 4**: o atacante tenta obter acesso como super usuário em mais dois hosts, mas só obtém sucesso em uma delas, onde também instala o programa *mstream*.
- **Fase 5**: com controle de 2 hosts, o atacante inicia um ataque DDoS ao mesmo alvo do cenário LLDOS1.0 (131.84.1.310) e com a mesma duração (5 segundos) 5", que obriga as máquinas a enviarem ao mesmo tempo uma grande quantidade de pacotes para o alvo 131.84.1.31 durante 5 segundos, com endereços IP de origem com valores aleatórios.

[1]

5.1.3 Geração de alertas dos cenários

Como mencionado no capítulo 4, o esquema de correlação utiliza alertas no formato IDMEF como entrada. Desta forma, foi utilizado o IDS Snort (versão 2.8.3.2) para geração dos alertas para ambos os cenários.

LLDOS1.0

Para o primeiro cenário, o Snort registrou 1002 alertas para o arquivo com tráfego capturado internamente (*inside-tcpdump*) e 2646 alertas para o arquivo com tráfego capturado externamente (*dmz-tcpdump*). Na fase 1, o Snort não detectou qualquer atividade relacionado aos *ICMP requests* feitos pelo atacante. Contudo, foi capaz de detectar os acessos ao serviço *sadmind* (porta 111) da fase 2, gerando alertas rotulados como ***RPC portmap sadmind request UDP*** (figura 5.1)

```
[**] [1:585:7] RPC portmap sadmind request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
03/07-12:11:09.493558 202.77.162.213:653 -> 172.16.114.50:111
UDP TTL:62 TOS:0x0 ID:10718 IpLen:20 DgmLen:84
Len: 56
[Xref => http://www.whitehats.com/info/IDS20]

[**] [1:585:7] RPC portmap sadmind request UDP [**]
[Classification: Decode of an RPC Query] [Priority: 2]
03/07-12:11:09.503613 202.77.162.213:654 -> 172.16.113.1:111
UDP TTL:62 TOS:0x0 ID:10719 IpLen:20 DgmLen:84
Len: 56
[Xref => http://www.whitehats.com/info/IDS20]
```

Figura 5.1: Alertas Snort do ataque RPC portmap sadmind request UDP.

Na fase 3, foram observados e registrados múltiplos alertas *RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt* e *RPC sadmind query with root credentials attempt UDP*. O primeiro indica um ataque de buffer overflow sob o sadmind e segundo a tentativa de acesso não autorizado. A figura 5.2 ilustra esses alertas.

```
[**] [1:1911:11] RPC sadmind UDP NETMGT_PROC_SERVICE CLIENT_DOMAIN overflow attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
03/07-12:33:10.617541 202.77.162.213:668 -> 172.16.115.20:60250
UDP TTL:62 TOS:0x0 ID:10830 IpLen:20 DgmLen:1440
Len: 1412
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0977][Xref =>
http://www.securityfocus.com/bid/866]

[**] [1:2256:3] RPC sadmind query with root credentials attempt UDP [**]
[Classification: Misc Attack] [Priority: 2]
03/07-12:34:59.310529 202.77.162.213:699 -> 172.16.112.50:60618
UDP TTL:62 TOS:0x0 ID:11095 IpLen:20 DgmLen:1440
Len: 1412
```

Figura 5.2: Alertas de tentativas de permissões root e tentativas de buffer-overflow.

Na fase 4, o atacante usa os comandos *rsh* e *Telnet* para instalar e iniciar o programa *mstream*. O Snort foi capaz de detectar o *rsh* (figura 5.3), mas não o *Telnet*.

```
[**] [1:610:5] RSERVICES rsh root [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
03/07-12:50:02.146207 172.16.115.20:1023 -> 202.77.162.213:514
TCP TTL:255 TOS:0x0 ID:47651 IpLen:20 DgmLen:125 DF
***AP*** Seq: 0xAFDF501E  Ack: 0xF4A1E8F7  Win: 0x2238  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS391]

[**] [1:610:5] RSERVICES rsh root [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
03/07-12:50:21.344974 172.16.112.10:1023 -> 202.77.162.213:514
TCP TTL:255 TOS:0x0 ID:388 IpLen:20 DgmLen:125 DF
***AP*** Seq: 0xADC6CEB3  Ack: 0x910F41D8  Win: 0x2238  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS391]

[**] [1:610:5] RSERVICES rsh root [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
03/07-12:50:38.344016 172.16.112.50:1023 -> 202.77.162.213:514
TCP TTL:255 TOS:0x0 ID:49210 IpLen:20 DgmLen:125 DF
***AP*** Seq: 0xB31AC496  Ack: 0x74316F71  Win: 0x2238  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS391]
```

Figura 5.3: O atacante consegue obter os obter os privilégios de root.

