Mineração de Exceções Aplicada aos Sistemas para Detecção de Intrusões

Eduardo Corrêa Gonçalves, Célio V. N. Albuquerque, Alexandre Plastino

Instituto de Computação, Universidade Federal Fluminense – UFF
Rua Passo da Pátria, 156, Bloco E, 3º andar – 24.210-240 – Niterói – RJ - Brasil
{egoncalves,celio,plastino}@ic.uff.br

Resumo

Os sistemas para a detecção de intrusões em redes de computadores freqüentemente utilizam modelos baseados em regras para o reconhecimento de padrões suspeitos nos dados do tráfego. Este trabalho apresenta uma técnica baseada na mineração de exceções que pode ser utilizada para aumentar a eficiência deste tipo de sistema. As exceções representam regras de associação que tornam-se extremamente fortes (exceções positivas) ou extremamente fracas (exceções negativas) em subconjuntos de uma base de dados que satisfazem condições específicas sobre atributos selecionados. São apresentados os resultados obtidos a partir da aplicação desta técnica sobre a base KDDCup99 que registra informações sobre conexões de rede.
Palavras-chave: Mineração de Dados, Redes de Computadores, Sistemas para Detecção de Intrusões.

Abstract

Network intrusion detection systems often employ rule-based models to recognize suspect patterns in traffic data. In this work, we present a technique based on exception mining that can be employed to improve the efficiency of this kind of system. Exceptions represent association rules that become either exceptionally weaker (negative exceptions) or exceptionally stronger (positive exceptions) in some subsets of the database, which satisfy specific conditions over selected attributes. The efficacy of this technique is demonstrated using the KDDCup99 network connection data set.
Key-words: Data Mining, Computer Networks, Intrusion Detection Systems.

1 Introdução

A crescente popularização das aplicações para o ambiente *Web* (e para redes de computadores em geral) trouxe a necessidade da elaboração de novos métodos para a proteção deste tipo de sistema. Dentre eles, encontram-se os mecanismos para a **detecção de intrusões**. A palavra intrusão é definida em [16] do seguinte modo: "ação de introduzir-se com astúcia ou violência; usurpação; posse violenta; intromissão". De maneira análoga, no domínio da Ciência da Computação, uma intrusão representa um conjunto de ações que visa comprometer a integridade, a confiabilidade ou a disponibilidade de um recurso de um sistema de informações [11].

As técnicas para a prevenção de intrusões - como a utilização de senhas, a criptografia de dados e o controle de acesso a arquivos - têm sido tradicionalmente empregadas pelos programadores. Nos últimos anos, métodos para a detecção de intrusões passaram a ser utilizados em conjunto com estas técnicas de prevenção, com o objetivo de servir como uma segunda linha de defesa para os sistemas de informação. Na detecção de intrusões, as atividades do sistema são monitoradas para que uma tentativa de intrusão possa ser reconhecida e, então, bloqueada. A necessidade da utilização de mecanismos para a detecção de intrusões é justificada por dois motivos principais, identificados em [13]:

- Os sistemas operacionais e os demais tipos de software estão se tornando cada vez mais complexos e, com isso, acabam por apresentar um maior número de vulnerabilidades que podem ser exploradas por programas maliciosos.
- Muitas aplicações antigas, que estão em uso atualmente na *Internet* e tornaram-se extremamente populares, foram desenvolvidas originalmente com propósito meramente científico e sequer oferecem mecanismos para a prevenção de intrusões.

Tipicamente, os sistemas para a detecção de intrusões (*intrusion detection systems* – IDS) em redes de computadores funcionam da forma descrita em [4] e reproduzida a seguir. Em primeiro lugar, é necessário coletar e armazenar continuamente dados sobre o tráfego da rede. Utiliza-se então um *software* para reconhecimento de padrões (denominado sensor), que é capaz de monitorar este tráfego e disparar alarmes caso algum padrão suspeito seja reconhecido em meio aos dados. Quando um alarme é disparado, os analistas de

segurança examinam o padrão suspeito e concluem se está ocorrendo algum tipo de intrusão, para que a ação necessária possa ser tomada (como, por exemplo, bloquear temporariamente os dados vindos do ISP associado ao tráfego suspeito). Em grande parte dos modelos de implementação deste tipo de sistema, os sensores consultam **bancos de dados de regras** para que as intrusões possam ser identificadas. Neste caso, cada regra do banco possui uma estrutura capaz de descrever as características de um determinado tipo de intrusão. Este modelo de implementação é similar ao utilizado pelos softwares antivírus modernos. O banco de dados de regras de um IDS precisa estar sempre atualizado para que a segurança da rede esteja garantida. O consagrado sistema *Snort* [18,19], por exemplo, disponibiliza em seu site atualizações constantes de bibliotecas de regras para a detecção de intrusões. Além disso, o site possui um formulário para que colaboradores possam enviar regras que descrevem padrões de intrusões. Cada regra enviada é avaliada por especialistas, podendo, posteriormente, ser incorporada ao banco de dados do *Snort*.

Técnicas de **mineração de dados** podem ser utilizadas para auxiliar na construção dos dicionários de regras utilizados pelos sistemas para detecção de intrusões. A mineração de dados é um processo realizado por meio de estratégias automatizadas para a análise de grandes bases de dados, procurando extrair das mesmas informações que estejam implícitas, que sejam previamente desconhecidas e potencialmente úteis. Dentre os diferentes tipos de informação que podem ser minerados em bases de dados, tais como clusters de dados, padrões em séries temporais e hierarquias de classificação, as **regras de associação** destacam-se por sua simplicidade e aplicabilidade. Detalhes sobre as diferentes técnicas e tarefas de mineração de dados podem ser encontrados em [5,6,8,17].

