Avaliação do Emprego de Raciocínio baseado em Casos para Identificar Cenários de Intrusão em Logs de Firewalls

Samir Lohmann, Luciano Paschoal Gaspary, Cristina Melchiors

Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPCA)
Universidade do Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos 950 – 93022-000 – São Leopoldo – RS – Brasil

{samir,paschoal,cmelch}@unisinos.br

Abstract. *The content analysis of firewall logs is fundamental to recognize suspicious event sequences that indicate strategies used by intruders in an attempt to obtain non-authorized access to stations and services. Such analysis, due to the large volume of stored log data, is not feasible to be performed by hand. This paper presents an approach that explores the case-based reasoning technique, from the Artificial Intelligence field, to identify, automatically, intrusion scenarios in firewall logs. The paper describes the evaluation of our approach carried out based on real log files generated by the university firewall, and discusses how the tunning of parameters that comprise a case influences alert generation, aiming at determining parameter combinations that lead to a satisfactory relation between detection of intrusion scenarios and number of alerts generated.*

Resumo. *A análise de conteúdo de logs gerados por firewalls mostra-se fundamental para reconhecer seqüências de eventos suspeitas que indiquem estratégias utilizadas por intrusos para tentar obter acesso não autorizado a estações e serviços. Tal análise, em função do grande volume de informações armazenadas, é inviável de ser realizada de forma manual.Este artigo apresenta uma abordagem que explora a técnica de raciocínio baseado em casos, da Inteligência Artificial, para identificar, de forma automática, cenários de intrusão em logs de firewalls. O artigo descreve a avaliação da abordagem, realizada com base em logs reais gerados pelo firewall da universidade, e discute como a sintonia dos parâmetros que compõem um caso influencia na geração de alertas, procurando determinar combinações que propiciem uma relação satisfatória entre detecção de cenários de intrusão x número de alertas gerados.*

1. Introdução

A estratégia de utilizar um *firewall* como mecanismo de segurança de borda permite centralizar, em apenas uma máquina, todo tráfego que provém da Internet com destino à rede privada e vice-versa. Nesse ponto de controle, todo e qualquer pacote (HTTP, FTP, SMTP, SSH, IMAP, POP3, entre outros) que entra ou sai é inspecionado, podendo ser aceito ou rejeitado, conforme as regras de segurança estabelecidas [Stallings 2000; Taylor 2002].

Nesse contexto, os *firewalls* armazenam – para cada acesso bem sucedido ou tentativa frustrada – registros em arquivos de *log*. Alguns dos dados registrados são: tipo de operação, endereços origem e destino, portas local e remota, entre outras. Dependendo do tamanho da rede e de seu tráfego, o *log* diário gerado pode ser maior que 1 GB [Symantec 2001]. Do ponto de vista da gerência de segurança este *log* é rico em informações, pois permite, entre outros aspectos, reconhecer seqüências de eventos suspeitas que indiquem estratégias utilizadas por invasores para tentar obter acesso indevido a estações e serviços.

Apesar de se reconhecer a importância desses indicadores, o crescimento da quantidade e da complexidade das informações transitadas diariamente entres as redes privadas e a Internet tem tornado inviável o controle manual dos arquivos de *log*. Diante do problema referido, este artigo apresenta uma abordagem que explora a técnica de raciocínio baseado em casos, da Inteligência Artificial, para identificar, de forma automática, cenários de intrusão em *logs* de *firewalls*. A abordagem fornece condições para que cenários de intrusão possam ser modelados como casos; assim, sempre que seqüências semelhantes se repetirem nos *logs*, a abordagem é capaz de identificá-las e notificar o gerente. O artigo enfatiza a avaliação da abordagem – introduzida pela primeira vez em [Locatelli 2004] – realizada com base em *logs* reais gerados pelo *firewall* da universidade, e discute como a sintonia dos parâmetros que compõem um caso influencia na geração de alertas, procurando determinar combinações que propiciem uma relação satisfatória entre detecção de cenários de intrusão *x* número de alertas gerados.