Por fim, na fase 5, o ataque DDoS ao vítima 131.84.1.31 é reconhecido pelo Snort (figura 5.4). O primeiro alerta foi registrado às 13:27:51 e o último as 13:27:56, contabilizando exatamente 5 segundos de ataque.

```
[**] [1:528:5] BAD-TRAFFIC loopback traffic [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/07-13:27:51.183436 127.192.221.148:5184 -> 131.84.1.31:14783
TCP TTL:255 TOS:0x8 ID:23466 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x7BE9C2B3  Ack: 0x0  Win: 0x4000  TcpLen: 20
[Xref => http://rr.sans.org/firewall/egress.php]

[**] [1:528:5] BAD-TRAFFIC loopback traffic [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/07-13:27:51.209230 127.230.246.186:5348 -> 131.84.1.31:31521
TCP TTL:255 TOS:0x8 ID:23630 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x7BE9C357  Ack: 0x0  Win: 0x4000  TcpLen: 20
[Xref => http://rr.sans.org/firewall/egress.php]

[**] [1:528:5] BAD-TRAFFIC loopback traffic [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
03/07-13:27:51.229332 127.177.55.200:5452 -> 131.84.1.31:30747
TCP TTL:255 TOS:0x8 ID:23734 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x7BE9C3BF  Ack: 0x0  Win: 0x4000  TcpLen: 20
[Xref => http://rr.sans.org/firewall/egress.php]
```

Figura 5.4: Três alertas referentes ao ataque DDoS.

LLDOS 2.0

Para o segundo cenário, o Snort registrou 937 alertas para o arquivo com tráfego capturado internamente (*inside-tcpdump*) e 1109 alertas para o arquivo com tráfego capturado externamente (*dmz-tcpdump*).

Assim como no primeiro cenário, o Snort não foi capaz de gerar alertas referentes atividade da fase 1 e registrou todas as atividades da fase 2 (acessos ao serviço *sadmind*). A fase 3 não gerou nenhum tipo de alerta. Já as fases 4 e 5 foram reconhecidas e tiveram seus alertas registrados de forma similar a estas fases no cenário LLDOS1.0 (figuras 5.3 e 5.4).

5.1.4 Avaliações e resultados do DARPA 2000

A utilização do dataset DARPA 2000 para avaliação do protótipo foi escolhido devido a grande quantidade de tráfego existente e reconhecido como anômalo, fornecendo assim uma análise qualitativa.

A primeira análise efetuada foi a de total de episódios freqüentes gerados em função do tamanho da janela para cada cenário (figura 5.5).