As **regras de associação multidimensionais**, definidas em [8], representam padrões de relacionamento extraídos a partir de repositórios como *data warehouses* e bancos de dados relacionais. Um exemplo de regra multidimensional, minerada a partir de uma base de dados que registra informações sobre cerca de 500.000 conexões de rede [9], é dado por:

$$\text{R1: } (Service = \text{“telnet”}) \wedge (FailedLogins \geq 1) \Rightarrow (Intrusion = \text{“yes”})$$

Esta regra indica que as conexões nas quais o serviço de rede utilizado foi o *telnet* e que apresentaram uma ou mais operações de *login* rejeitadas, tendem a ser tentativas de intrusões.

Um dos grandes problemas associados aos sistemas para a detecção de intrusões que utilizam modelos baseados em regras consiste no fato de eles dispararem uma grande quantidade de alarmes falsos, tornando difícil o processo de análise por parte dos especialistas em segurança. Para tratar este problema, este trabalho propõe o emprego de uma técnica para **mineração de exceções**, definida em [7]. Nesta abordagem, um especialista tem a possibilidade de explorar um conjunto de regras de associação, descobrindo o quanto a força (medida de interesse) destas regras afasta-se de seus valores médios em diferentes subconjuntos da base de dados. Exceções (negativas ou positivas) são mineradas quando estes desvios apresentam valores significativos. A técnica para mineração de exceções é capaz de aumentar a qualidade das bases de dados de regras que são utilizadas pelos sistemas para a detecção de intrusões.

O restante do trabalho está dividido da seguinte forma. A Seção 2 apresenta alguns conceitos preliminares sobre regras de associação multidimensionais. A seguir, a Seção 3 discute a aplicação das regras de associação nos modelos de sistemas de detecção de intrusões. A Seção 4 apresenta a contribuição principal deste trabalho: a aplicação da mineração de exceções em bases de dados de tráfego de rede. Na Seção 5, apresentam-se alguns resultados experimentais obtidos a partir da aplicação da técnica para a mineração de exceções sobre a base de dados de [9]. Finalmente, na Seção 6 são apresentadas as conclusões e comentários finais.

2 Regras de Associação

As regras de associação multidimensionais [8] representam padrões de relacionamento extraídos a partir de repositórios como *data warehouses* e bancos de dados relacionais. Uma regra de associação multidimensional é uma expressão da forma $A \Rightarrow B$, onde A e B representam conjuntos de condições sobre diferentes atributos de uma relação. A é denominado antecedente da regra de associação e B é denominado conseqüente. O suporte de um conjunto de condições Z, $Sup(Z)$, representa a porcentagem de tuplas da relação que satisfazem a todas as condições de Z. O suporte de uma regra $A \Rightarrow B$, $Sup(A \Rightarrow B)$, corresponde à probabilidade de uma tupla satisfazer a todas as condições em $A \cup B$. Já a confiança de $A \Rightarrow B$, $Conf(A \Rightarrow B)$, corresponde à probabilidade de uma tupla satisfazer B dado que ela satisfaz A, ou seja, $Conf(A \Rightarrow B) = Sup(A \Rightarrow B) \div Sup(A)$. O problema da mineração de regras de associação consiste em gerar todas as regras que possuam suporte e confiança maiores ou iguais, respectivamente, a um suporte mínimo (*SupMin*) e uma confiança mínima (*ConfMin*) [1].

Tipicamente, o processo de mineração de regras de associação é dividido em duas etapas:

1. Determinar todos os conjuntos de condições que possuem suporte maior ou igual a $SupMin$. Estes conjuntos são chamados de conjuntos freqüentes.

2. Para cada conjunto freqüente encontrado na Etapa 1, gerar todas as regras de associação que possuem confiança maior ou igual a *ConfMin*.

Observe que neste modelo, para que uma regra seja considerada forte, contendo informação

interessante, é necessário que ela apresente bons valores de suporte (Etapa 1) e confiança (Etapa 2). A decisão sobre quais regras devem ser mantidas e quais deverão ser descartadas durante o processo de mineração é baseada nos valores destes dois índices. Isto significa que o suporte e a confiança atuam como **medidas de interesse** no processo de mineração de regras de associação.

3 Regras de Associação nos Sistemas para Detecção de Intrusões

Esta seção discute a aplicação das regras de associação multidimensionais em algumas abordagens propostas para a construção das bases de dados de sistemas para a detecção de intrusões. No entanto, antes de iniciar esta discussão, serão apresentados alguns conceitos preliminares sobre sistemas para detecção de intrusões.

Os sistemas para a detecção de intrusões podem utilizar duas abordagens na execução de sua tarefa: **detecção de abusos** (*misuse detection*) e **detecção de anomalias** (*anomaly detection*). Em ambos os casos, os sensores precisam consultar uma base de dados que armazene padrões descobertos em dados de tráfego de rede. A abordagem baseada na detecção de abusos consiste em manter um banco de dados com as características dos ataques conhecidos. Os dados monitorados pelos sensores são avaliados sobre esta base de dados e os alarmes são disparados quando ocorre alguma ação que possua padrões idênticos aos de uma intrusão registrada na base. Já a abordagem baseada na detecção de anomalias funciona de maneira exatamente oposta. Esta abordagem requer uma base de dados que registre o modelo normal de comportamento do sistema. Qualquer ação que desvie significativamente deste modelo normal de comportamento é considerada intrusiva. A detecção de anomalias possui a vantagem de ser capaz de identificar ataques desconhecidos. No entanto, pode apresentar problemas quando a operação normal do sistema é irregular, o que ocasiona o disparo de muitos alarmes falsos [14,15]. Por sua vez, a abordagem baseada na detecção de abusos não consegue identificar ataques desconhecidos. Porém, é mais eficiente para a detecção de ataques conhecidos e costuma disparar um número menor de alarmes falsos. Alguns sistemas para a detecção de intrusões são implementados utilizando as duas abordagens em conjunto [19].