O artigo está organizado da seguinte forma: a seção 2 descreve trabalhos relacionados. Na seção 3 apresenta-se a abordagem proposta para identificar, automaticamente, cenários de intrusão. A seção 4 aborda a ferramenta desenvolvida e a seção 5, a avaliação experimental realizada. Por fim, a seção 6 encerra o artigo com considerações finais e perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Uma caracterização quantitativa das atividades de intrusão efetuadas na Internet global, efetuada com base na análise de *logs* de *firewalls*, foi realizada por Yegneswaran em [Yegneswaran 2003]. O trabalho envolveu a coleta, durante um período de 4 meses, de mais de 1.600 *logs* de *firewalls* e sistemas de detecção de intrusão distribuídos pelo mundo. Os resultados permitiram caracterizar diversos tipos de varreduras e sua relação com a disseminação de vírus e *worms*. Pesa sobre o trabalho o fato de ter sido realizado de forma *ad-hoc*, sem o apoio de ferramentas (o que compromete a sua utilização contínua). Além disso, a abordagem é exclusivamente quantitativa, o que dificulta o entendimento de algumas situações em que os eventos precisam ser analisados mais de perto para se poder confirmar uma atividade suspeita.

Em relação à análise de eventos, técnicas da Inteligência Artificial têm sido aplicadas para relacionar eventos gerados por sistemas de detecção de intrusão (e não por *firewalls*) [Debar 2001; Ning 2002; Porras 2002]. Ning em [Ning 2002] apresenta um método que correlaciona pré-requisitos e conseqüências de alertas gerados por sistemas de detecção de intrusão a fim de determinar os vários estágios de um ataque. Os autores sustentam o argumento de que um ataque geralmente tem diferentes estágios e não acontece isoladamente, ou seja, cada estágio do ataque é pré-requisito para o próximo. Por exemplo, uma varredura de portas pode identificar os *hosts* que possuem serviços vulneráveis; com base nisso, o atacante pode explorar esses *hosts* para executar código arbitrário com privilégios do sistema local ou causar uma negação de serviço. O fato de pré-requisitos e conseqüências serem modelados como *predicados* dificulta a implantação em larga escala da abordagem, uma vez que a definição desses predicados não é uma tarefa fácil e a base de casos precisa ser constantemente atualizada, o que requer trabalho substancial. A proposta também é limitada ao não ser efetiva para identificar ataques onde a relação causa e conseqüência não pode ser estabelecida. Por exemplo, dois ataques (*Smurf* e SYN *flooding*) disparados quase ao mesmo tempo contra o mesmo alvo a partir de dois locais diferentes não seriam relacionados (embora exista forte conexão entre eles: mesmo instante e mesmo alvo).

As abordagens descritas em [Debar 2001; Porras 2002] são capazes de analisar alertas gerados por dispositivos de segurança dispersos geograficamente. Ambas propõem algoritmos para agregação e correlação de alertas. A primeira define um modelo de dados unificado para representar alertas associados à detecção de intrusão e um conjunto de regras para processá-los. O algoritmo de deteção é capaz de identificar (*i*) alertas reportados por diferentes dispositivos,

mas que estão relacionados com o mesmo ataque (duplicatas), e (*ii*) alertas que estão relacionados e, portanto, são gerados juntos (conseqüências). A segunda abordagem utiliza estratégias como análise de topologia, priorização e agregação de alertas que possuem atributos comuns. Um cálculo de ordenação de incidentes é realizado usando adaptação do *framework* de Bayes para propagação de *crenças* em árvore. As duas abordagens mencionadas tendem a não lidar bem com a detecção de cenários que se diferenciam (mesmo que sutilmente) daqueles que foram descritos através de regras de fusão e agregação.

Outras técnicas da Inteligência Artificial têm sido aplicadas no processamento de eventos, mas ainda no contexto de sistemas de detecção de intrusão. Uma das mais empregadas é o paradigma *case-based reasoning* (raciocínio baseado em casos, RBC). Schwartz em [Schwartz 2002] apresenta uma ferramenta que aplica esse paradigma em um variação do sistema de detecção de intrusão Snort, onde cada assinatura do sistema é mapeada para um caso. Outro sistema que utiliza o paradigma RBC é apresentado por Esmaili et al. em [Esmaili 1996]. Esse sistema utiliza RBC para detecção de intrusão utilizando os registros de auditoria produzidos pelo sistema operacional. Os casos representam cenários de intrusão, formados por seqüências de comandos do sistema operacional que resultam em um acesso não autorizado.

3. Modelagem de Cenários de Intrusão como Casos

Esta seção descreve a abordagem proposta pelo nosso grupo de pesquisa para analisar eventos gerados por *firewalls* e identificar, de forma automática, cenários de intrusão com o apoio da técnica de raciocínio baseado em casos.