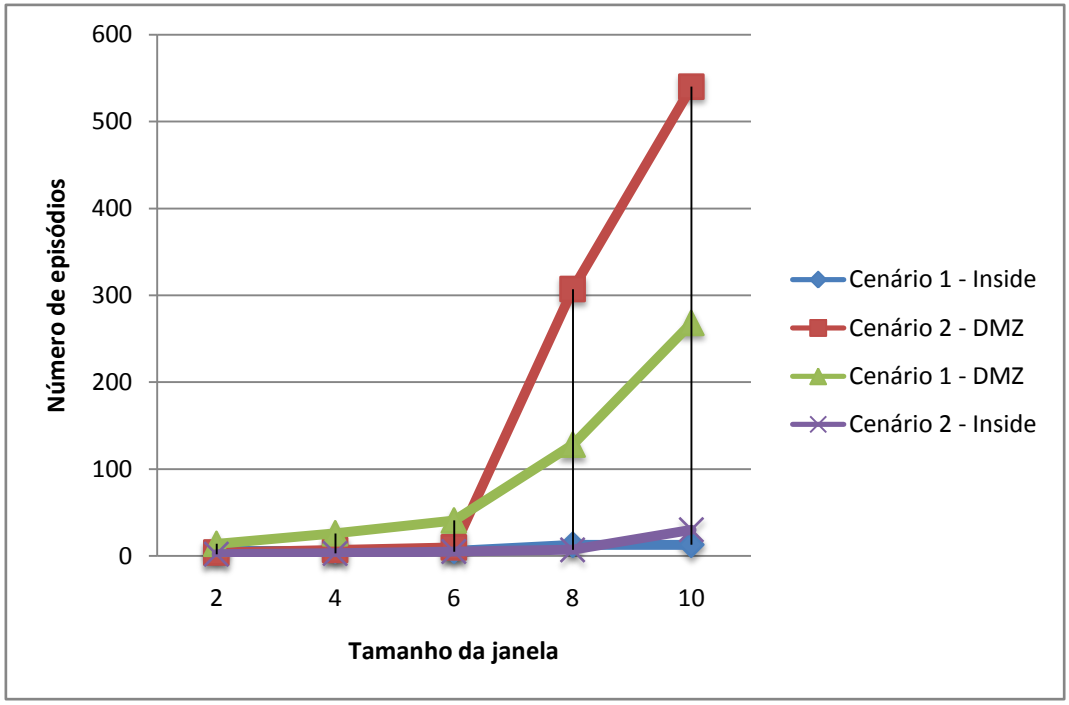

Figura 5.5: Relação número de episódios freqüentes em função do tamanho da janela, com limiar da confidência de 0.001.

Observando a figura 5.5 percebe-se que o número de episódios freqüentes cresce com o aumento do tamanho da janela. Nos cenários 1 e 2 inside, o número de episódios freqüentes é bem inexpressivo quantativamente se comparados aos cenários 1 e 2 DMZ. Após uma análise na tabela de correspondência, do módulo de tradução, observou-se que a grande maioria dos tipos de eventos (mais de 98%) que ocorriam nos cenários 1 e 2 DMZ não se repetiam nos cenários 1 e 2 inside. Tal fato se deve ao ataque DDoS gerado nas duas redes internas, que ocasionou uma grande quantidade de pacotes com endereços de origem com valores aleatórios e, conseqüentemente, foram criados milhares alertas (gerado pelo Snort) que quando correlacionados não apresentavam potencial de repetição, ou seja, não eram freqüentes.

É importante ressaltar que segundo Mannila et al. [17], o aumento do número de episódios freqüentes de acordo com o aumento do tamanho da janela não ocorre em todos os tipos de dados analisados.

A figura 5.6 exemplifica a disposição dos tipos de eventos durante o ataque DDoS no cenário 1 inside (o mesmo ocorre no cenário 2 inside).

```
6099 N2
6195 E5
6197 F5
6230 G5
6300 H5
6389 I5
6633 J5
6646 K5
6736 L5
7366 M5
7368 N5
7419 O5 P5 O5
7557 Q5 R5 S5 T5 U5 V5 W5 X5 Y5 Z5 A6 B6 C6 D6 E6 F6 G6 H6 I6 J6 K6 L6 M6 N6 O6 P6 Q6 R6 S6
7558 Y9 Z9 A10 B10 C10 D10 E10 F10 G10 H10 I10 J10 K10 L10 M10 N10 O10 P10 Q10 R10 S10 T10 U
7559 Y15 Z15 A16 B16 C16 D16 E16 F16 G16 H16 I16 J16 K16 L16 M16 N16 O16 P16 Q16 R16 S16 T16
7560 O21 P21 Q21 R21 S21 T21 U21 V21 W21 X21 Y21 Z21 A22 B22 C22 D22 E22 F22 G22 H22 I22 J22
7561 E27 F27 G27 H27 I27 J27 K27 L27 M27 N27 O27 P27 Q27 R27 S27 T27 U27 V27 W27 X27 Y27 Z27
7562 R33 S33 T33 U33 V33 W33 X33 Y33 Z33 A34 B34 C34 D34 E34 F34 G34 H34 I34 J34 K34 L34 M34
7717 W37
7860 X37
8086 Y37
8088 Z37
8385 A38
8806 B38
8808 C38
9526 D38
9528 E38
9531 F38 G38 F38
9679 H38
9974 I38
10246 J38
10248 K38
10344 L38
10631 M38 N38
10651 O38 P38
```

Figura 5.6: Disposição dos tipos de evento em relação ao ataque DDoS no cenário 1 inside.