A seguir será apresentado um exemplo de como a mineração de regras de associação pode ser aplicada sobre uma base de dados de tráfego da rede para que as características de uma intrusão possam ser descobertas. Considere a base de dados hipotética apresentada na Tabela 1. Esta base de dados possui estrutura similar à base disponibilizada em [9], que é comumente utilizada como *benchmark* pelos sistemas de detecção de intrusões.

Na base de dados da Tabela 1, os dados de conexões normais encontram-se misturados aos dados de tentativas de intrusão. Cada registro representa as informações de uma conexão de rede, identificadas através do campo **Id**. As conexões são caracterizadas através de outros quatro atributos. O atributo **Serviço** registra o serviço de rede utilizado durante a conexão (neste exemplo, os serviços limitam-se a dois tipos: "Telnet" e "FTP"). O atributo **Logins Rejeitados** é utilizado para indicar se houve alguma tentativa de *login* rejeitada durante o tempo de duração da conexão. O atributo **Flag** corresponde ao status da conexão TCP. Na base hipotética da Tabela 1, os valores possíveis para o campo **Flag** são os seguintes: "SF" (para conexões que realizaram um *handshake* SYN normal e foram terminadas também de maneira normal, com um FIN), "RSTO" (para conexões que foram terminadas através de um pacote *reset* enviado pelo *host* cliente para o *host* servidor) e "S0" (conexões onde houve um SYN inicial, mas não houve continuação do processo de *handshake*). Outros valores possíveis para o atributo **Flag** são apresentados em [2]. Por fim, o atributo **Intrusão** é utilizado para rotular cada conexão como uma conexão normal ou como uma tentativa de intrusão.

Tabela 1 – Base de dados de conexões de rede

Id	Serviço	Logins Rejeitados	Flag	Intrusão
1	Telnet	SIM	RSTO	SIM
2	FTP	SIM	SF	SIM
3	Telnet	SIM	SF	NÃO
4	Telnet	SIM	RSTO	SIM
5	Telnet	SIM	RSTO	SIM
6	FTP	NÃO	SF	SIM
7	Telnet	SIM	S0	SIM
8	Telnet	SIM	SF	NÃO
9	FTP	NÃO	SF	NÃO
10	Telnet	NÃO	RSTO	NÃO

Observe que a base hipotética apresentada na Tabela 1 é um exemplo de base pré-processada (preparada) por um analista. Existe um atributo que é utilizado para classificar uma conexão como normal ou como uma tentativa de intrusão. Em [3], é destacado que as organizações raramente possuem registros de ataques rotulados, uma vez que o processo de construção de uma base com estas características consome enorme tempo e esforço. Na prática, são geralmente os desenvolvedores dos sistemas para detecção de intrusões que preparam este tipo de base, com os objetivos de gerar bancos de dados de regras e testar os algoritmos utilizados pelos sensores.

Considere que o desenvolvedor de um IDS esteja interessado em aplicar um algoritmo de regras de associação sobre a base de dados apresentada na Tabela 1. Suponha que como entrada foram especificados os parâmetros *SupMin* = 40% e *ConfMin* = 60%. Entre as várias regras que o algoritmo para mineração de regras de associação seria capaz de gerar, encontra-se a seguinte: "quando o serviço de rede utilizado foi o *telnet* e ocorreu alguma tentativa de login que foi rejeitada, a conexão costuma ser uma tentativa de intrusão". Esta

regra, que pode ser representada por *(Serviço = "Telnet") ∧ (Logins Rejeitados = "SIM") ⇒ (Intrusão = "SIM")*, possui suporte e confiança iguais a 40% e 66,67%, respectivamente. Este é um exemplo de padrão descoberto através de um algoritmo de mineração de regras de associação, que poderia ser utilizado em um banco de dados de regras de um IDS.

A aplicação das regras de associação em sistemas para a detecção de intrusão foi proposta pela primeira vez em [11]. O trabalho descreve um modelo para a implementação de um IDS inteiramente baseado em técnicas de mineração de dados. Neste trabalho, grupos formados por uma ou mais regras de associação são utilizados para descrever o comportamento normal de uma rede. Por sua vez, a proposta de [3] utiliza dicionários contendo conjuntos de condições freqüentes (gerados durante a primeira fase do processo de mineração de regras de associação, conforme foi descrito na Seção 2) para definir a atividade normal da rede e, deste modo, tornar possível a descoberta de anomalias.

4 Exceções

O modelo tradicional para mineração de regras de associação possui alguns problemas, apontados em [20]. A quantidade de regras geradas costuma ser muito volumosa, dificultando o processo de análise por parte dos usuários. No entanto, o número de regras que são, de fato, úteis costuma bem menor. Freqüentemente as regras úteis possuem suporte baixo e, com isso, podem não ser descobertas através de métodos convencionais para a mineração de regras de associação, uma vez que a medida de suporte é utilizada como entrada nestes processos. A **mineração de exceções** [7,20] é uma técnica que pode ser utilizada para lidar com este problema. Estas estratégias são direcionadas para a descoberta de regras úteis, inesperadas ou com suporte e confiança com valores baixos.

Esta seção apresenta a contribuição central deste trabalho: a aplicação da técnica para a mineração de exceções definida em [7] com o objetivo de melhorar a qualidade do banco de dados de regras de um IDS. Para ilustrar os conceitos apresentados ao longo da seção, será utilizado um exemplo obtido a partir da base de dados hipotética apresentada na Tabela 1.