A figura 1 ilustra um conjunto de eventos recuperado de um *log* gerado pelo Symantec Enterprise[1], *firewall* sobre o qual foi avaliada a abordagem proposta. Nela podem ser identificados três comportamentos suspeitos, detalhados abaixo.

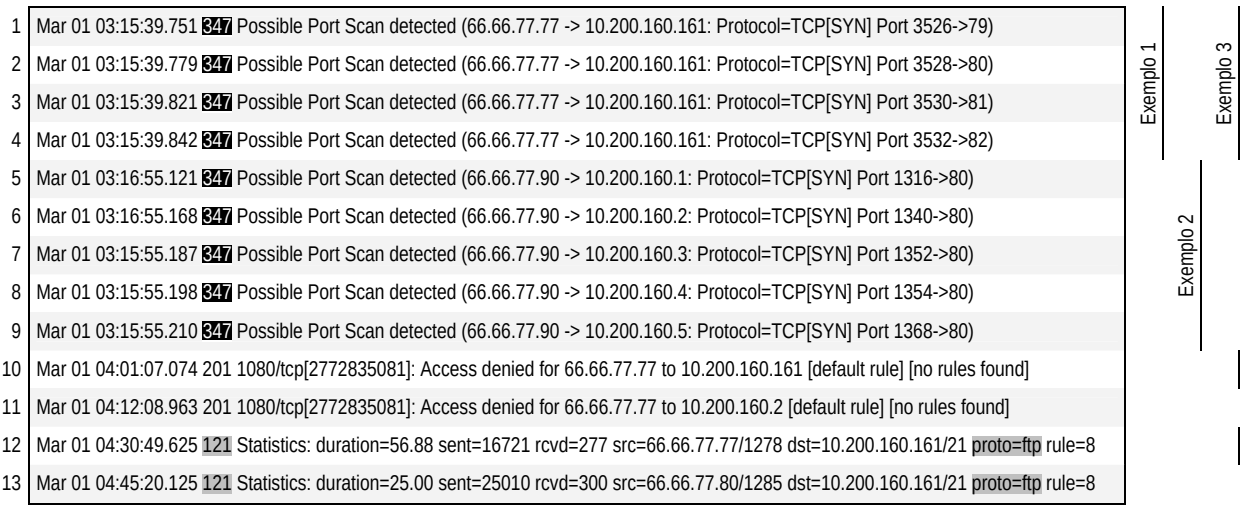

Figura 1. Conjunto real de eventos extraídos de um *log* e suas relações

Exemplo 1. O primeiro consiste de uma varredura de portas vertical e é composto pelos eventos 1, 2, 3 e 4. Essa varredura se caracteriza por sondagens oriundas de apenas um endereço IP destinadas a múltiplas portas de outro endereço IP. Observe que foram disparadas quatro sondagens, em menos de um segundo, do *host* 66.66.77.77 para o *host* 10.200.160.161.

[1] Os endereços IP foram substituídos por valores fictícios.

Exemplo 2. O segundo comportamento suspeito compreende uma varredura de portas horizontal e inclui os eventos 5, 6, 7, 8 e 9. Nesse caso, as sondagens partem de apenas um endereço IP e são destinadas a uma única porta de múltiplos endereços IP. Como pode ser observado na figura 1, o provável invasor 66.66.77.90 sondou a porta 80 de vários *hosts* diferentes em busca de algum em que houvesse um servidor HTTP disponível.

Exemplo 3. Por fim, o terceiro cenário de intrusão corresponde a uma varredura seguida de um acesso bem sucedido, incluindo os eventos 1, 2, 3, 4, 10 e 12. A estação 10.200.160.161 sofreu quatro sondagens (portas 79, 80, 81 e 82) e uma tentativa de acesso mal sucedida à porta 1080. Tanto as sondagens quanto a tentativa de acesso partiram do *host* 66.66.77.77 que, por fim, obteve acesso à estação utilizando o protocolo FTP (evento 12); permaneceu 56.88 segundos conectado, enviou 16.721 bytes e recebeu 277. O número um tanto quanto elevado de dados enviados indica que fez *upload* na estação alvo (10.200.160.161).

Em virtude da grande quantidade de eventos gerados pelos *firewalls*, cenários como os recém mencionados passam muitas vezes despercebidos pelo gerente de segurança. A abordagem detalhada nesta seção propõe a utilização do paradigma *case-based reasoning* (raciocínio baseado em casos, RBC) para identificar cenários de intrusão de forma automática.