Como se observa na figura 5.6, o padrão normal da rede foi quebrado pelo ataque DDoS, entre os tempos 7557 e 7562. Vale ressaltar apesar de o ataque DDoS ocorrer em apenas 5 segundos, as aproximações temporais para análise dos alertas geraram 1 segundo a mais.

O ataque DDoS também fica claro quando é analisado os tipos de eventos em ambos os cenários. Na figura 5.7 pode-se observar a disparidade na quantidade distinta de tipos de eventos.

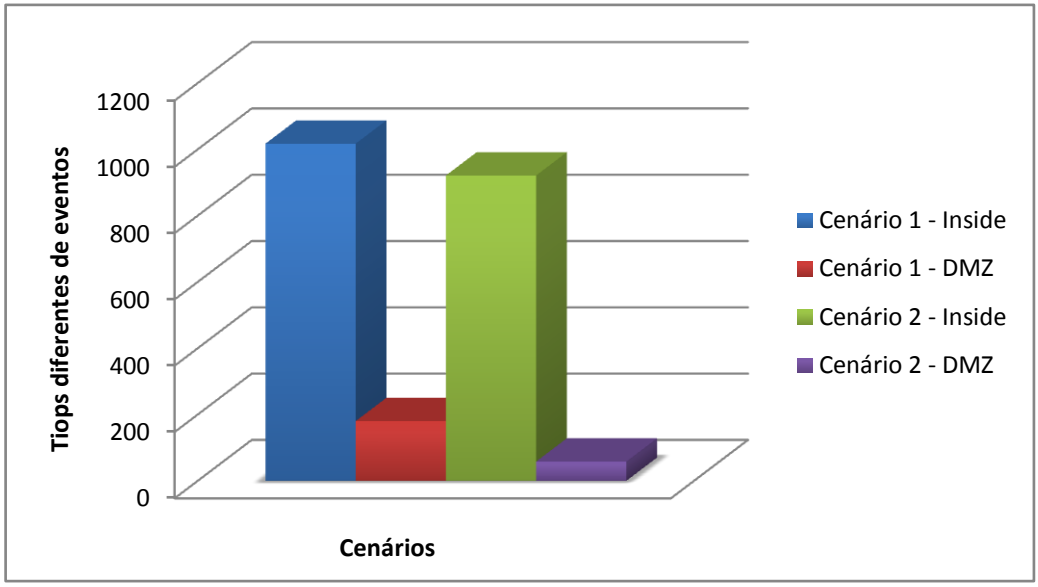

Figura 5.7: Diversidade de tipos de eventos em todos os cenários causada pelo ataque DDoS.

Por fim, uma análise das regras geradas é extraída do cenário 1 DMZ. A figura 5.8 ilustra as regras obtidas, de acordo com as confidências definidas, para os eventos ocorridos quando o atacante tenta obter privilégios root através do sadmind (seção 5.1.1)

```
Rule 302:  A --------> AAAAAAAN1 with confidence 0,02.
Rule 303:  N1 --------> AAAAAAAN1 with confidence 0,62.
Rule 304:  AA --------> AAAAAAAN1 with confidence 0,02.
Rule 305:  AN1 --------> AAAAAAAN1 with confidence 0,98.
Rule 306:  AAA --------> AAAAAAAN1 with confidence 0,03.
Rule 307:  AAN1 --------> AAAAAAAN1 with confidence 0,98.
Rule 308:  AAAA --------> AAAAAAAN1 with confidence 0,03.
Rule 309:  AAAN1 --------> AAAAAAAN1 with confidence 0,98.
Rule 310:  AAAAA --------> AAAAAAAN1 with confidence 0,03.
Rule 311:  AAAAN1 --------> AAAAAAAN1 with confidence 0,98.
Rule 312:  AAAAAA --------> AAAAAAAN1 with confidence 0,03.
Rule 313:  AAAAAN1 --------> AAAAAAAN1 with confidence 0,98.
Rule 314:  AAAAAAA --------> AAAAAAAN1 with confidence 0,03.
Rule 315:  AAAAAAN1 --------> AAAAAAAN1 with confidence 1,00.
```

Figura 5.8: Regras e confidências de um *sadmind request*.