4.1 Exceções Negativas

As exceções negativas representam regras de associação multidimensionais que tornam-se mais fracas em determinados subconjuntos da base de dados. Estas exceções são mineradas se possuírem o suporte ou a confiança significativamente inferior a uma determinada expectativa. Com o objetivo de ilustrar a proposta, considere a base de dados de conexões, apresentada na Tabela 1. Na seção anterior, foi demonstrado que *(Serviço = "Telnet") ∧ (Logins Rejeitados = "SIM") ⇒ (Intrusão = "SIM")* é um exemplo de regra de associação que pode ser extraída desta base. Entretanto, observando a base de dados é possível notar que **nenhuma** das conexões onde serviço de rede utilizado foi o *telnet* e ocorreu alguma tentativa de *login* rejeitada, representou uma tentativa de intrusão **quando o flag da conexão possui o valor "SF"** (observe os registros com valores de Id iguais a 3 e 8). Este padrão negativo pode ser representado da maneira apresentada na Figura 1.

$$(Serviço = \text{"Telnet"}) \wedge (Logins\ Rejeitados = \text{"SIM"}) \overset{-c-s}{\Rightarrow} (Intrusão = \text{"SIM"})\ [(Flag = \text{"SF"})]$$

Figura 1 – Exceção Negativa

O símbolo "$\overset{-c-s}{\Rightarrow}$" é utilizado para indicar que o suporte e a confiança da regra de associação *(Serviço = "Telnet") ∧ (Logins Rejeitados = "SIM") ⇒ (Intrusão = "SIM")* é significativamente inferior ao esperado no subconjunto da base de dados definido pelas conexões que realizaram um *handshake* SYN normal e foram terminadas também de maneira normal, com um FIN (condição *Flag = "SF"*). A definição formal de exceção negativa é apresentada a seguir:

Definição 1 *(Exceção Negativa)*
*Seja R: $A \Rightarrow B$ uma regra de associação multidimensional obtida a partir de uma base de dados. Seja Z um conjunto de condições definidas sobre atributos distintos desta base, onde $A \cap B \cap Z = \varnothing$. Z é denominado **conjunto de prova**. Uma exceção negativa associada à regra R é uma expressão com uma das seguintes formas:*

$A \overset{-s}{\Rightarrow} B\ [Z]$ - Exceção negativa com relação ao suporte.

$A \overset{-c}{\Rightarrow} B\ [Z]$ - Exceção negativa com relação à confiança.

$A \overset{-c-s}{\Rightarrow} B\ [Z]$ - Exceção negativa com relação ao suporte e à confiança.

Uma exceção negativa possui o objetivo de indicar o quanto a presença de um conjunto de prova pode enfraquecer uma regra multidimensional que é, originalmente, forte. Uma exceção negativa é minerada a partir de uma **exceção candidata**. As exceções candidatas são formadas através da combinação de uma regra de associação multidimensional $A \Rightarrow B$ com um conjunto de prova Z e são representadas da seguinte forma: $A \Rightarrow B\ [Z]$. Uma exceção negativa é minerada apenas quando a exceção candidata possui suporte ou confiança abaixo de uma determinada expectativa. Esta expectativa é calculada em função do suporte da regra original $A \Rightarrow B$ e do suporte do conjunto de prova Z.

Definição 2 *(Suporte Esperado de uma Exceção Candidata)*
Seja C: $A \Rightarrow B\ [Z]$ uma exceção candidata. O suporte real de C é obtido por $(A \cup B \cup Z)$. Já o suporte esperado de C, denotado por SupEsp(C) é computado por: SupEsp(C) = ***$Sup(A \Rightarrow B)\ X\ Sup(Z)$.***

Definição 3 *(Confiança Esperada de uma Exceção Candidata)*
Seja C: $A \Rightarrow B\ [Z]$ uma exceção candidata. A confiança real de C é obtida por $Sup(A \Rightarrow B\ [Z]) \div Sup(A \cup Z)$. Já a confiança esperada de C, denotada por ConfEsp(C), é computada por:
$ConfEsp(C) = SupEsp(C) \div SupEsp(A \cup Z) =$
$(Sup(A \Rightarrow B)\ X\ Sup(Z)) \div (Sup(A)\ X\ Sup(Z)) =$
$Sup(A \Rightarrow B) \div Sup(A) =$ ***$Conf(A \Rightarrow B)$***.

Para melhor ilustrar os conceitos de suporte esperado e confiança esperada, será utilizado o exemplo da exceção negativa apresentada na Figura 1 (minerada a partir da base de dados hipotética da Tabela 1). Neste caso, o processo que levou à mineração desta exceção negativa começou com a concatenação da regra de associação convencional R_1 : *(Serviço = "Telnet")* $\wedge$ *(Logins Rejeitados = "SIM")* $\Rightarrow$ *(Intrusão = "SIM")* e do conjunto de prova $Z_1 = \{(Flag = "SF")\}$. Gerou-se então a exceção candidata C_1 : *(Serviço = "Telnet")* $\wedge$ *(Logins Rejeitados = "SIM")* $\Rightarrow$ *(Intrusão = "SIM") [(Flag = "SF")]*. Para que possa ser definido se esta exceção candidata representa, de fato, uma exceção negativa interessante, é necessário realizar os seguintes passos:

1. Obter o suporte real de $C_1 = Sup(C_1) = Sup(R_1 \cup Z_1) = 0\%$.

2. Obter o suporte esperado de $C_1 = SupEsp(C_1) = Sup(R_1)\ X\ Sup(Z_1) = 40\%\ X\ 50\% = 20\%$.

3. Obter a confiança real de $C_1 = Conf(C_1) = Sup(R_1 \cup Z_1) \div Sup(A \cup Z_1) = 0\% \div 20\% = 0\%$.

4. Obter a confiança esperada de $C_1 = ConfEsp(C_1) = Conf(R_1) = 66{,}67\%$.

Observe que o suporte e a confiança reais da exceção candidata possuem valores significativamente inferiores aos valores esperados. Por esta razão, foi possível minerar a exceção negativa apresentada na Figura 1.