Raciocínio baseado em casos [Kolodner 1993] é um paradigma da Inteligência Artificial que utiliza o conhecimento de experiências anteriores para propor soluções em novas situações. As experiências passadas são armazenadas em um sistema RBC como casos. Durante o processo de raciocínio para a resolução de uma nova situação, esta é comparada aos casos armazenados na base de conhecimento e os casos mais similares são utilizados para propor soluções ao problema corrente.

O paradigma RBC tem diversas vantagens sobre outros paradigmas de raciocínio. Uma delas diz respeito à facilidade na aquisição do conhecimento, que é realizada buscando-se experiências reais de situações passadas [Esmaili 1996]. Outra vantagem é a possibilidade de se obter casamento parcial entre a nova situação e os casos, permitindo maior flexibilidade em domínios onde os sintomas e as condições do problema podem ter pequenas variações ao ocorrerem em situações reais.

3.1. Estrutura do caso

Em nossa abordagem um caso armazenado representa um possível cenário de intrusão ou atividade suspeita, que pode ser identificada a partir dos eventos arquivados pelo *firewall* no *log*. A estrutura de um caso é apresentada na figura 2a. Como pode ser observado, um caso é formado por: (a) parte administrativa, com campos para identificação e anotações, que não são utilizados durante o processo de raciocínio; (b) parte classificatória, que contém campo usado para dividir o *log* em partes (conforme será demonstrado a seguir); e (c) parte descritiva, que contém os atributos utilizados no casamento do caso.

A similaridade entre os eventos do *log* real e os casos armazenados é calculada pela presença no *log* de eventos com determinadas características, o que foi denominado na abordagem de *sintoma*. Um sintoma é a representação de um ou vários eventos suspeitos, que devem ser identificados no *log* para que o caso armazenado seja similar à situação corrente.

Um caso pode conter um ou mais sintomas, conforme as características do cenário de intrusão ou da atividade suspeita sendo descrita. Um exemplo de caso com dois sintomas é apresentado na figura 2b. O caso modelado, simplificado para facilitar a descrição da abordagem, sugere que um alarme seja gerado sempre que forem observados *em torno de* cinco sondagens seguidas de um acesso bem sucedido partindo de uma mesma estação origem. O sintoma S_1 representa eventos do tipo PORT_SCANNING, tais como os eventos 1 a 4 da figura 1, enquanto o sintoma S_2 representa eventos do tipo STATISTIC, como o 12 na mesma figura.

Os parâmetros dos eventos do *log* tais como data, hora, tipo do evento e IP origem são representados em um caso como atributos do evento que compõe o sintoma. Nem todos os atributos precisam estar definidos (preenchidos); apenas os definidos serão utilizados no cálculo da similaridade (apresentado a seguir). Considerando o caso A da figura 2b, apenas o atributo *Tipo_Evento* está sendo usado para identificar o evento que constitui o sintoma S_1. O mesmo ocorre com a definição do sintoma S_2.