Uma vez que o evento A indica a realização de um *port scan* na rede e o evento N1 a tentativa de obter privilégios de root pelo programa sadmind, a figura 5.8 indica, com suas respectivas confidências, a probabilidade de um atacante estar tentando realizar um ataque ao *sadmind*. Desta forma, a presença de grandes quantidades de eventos A serve de indicativo da presença do evento N1 no futuro. Sendo assim, é possível realizar predições, com determinado percentual de garantia, sobre a ocorrência de um evento dado que outro ocorreu.

5.2 GPRT

Para realizar s testes em ambiente real foram utilizadas as instalações do Grupo de Pesquisa em Redes e Telecomunicações (GPRT) do Centro de Informática (CIn) da UFPE, onde existe uma rede composta por mais de 60 computadores e dois ponto de saída: um com conexão a Internet via PoP-PE (ITEP) e outro com a rede da UFPE via NTI.

Para coletar o tráfego dessas redes, o Snort (versão 2.8.3.2) foi instalado nos dois servidores gateways (um para cada conexão) e foram configuradas as mesmas regras para detecção de intrusões. O tráfego foi coletado durante os dias 17 e 19 de novembro de 2009, iniciando as 14:29:55 horas do dia 17 e terminando as 18:18:27 horas do dias 19.

5.2.1 Análise do tráfego do GPRT

Diferente do cenário anterior, o tráfego do GPRT é útil para a realização de análises quantitativas, uma vez que não existe conhecimento prévio sobre o tráfego capturado.

A tabela 5.1 contém os dados relativos à execução do protótipo com tráfego do GPRT.

Tabela 5.1: Resultado da geração de episódios freqüentes no tráfego GPRT.

Tamanho do Episódio	Possíveis Episódios	Candidatos	Episódios Freqüentes	Correspondem
1	*348*	*348*	*11*	*3,1%*
2	*82369*	*121*	*4*	*3,3%*
3	2.10^{7}	*8*	*5*	*62,5%*
4	7.10^{9}	*7*	*6*	*85,7%*
5	2.10^{12}	*8*	*6*	*75%*
6	6.10^{14}	*8*	*7*	*87,5%*
7	2.10^{17}	*9*	*6*	*66,6%*
8	5.10^{19}	*8*	*4*	*50%*
9	1.10^{22}	*4*	*2*	*50%*
10	4.10^{24}	*2*	*1*	*50%*

De modo geral, pode-se observar que a técnica da geração de candidatos diminui bastante o custo do cálculo dos episódios freqüentes.

O gráfico da figura 5.9 mostra a relação entre a quantidade de regras e o limiar (*threshold*) da confidência. Este gráfico é útil porque exprime a noção, em ordem de grandeza, de que a quantidade de regras aumenta em relação ao limiar da confidencia.