A **Medida da Força para Exceções Negativas** (MF-), definida em [7], pode ser utilizada para medir o desvio do valor do suporte real em relação ao esperado. Esta medida é computada da seguinte maneira:

$$MF\text{-}(A \Rightarrow B[Z]) = 1 - Sup(A \Rightarrow B[Z]) \div SupEsp(A \Rightarrow B[Z])\ .$$

Note que o valor da medida cresce quando o suporte real da exceção candidata é inferior e se distancia do valor esperado. Se o suporte real é superior ou igual ao esperado, a exceção negativa não possui nenhum interesse.

Já a **Medida do Desvio da Confiança para Exceções Negativas** (DC-), definida em [7] pode ser utilizada para medir o desvio do valor da confiança real em relação ao valor esperado. O desvio da confiança é computado da seguinte forma:

$$DC\text{-}(A \Rightarrow B[Z]) = 1 - Conf(A \Rightarrow B[Z]) \div ConfEsp(A \Rightarrow B[Z]).$$

Mais uma vez, o valor da medida cresce quando a confiança real da exceção candidata é inferior e se distancia do valor esperado.

4.2 Exceções Positivas

Também é possível extrair exceções positivas a partir de bases de dados. Estas exceções são mineradas quando possuem valor de suporte ou confiança significativamente superior a um valor esperado. A idéia da exceção positiva é a de avaliar o quanto a presença de um conjunto de prova é capaz de fortalecer uma regra de associação que pode, originalmente, ser fraca. Assim como ocorre com as exceções negativas, as exceções positivas também são obtidas a partir de exceções candidatas. A definição formal de exceção positiva é apresentada a seguir.

Definição 4 *(Exceção Positiva)*
Seja R: $A \Rightarrow B$ uma regra de associação multidimensional obtida a partir de uma base de dados. Seja Z um conjunto de de prova. Uma exceção positiva associada à regra R é uma expressão com uma das seguintes formas:

$A \overset{+s}{\Rightarrow} B\ [Z]$ *- Exceção positiva com relação ao suporte.*

$A \overset{+c}{\Rightarrow} B\ [Z]$ *- Exceção positiva com relação à confiança.*

$A \overset{+c+s}{\Rightarrow} B[Z]$ *- Exceção positiva com relação ao suporte e à confiança.*

O método de mineração de exceções positivas é similar ao da mineração de exceções negativas, assim como as medidas de interesse utilizadas para avaliar este tipo de padrão. Estas medidas foram introduzidas em [7]. A **Medida da Força para Exceções Positivas** (MF+) avalia o quanto o suporte real de uma exceção candidata é superior ao suporte esperado. De maneira análoga, **Medida do Desvio de Confiança para Exceções Positivas** (DC+) avalia o quanto a confiança real de uma exceção candidata é maior do que o valor que era esperado. Estas medidas são computadas da forma descrita a seguir.

$$MF^{+}(A \Rightarrow B[Z]) = 1 - SupEsp(A \Rightarrow B[Z]) \div Sup(A \Rightarrow B[Z])\ .$$

$$DC^{+}(A \Rightarrow B[Z]) = 1 - ConfEsp(A \Rightarrow B[Z]) \div Conf(A \Rightarrow B[Z]).$$

Note que desta vez o suporte esperado e a confiança esperada encontram-se no numerador de cada

uma das fórmulas. A mineração de exceções positivas é capaz de revelar padrões interessantes que não seriam minerados por um algoritmo para a mineração de regras de associação convencionais, como por exemplo regras com suporte muito baixo (inferior ao suporte mínimo), mas com uma confiança alta.

4.3 Algoritmo

A Figura 2 apresenta um procedimento para a mineração de exceções negativas e positivas com relação à medida de confiança. Este algoritmo foi implementado na linguagem C++ e aplicado sobre a base de dados de conexões de rede [9]. Os resultados obtidos são apresentados na seção 5. O algoritmo para mineração de exceções com relação à medida de confiança utiliza os seguintes parâmetros como entrada:

1. *R* – um conjunto de regras de associação multidimensionais.
2. *A* – um conjunto de atributos que formarão os conjuntos de prova.
3. *Supmin* ≥ *0*, *DCmin* ≥ *0* – valores mínimos para s medidas de interesse.

Como saída são apresentados os seguintes resultados:

1. *EN* – um conjunto de exceções negativas.
2. *EP* – um conjunto de exceções positivas.

```
procedimento minerarExceções

1.  CP = gerar todos os conjuntos de prova a
    partir de A;
2.  Candidatas = ∅;
3.  ConjCondições = ∅;
4.  para cada regra A ⇒ B em R faça
5.    para cada conjunto de prova Z em CP faça
6.      Candidatas = Candidatas ∪ (A ⇒ B [Z]);
7.      X' = {{A}, {B}, {Z}, {A,B}, {A,Z},
                {B,Z}, {A,B,Z}};
8.      ConjCondições = ConjCondições ∪ X';
9.    fim para;
10. fim para;

11. obter suporte dos conjuntos de
    ConjCondições;

12. EN = ∅;
13. EP = ∅;
14. para cada regra candidata C':A ⇒ B [Z] em
    Candidatas faça

15.   se Sup(A ∪ Z) ≥ Supmin e
         Sup(B ∪ Z) ≥ Supmin então

      // testa se é exceção negativa
16.     se DC-(C') ≥ Dcmin então
                          -c
            EN = EN ∪ A⇒ B [Z];
17.     fim se;

      // testa se é exceção positiva
18.     se DC+(C') ≥ Dcmin então
                          +c
            EP = EP ∪ A⇒ B [Z];
19.     fim se;
20.   fim se;
21. fim para;
```

Figura 2 – Algoritmo para minerar exceções com relação à medida de confiança

O algoritmo está dividido em quatro fases. A fase 1 (linha 1) determina todos os conjuntos de prova que serão utilizados na geração das regras candidatas, a partir dos atributos especificados pelo usuário no conjunto *A*.

A fase 2 (linhas 2-10) apresenta a rotina utilizada para a geração das exceções candidatas. O conjunto *Candidatas* e a estrutura *ConjCondições* são inicializados nas linhas 2 e 3, respectivamente. O primeiro conjunto é utilizado para armazenar todas as exceções candidatas, que são geradas através da combinação de cada conjunto de prova em *CP* com cada regra de associação multidimensional em *R* (linha 6). J´a a estrutura *ConjCondições* é utilizada para auxiliar no cálculo dos valor da medida do desvio da confiança das exceções. Para que a medida do desvio da confiança de uma exceção candidata $A \Rightarrow B\ [Z]$ possa ser calculada, é necessário que os valores de suporte dos conjuntos: {A}, {B}, {Z}, {A,B}, {A,Z}, {B,Z} e {A,B,Z} sejam conhecidos. A estrutura *ConjCondições* é utilizada para manter contadores para todos estes conjuntos (linhas 7-8).

Na fase 3 (linha 11), a base de dados é inteiramente varrida, com o objetivo de contar o suporte de todos os conjuntos armazenados na *ConjCondições* . É importante observar que basta percorrer a base uma única vez, visto que os conjuntos que precisam ter os valores de suporte contabilizados já são conhecidos.

Finalmente, na fase 4 (linhas 12-21), as exceções negativas e positivas com relação à medida da confiança são geradas. Cada uma das exceções candidatas é analisada da seguinte maneira. Primeiro são verificadas as restrições com relação ao suporte mínimo (parâmetro *Supmin*), com o objetivo de garantir a significância estatística dos padrões minerados (linha 15). Em seguida computa-se o valor da medida DC- para a exceção candidata que está sendo correntemente analisada e verifica-se se o valor obtido é superior ao especificado no parâmetro *Dcmin* (linha 16). Para a realização deste cálculo, os valores de suporte dos conjuntos de condições armazenados na ´ *ConjCondições* devem ser consultados (valores estes obtidos na terceira fase do procedimento). Se o resultado apresentado pela medida DC- for superior ao valor mínimo especificado pelo usuário em *DCmin*, uma **exceção negativa** com relação à medida da confiança pode ser minerada e armazenada no conjunto *EN*. Caso contrário, a medida DC+ é computada (linha 18). Se o valor de DC+ for superior a $Dcmin$, uma **exceção positiva** com relação à medida da confiança pode ser minerada e armazenada no conjunto *EP*.

O procedimento para mineração de exceções negativas e positivas aqui apresentado é um exemplo de procedimento cooperativo, pois o foco de todo o processo de mineração é especificado pelo usuário. Ao

usuário é dada a possibilidade de escolher um conjunto de regras transacionais cuja variação da força ele está interessado em investigar em diferentes subconjuntos da base de dados. O usuário também é responsável por especificar estes subconjuntos da base de dados, selecionando os atributos que são utilizados para gerar os conjuntos de prova. O procedimento de mineração de exceções realiza um esforço proporcional às escolhas realizadas pelo usuário, especialmente concentrado na criação da *ConjCondições* e na contagem do suporte dos conjuntos de condições que nela estão armazenados. Apenas os atributos que compõem as regras candidatas são relevantes e, com isso, um número muito menor de conjuntos necessita ter o suporte contabilizado.

5 Resultados

A técnica para a mineração de exceções foi aplicada sobre a base de dados disponível em [9]. Esta base de dados foi construída como parte de um projeto de pesquisa no campo da detecção de intrusões, realizado pelo MIT [12]. O método utilizado para sua construção foi o seguinte. Durante 9 semanas coletou-se dados do tráfego de uma rede no nível *tcpdump* [21] (dados do cabeçalho dos pacotes). Esta rede foi montada especialmente para o projeto do MIT. Os responsáveis pelo projeto construíram programas de computador que simulavam diversos tipos de intrusão. Pacotes foram enviados para a rede a partir destes programas. Desta forma, foi possível gerar um arquivo de dados tcpdump que continha registros de pacotes normais e registros dos pacotes gerados pelos programas que simulavam as intrusões.

A base de dados do MIT foi, posteriormente, preparada para ser utilizada na *KDDCup99* [9], uma competição que avaliou algoritmos de mineração de dados para sistemas de detecção de intrusões. Esta preparação, descrita de maneira breve em [11], consistiu em transformar os dados do nível *tcpdump* para o nível de conexões. A base de [9] é disponibilizada em duas versões: completa (que registra 5 milhões de conexões) e simplificada (uma amostra contendo 500.000 conexões selecionadas de maneira aleatória). As duas bases contém um total de 41 atributos. No teste realizado neste trabalho, foi utilizada a base simplificada e foram selecionados apenas os atributos apresentados na Tabela 2. Houve a necessidade de realizar algumas transformações para que a técnica da mineração de exceções pudesse ser aplicada. Os atributos numéricos ***Duration***, ***SrcBytes***, ***DstBytes***, ***FailedLogins***, ***NumRoot***, ***Count*** e ***SrvCount*** tiveram que ser discretizados, pois o algoritmo para mineração de exceções trabalha apenas com dados categóricos.

O atributo ***Class*** merece uma explicação adicional. Este atributo indica se uma conexão é normal ou se representa algum tipo de intrusão, como, por exemplo, *smurf* e *syn flood*, que correspondem a duas intrusões do tipo negação de serviço. As intrusões do tipo negação de serviço consomem recursos dos servidores e roteadores para que usuários legítimos sejam impedidos de ter acesso a um determinado serviço [10]. A intrusão *smurf* consegue realizar esta tarefa da seguinte forma. O atacante envia pacotes ICMP para o endereço IP de difusão de uma determinada rede usando o endereço IP de origem da vítima. Como conseqüência, todas as estações daquela rede respondem à requisição, enviando uma resposta para a vítima, que acaba "inundada" por um número extremamente volumoso de pacotes.