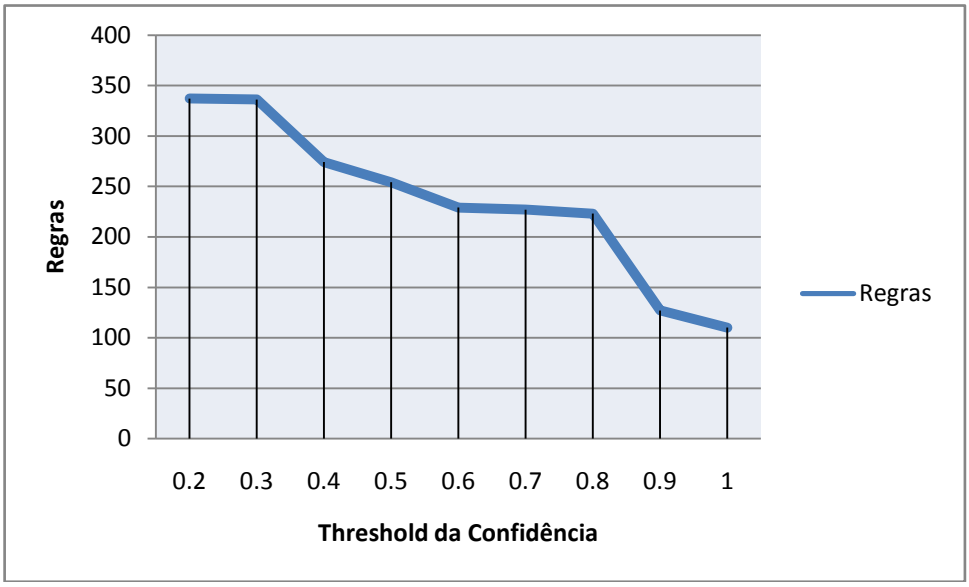

Figura 5.9: Relação número de regras e limiar da confidência.

6 Conclusão

Hoje em dia, anomalias continuam causando inúmeros prejuízos a empresas e instituições. Ataques de negação de serviço, scans, worms, vírus e outros tipos de males ainda geram problemas a milhares de administradores de rede e até mesmo a usuários comuns. Apesar do desenvolvimento e aperfeiçoamento das técnicas de detecção e predição de anomalias, muito do tráfego que circula na rede ainda é malicioso, causando prejuízos econômicos e financeiros. Desta forma, esse trabalho contribuiu para o estudo e implementação de um protótipo que faz a detecção e predição de anomalias baseado em freqüência de episódios.

Este documento apresentou, de maneira geral, uma análise da situação atual de anomalias em tráfego de rede, assim como, um estudo do estado da arte dos principais trabalhos relacionados com detecção e predição de anomalias. Posteriormente foi mostrada com detalhes a técnica de freqüência de episódios, com uma abordagem teórica.

Depois, foi apresentada a arquitetura, implementação, cenários, testes e resultados do protótipo que faz detecção e predição de anomalias em tráfego de redes de computadores. Os resultados indicaram que o protótipo conseguiu detectar anomalias no tráfego, a exemplo do ataque DDoS, que foi o principal ataque do cenário de testes. Através das regras geradas pelo protótipo também foi possível detectar padrões que podem ser utilizados para predição de anomalias.

6.1 Dificuldades encontradas

Durante o desenvolvimento do deste trabalho foram encontrados vários desafios que, ao mesmo tempo, tornaram a jornada mais difícil, mas que também, após ultrapassá-los, engrandeceu e trazendo mais qualidade ao resultado final.

Entre os desafios encontrados pode-se citar:

- Entendimento detalhado da parte teórica de freqüência de episódios, apesar de ser um assunto desafiador e bastante estimulante, possui muitos detalhes teóricos, a exemplo da parte que trata da sua complexidade algorítmica.
- Outro desafio foi a manipulação e aquisição dos tráfegos para rodar no protótipo. No caso do tráfego do DARPA, o problema foi o plugin do Snort, que por algum motivo desconhecido, não gerou os alertas no formato IDMEF. A saída foi fazer um parser, que lê as os alertas no formato padrão do Snort e transforma no formato IDMEF. No caso do tráfego do GPRT, o problema foi que foi tentado sem sucesso instalar o plugin do IDMEF No gateway que colhia o tráfego. Desta forma, foi

necessário colher o tráfego e salva-lo no formato .pcap, e depois rodar-lo no Snort, em uma máquina que tem suporte ao plugin.

6.2 Trabalhos futuros

Alguns trabalhos futuros são:

- Implementar no protótipo um módulo que ao gerar as regras do episódios freqüentes, realize alguma ação para evitar os ataques que estão previstos (com uma confidência elevada) de ocorrer. Alguns desses mecanismos podem ser: transformar a regra do episódio freqüente em uma regra que realimenta o IDS, aplicar alguma política que manipule o firewall à medida que o protótipo é utilizado, bloquear os pacotes que chegar de um host cujo ataque está previsto, entre outros.
- Outra melhoria pode ser de combinar outras técnicas que também façam a correlação e predição de eventos, para maximizar a taxa de acerto nas anomalias das redes.