Tabela 2 – Base de dados de conexões de rede

Atributo	Descrição	Exemplos de Valores
Duration	Duração da conexão em segundos.	'0', '1-60', '>60'.
ProtocolType	Tipo de protocolo.	'tcp', 'udp', 'icmp'.
Service	Serviço de rede.	'http', 'telnet', 'ftp', 'echo reply' e outros.
SrcBytes	Número de bytes de dados enviados pelo cliente ao servidor.	Variando entre '0' e '>1032'.
DstBytes	Número de bytes de dados enviados pelo servidor ao cliente.	Variando entre '0' e '>1024'.
Flag	Status da conexão	'SF', 'RSTO', 'S0' e outros.
FailedLogins	Número de tentativas de login que falharam.	'0', '>0'.
NumRoot	Número de acessos "root"	'0', '>0'.
Count	Número de conexões para o mesmo servidor idênticas à conexão atual numa janela de 2 segundos.	'1', '2', '3-10', '11-500'.
SrvCount	Número de conexões para o mesmo servidor com o mesmo serviço da conexão atual numa janela de 2 segundos.	'1', '2', '3-10', '11-500'.
Class	Indica se a conexão é uma conexão normal ou algum tipo de intrusão.	'normal', 'smurf', 'syn flood', 'guess password' e outros.

A seguir serão apresentados alguns resultados obtidos a partir da mineração da base de dados de conexões. Os exemplos apresentam exceções negativas e positivas com relação à medida de confiança que puderam ser descobertas a partir da implementação do algoritmo apresentado na Figura 2.

O primeiro teste realizado demonstra como a mineração de exceções pode ser utilizada para que sejam obtidas regras que caracterizam o tráfego normal de uma rede. A regra avaliada foi a seguinte:

R_1: *(Service = "echo reply")* ⇒ *(Class = "normal")*

Conf = 0,12%.

Esta regra evidencia uma informação curiosa: apenas 0,12% das conexões registradas na base simplificada de [9] que possuem o serviço de rede "*echo reply*" (resposta a um *ping*) representaram conexões normais. Através da aplicação da técnica para mineração de exceções, foi possível descobrir algumas exceções positivas, que são apresentadas na Figura 3. Observando os resultados, é possível notar que as conexões com serviço "*echo reply*" têm uma enorme chance de corresponderem à conexões normais quando são avaliadas contra os conjuntos de prova *(SrvCount = "1"), (Count = "1") e (SrcBytes = "1-519").*

EP_1: *(Service = "echo reply")* $\overset{+c}{\Longrightarrow}$ *(Class = "normal") [(SrvCount = "1")]*
DC^+ *= 0,9987 – Conf = 94,20%.*

EP_2: *(Service = "echo reply")* $\overset{+c}{\Longrightarrow}$ *(Class = "normal") [(Count = "1")]*
DC^+ *= 0,9983 – Conf = 76,52%.*

EP_3: *(Service = "echo reply")* $\overset{+c}{\Longrightarrow}$ *(Class = "normal") [(SrcBytes = "1-519")]*
DC^+ *= 0,9989 – Conf = 98,29%.*

Figura 3 – Exceções à regra
(Service = "echo reply") ⇒ *(Class = "normal")*

O segundo teste foi executado para demonstrar que as exceções podem ser capazes de representar os padrões de uma intrusão. A regra avaliada foi a seguinte:

R_2: *(Service = "echo reply")* ⇒ *(Class = "smurf")*

Conf = 99,12%.

Esta regra indica que, na base na base simplificada de [9], a grande maioria (99,12%) das conexões cujo serviço de rede é "*echo reply*", representaram intrusões do tipo *smurf*. A Figura 4 apresenta exceções negativas e positivas a esta regra.

EP_1: *(Service = "echo reply")* $\overset{+c}{\Longrightarrow}$ *(Class = "normal") [(SrvCount = "1")]*
DC^+ *= 0,9987 – Conf = 94,20%*
.
EP_2: *(Service = "echo reply")* $\overset{+c}{\Longrightarrow}$ *(Class = "normal") [(Count = "1")]*
DC^+ *= 0,9983 – Conf = 76,52%.*

EP_3: *(Service = "echo reply")* $\overset{+c}{\Longrightarrow}$ *(Class = "normal") [(SrcBytes = "1-519")]*
DC^+ *= 0,9989 – Conf = 98,29%.*

Figura 4 – Exceções à regra
(Service = "echo reply") ⇒ *(Class = "smurf")*

As exceções positivas EP_4 e EP_5 indicam conjuntos de prova que tornam a regra, que originalmente é forte, ainda mais forte. A exceção negativa EN_1 indica um conjunto de prova que é capaz de quebrar a regra forte.

Por fim, realizou-se o teste da seguinte regra:

R_3: *(Service = "telnet")* ∧ *(FailedLogins = ">0")* ⇒ *(Class = "guess password")*
Conf = 83,87%.

Esta regra indica que a grande maioria (83,87%) das conexões com o serviço *telnet* e que apresentaram uma ou mais tentativas de *login* não aceitas, representaram uma tentativa de intrusão do tipo *guess password*. A Figura 5 apresenta uma exceção positiva e uma exceção negativa a esta regra.

EP_6: *(Service = "telnet")* ∧ *(FailedLogins = ">0")* $\overset{+c}{\Longrightarrow}$ *(Class = "guess password") [(SrvCount = "2")]*

DC^+ *= 0,1613 – Conf = 100%.*

EN_2: *(Service = "telnet")* ∧ *(FailedLogins = ">0")* $\overset{-c}{\Longrightarrow}$ *(Class = "guess password") [(Flag = "SF")]*

DC^- *= 0,8916 – Conf = 9,09%.*

Figura 5 – Exceções à regra *(Service = "telnet")* ∧ *(FailedLogins = ">0")* ⇒ *(Class = "guess password")*

6 Conclusões

Este trabalho apresentou a aplicação de uma técnica para a mineração de exceções que visa melhorar a qualidade das bases de dados de regras que são utilizadas por alguns modelos de sistemas para detecção de intrusões. As exceções representam regras de associação cuja força afasta-se de seus valores médios em determinados subconjuntos de uma base de dados.

Uma das principais vantagens da mineração de exceções é o fato de que o método permite que as regras de associação e os atributos que formam os conjuntos de prova sejam combinados de maneira *ad-hoc* por um usuário no processo de busca por exceções. Outra vantagem é o fato do método ser capaz de descobrir padrões muito raros em bases de dados. Além disso, a mineração de exceções pode ser utilizada para auxiliar

na construção de bases de regras para sistemas baseados na detecção de anomalias e também para sistemas baseados na detecção de abusos.

Os principais problemas da técnica para a mineração de exceções são identificados a seguir. Em primeiro lugar, a técnica necessita de uma base de dados de tráfego de rede pré-processada e bem construída, para que regras possam ser descobertas, o que é um problema de todas as abordagens baseadas em mineração de dados quando aplicadas ao campo de redes de computadores. Um segundo problema, específico da mineração de exceções, consiste no fato da técnica não trabalhar com atributos numéricos (todos os atributos deste tipo precisam ser discretizados na fase de pré-processamento da base).

Neste trabalho, a mineração de exceções foi testada sobre uma base de conexões de rede que foi construída artificialmente, para um programa de pesquisa no campo da detecção de intrusões. Como trabalho futuro, pretende-se realizar uma avaliação do algoritmo para mineração de exceções sobre bases que possuam registros de conexões reais com o objetivo de identificar um conjunto de exceções que representem informações raras e surpreendentes. Pretende-se também realizar uma avaliação do desempenho do algoritmo a partir destas bases.

Referências

[1] AGRAWAL, R., IMIELINSKI, T. & SRIKANT, R. Mining Association Rules between Sets of Items in Large Databases. *Prodeedings of the ACM SIGMOD International Conference on Management of Data*, Washington, EUA, 1993, 207--216.

[2] ARLITT, M. & WILLIAMSON, C. An Analysis of TCP Reset Behaviour on the Internet. *ACM SIGCOMM Computer Communication Review*, 35 (1), 2005, 37-44.

[3] BARBARA, D., COUTO, J., JAJODIA, S. & WU, N. ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection, *ACM SIGMOD Record*, 30 (4), 2001, 15-24.

[4] BLOEDORM, E., CHRISTIANSEN, A. D., HILL, W., SKOURUPKA, C., TALBOT, L. M. & TIVEL, J. Data Mining for Network Intrusion Detection: How to Get Started. *MITRE Technical Report*, 2001.

[5] ELMASRI, R. & NAVATHE, S. B, Fundamentals of Database Systems, Addison Wesley, 4a Edição, 2004.

[6] FAYYAD, U. M., PIATETSKY-SHAPIRO, G. & SMITH, P. From data mining to knowledge discovery: an overview, *Advances in Knowledge Discovery and Data Mining*, 1-34. AAAI/MIT Press, 1996.

[7] GONÇALVES, E. C. & PLASTINO, A. Mineração de Regras de Associação Híbridas. *XXV Congresso da SBC / ENIA – Encontro Nacional de Inteligência Artificial*, São Leopoldo, Brasil, 2005.

[8] HAN, J. & KAMBER, M. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

[9] HETTICH, S. e BAY, S. D. The UCI KDD Archive / KDD Cup 1999 Data Set, *Home page*. http://kdd.ics.uci.edu/databases/kddcup99/task.html, (10/04/2006).

[10] LAUFER, R. P., VELLOSO, P. B. & DUARTE, O. C. M. B. Um Novo Sistema de Rastreamento de Pacotes IP contra Ataques de Negação de Serviço. *XIII Simpósio Brasileiro de Redes de Computadores, SBRC'2005*, Fortaleza, Brasil, 2005.

[11] LEE, W. & STOLFO, S. J. Data Mining Approaches for Intrusion Detection. *Proceedings of the 7th USENIX Security Simposium*, San Antonio, EUA, 1998.

[12] MIT LINCOLN LABORATORIES. DARPA Intrusion Evaluation Detection, *Home page,* http://www.ll.mit.edu/IST/ideval/ (15/04/2006).

[13] NING, P. & JAJODIA, S. Intrusion Detection Techniques, *The Internet Encyclopedia*. John Wiley & Sons, 2003.

[14] NING, P. & XU, D. Hypothesizing and Reasoning about Attacks Missed by Intrusion Detection Systems. *ACM Transactions on Information and System Security (TISSEC) archive*, 7 (4), 2004, 591-627.

[15] NONG, Y. & FARLEY, T. A Scientific Approach to Cyberattack Detection. *IEEE Computer*, 38 (11), 2005, 55-61.

[16] DICIONÁRIO BRASILEIRO DA LÍNGUA PORTUGUESA. Rio de Janeiro, Organizações Globo, 1993, p. 430.

[17] RESENDE, S. O. Mineração de Dados, *Mini-Curso, XXV Congresso da SBC / ENIA – Encontro Nacional de Inteligência Artificial*, São Leopoldo, Brasil, 2005.

[18] ROESCH, M. Snort - Lightweight Intrusion Detection for Networks. *Proceedings of the 13th USENIX Conference on System Administration*, Seattle, USA, 1999.

[19] SNORT INTRUSION DETECTION AND PREVENTION SYSTEM, *Home page,* http://www.snort.org (11/04/2006).

[20] SUZUKI, E. Discovering Interesting Exception Rules with Rule Pair. *Proceedings of the ECML/PKDD 2004 Workshop on Advances in Inductive Rule Learning*, Pisa, Itália, 2004, 163-178.

[21] TCPDUMP PUBLIC REPOSITORY, *Home page,* http://www.tcpdump.org (15/04/2006).