Referências

[1] P. Hayati and V. Potdar. Evaluation of spam detection and prevention frameworks for email and image spam - a state of art.The 2nd International Workshop on Applications of Information Integration in Digital Ecosystems (AIIDE 2008), Novermber2008.

[2] K. Julisch. Mining alarm clusters to improve alarm handling efficiency. In Proceedings of the17th Annual Computer Security Applications Conference (ACSAC), pages 12–21, December2001.

[3] K. Julisch. Dealing with false positives in intrusion detection. In The 3th Workshop on RecentAdvances in Intrusion Detection, October 2000.

[4] H. Debar and A. Wespi. Aggregation and correlation of intrusion-detection alerts. In Recent Advances in Intrusion Detection, LNCS 2212, pages 85 – 103, 2001.

[5] O. Dain and R.K. Cunningham. Fusing a heterogeneous alert stream into scenarios. In Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications, pages 1–13, November 2001.

[6] B. Morin and H. Debar. Correlation of intrusion symptoms: an application of chronicles. In Proceedings of the 6th International Conference on Recent Advances in Intrusion Detection (RAID'03), September 2003.

[7] S.T. Eckmann, G. Vigna, and R.A. Kemmerer. STATL: An Attack Language for State-based Intrusion Detection. Journal of Computer Security, 10(1/2):71–104, 2002.

[8] S. Templeton and K. Levit. A requires/provides model for computer attacks. In Proc. of New Security Paradigms Workshop, pages 31-38. September 2000.

[9] F. Cuppens and A. Miege, "Alert Correlation in a Cooperative Intrusion Detection Framework," in IEEE Security and Privacy, 2002.

[10] S. Staniford, J. Hoagland, and J. McAlerney. Practical automated detection of stealthy portscans. To appear in Journal of Computer Security, 2002.

[11] Xinzhou Qin , Wenke Lee, A probabilistic-based framework for infosec alert correlation, Georgia Institute of Technology, Atlanta, GA, 2005.

[12] [Qin and Lee 03] Qin, X., Lee, W.: "Statistical Causality Analysis of INFOSEC Alert Data"; Proc. 6th International Symposium on Recent Advances in Intrusion Detection, LNCS 2820, Springer Berlin/Heidelberg (2003), 73-93.

[13] N. Joukov and T. Chiueh. Internet worms as internet-wide threat. Technical Report. Depart-ment of Computer Science, Stony Brook University, 2003.

[14] Jung, J., Krishnamurthy, B. and Rabinovich, M. (2002). Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In Proceedings of ACM WWW.

[15] S. Sarvotham, R. Riedi, R. Baraniuk, "Connectionlevel analysis and modeling of network traffic," in ACM SIGCOMM Internet Measurement Workshop, San Francisco, CA, USA, November 2001, pp. 99–103.

[16] W. E. Leland, M. S. Taqqu, W. Willinger, e D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). IEEE/ACM Transactions on Networking, 2(1):1-15, 1994.

[17] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo, "Discovery of Frequent Episodes in Event Sequences", Data Mining and Knowledge Discovery l(3): 259-289 (1997)

[18] M. J. Atallah, R. Gwadera, andW. Szpankowski. Detection of significant sets of episodes in event sequences. In Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), pages 3–10, Brighton, UK, 01-04 November 2004.

[19] S. Harms, S. Goddard, S. E. Reichenbach, W. J. Waltman, and T. Tadesse. Data mining in a geospatial decision support system for drought risk management. In Proceedings of the 2001 National Conference on Digital Government Research, pages 9--16, Los Angelos, California, USA, May 2001b.

[20] Toma, T., Abu-Hanna, A., Bosman, RJ. Predicting mortality in the intensive care using episodes, (2005) Lecture Notes in Computer Science, 3561 PART I, Pages 447-458.

[21] Srivatsan Laxman, P. Sastry, K. Unnikrishnan, "Discovering Frequent Generalized Episodes When Events Persist for Different Durations," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 9, pp. 1188-1201, June 2007, doi:10.1109/TKDE.2007.1055

[22] Sadoddin, R. and Ghorbani, A. Alert correlation survey: framework and techniques. In Proceedings of the 2006 international Conference on Privacy, Security and Trust - PST '06. (Markham, Ontario, Canada, October 30 - November 01, 2006). vol. 380. ACM, New York, NY, 1-10. DOI= http://doi.acm.org/10.1145/1501434.1501479

[23] MITRE. Common Vulnerabilities and Exposures (CVE). 2009. http://cve.mitre.org.

[24] Ramasubramanian, P. et al., "Quickprop Neural Network Ensemble Forecasting Framework For A Database Intrusion Prediction System," Neural Information Processing—Letters and Reviews, Oct. 2004, pp. 9-16, vol. 5, No. 1.

[25] Valdes, A. and Skinner, K. Probabilistc Alert Correlation. In Proceedings of the Recent Advances in Intrusion Detection (RAID). Davis, CA. 2001.

[26] Julish, K. Mining Alarm Clusters to Improve Alarm Handling Efficiency. Proceedings of the 17th Annual Conference on Computer Security Applications. New Orleans, LA.

[27] Cuppens, F. Managing alerts in a multi-intrusion detection environment. Proceedings of the 17th Annual Conference on Computer Security Applications (ACSAC); 2001. pp. 22-31.

[28] Yiu-Ming Cheung, Wai-Man Leung, and Lei Xu, Adaptive rival penalized competitive learning and combined linear predictor model for financial forecast and investment, International Journal of Neural Systems, Vol. 8, Nos. 5/6, October/December 1997.

[29] Pradeep Kannadiga, Mohammad Zulkernine, Anwar Haque: E-NIPS: An Event-Based Network Intrusion Prediction System. ISC 2007: 37-52

[30] An agent-based Bayesian forecasting model for enhanced network security (with J. Pikoulas, W. Buchanan and M. Manion). In ECBS '01: Proceedings of the 8th IEEE International Conference on Engineering of Computer-Based Systems, IEEE Computer Society, 2001, 247-254.

[31] Hu P., Heywood M.I., Predicting Intrusions with Local Linear Models, IEEE International Joint Conference on Neural Networks, July 20th-24th 2003.

[32] Debar, H., Curry, D., and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)", RFC 4765, March 2007.

[33] Debar, H. and Wespi, A. Aggregation and Correlation of Intrusion-Detection Alerts. In Proceedings of the Recent Advances in Intrusion Detection (RAID). Davis, CA. 2001.

[34] Zhu, B. and Ghorbani, A. A. Alert correlation for extracting attacks strategies. International Journal of Network Security, 3(2):244-258, 2006.

[35] Morin, B. and Debar, H. Correlation of Intrusion Symptoms: an Application of Chronicles. Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID); 2003. pp. 94-112

[36] Dousson, C. Suivi d'évolutions et reconnaissance de chroniques. PhD Thesis. 1994.

[37] Cuppens, F. and Ortalo, r. LAMBDA: A language to model a database for detection attacks. Proceedings of the 3th International Symposium on Recent Advances in Intrusion Detection (RAID); 2000. pp. 197-216.

[39] Vankamamidi, R. ASL: A specification language for intrusion detection and network monitoring. Master´s Thesis. Iowa State University, 1998.

[40] Templeton, S. and Levitt, L. A requires/provides model for computer attacks. Proceedings of new security paradigms workshop; 2000. pp. 31-38.

[41] Totel, E., Vivinis, B., and Mé, L. A language driven ids for event and alert correlation. SEC, pp. 209-224. 2004.

[42] Dain, O. and Cunningham, R. Fusing a heterogeneous alert stream into scenarios. Proceedings of the 2001 ACM workshop on data mining for security applications; 2001. pp. 1-13.

[43] Ning, P., Cui, Y., and Reeves, D. Constructing attack scenarios through correlation of intrusion alerts. Proceedings of the ACM Conference of Computer and Communications Security. pp. 245-254. Washington DC. 2002.

[44] Qui, X. and Le, W. Statistical Causality of INFOSEC Alert Data. Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID); 2003.

[45] Qui, X. A Probabilistic-Based Framework for INFOSEC Alert Correlation. PhD Thesis, Georgia Institute of Technology, 2005.

[46] Almgren, M., Lindqvist, U., and Jonsson, E. A multi-sensor model to improve automated attack detection. Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID); 2008. pp 291-310.