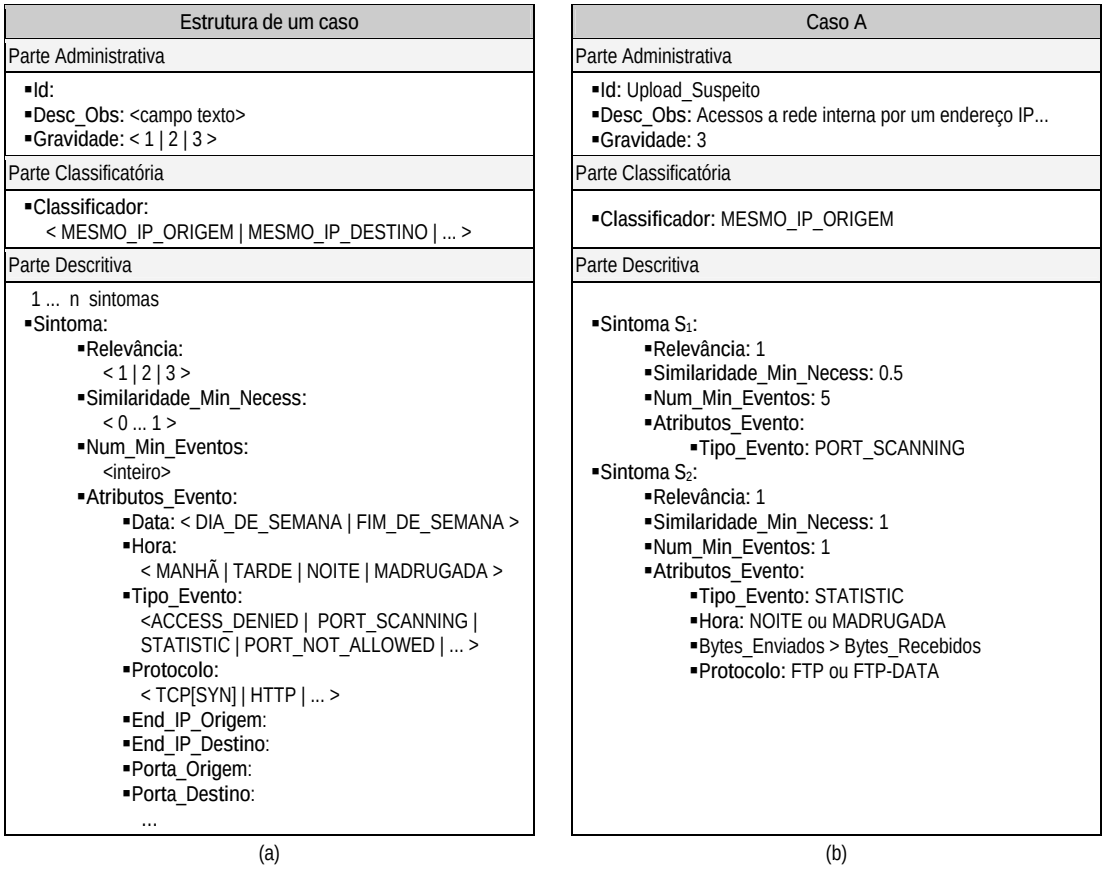

Figura 2. Modelagem de cenários de intrusão e atividades suspeitas como casos

3.2. Processo de raciocínio

O casamento dos eventos do *log* com um caso armazenado inicia pela separação desses eventos em partes. O critério a ser adotado nessa separação é determinado pelo campo *classificador* (vide figura 2a). Cada parte é chamada *caso corrente* e é comparada com o caso armazenado de forma separada. Tome-se como ilustração a comparação dos eventos do *log* apresentados na figura 1 com o caso A, figura 2b. O caso A tem como características classificadoras a utilização de mesmo endereço IP origem (campo *classificador* igual a MESMO_IP_ORIGEM). Assim, durante o processo de raciocínio, os eventos do *log* exemplo são divididos em dois casos diferentes, um contendo os eventos 1 a 4 e 10 a 12 (que chamaremos caso corrente 1) e outro contendo os eventos 5 a 9 (que chamaremos caso corrente 2).

Após a separação dos eventos do *log* em casos correntes, como explicado acima, cada caso corrente deve ser comparado ao caso armazenado para se calcular a similaridade entre eles, através de um processo chamado casamento entre caso corrente e caso armazenado. Esse casamento é feito utilizando a similaridade dos eventos do caso corrente em relação a cada sintoma presente no caso armazenado, em uma etapa que é denominada casamento de um sintoma. Voltando ao caso A e ao caso corrente 1 do exemplo anterior, a similaridade entre eles

é calculada usando a similaridade dos sintomas do caso A, que são o sintoma S_1 e o sintoma S_2. Por fim, a similaridade de um sintoma é calculada com base na similaridade dos eventos do caso corrente com os atributos do evento daquele sintoma (*Atributos_Evento*). No exemplo, a similaridade do sintoma S_1 é calculada usando a similaridade de cada evento do caso corrente 1 (eventos 1 a 4 e 10 a 12) com os atributos do evento daquele sintoma (campo *Tipo_Evento* igual a PORT_SCANNING). Estas etapas são explicadas a seguir.

A similaridade de um evento do caso corrente com os atributos do evento de um sintoma do caso armazenado é calculada pelo somatório da similaridade de cada atributo definido no sintoma, dividido pelo número de atributos definidos. A abordagem permite que a similaridade dos atributos de um evento seja parcial ou total. Retomando o exemplo do casamento entre o caso corrente 1 e o caso A, no cálculo da similaridade dos eventos em relação ao sintoma S_1, há apenas um atributo definido que é o *Tipo_Evento*. A similaridade dos eventos 1 a 4 resulta em 1 (100%), pois esses eventos são do tipo PORT_SCANNING, que é o mesmo tipo de evento definido no atributo *Tipo_Evento*. Já a similaridade dos eventos 10 a 12 resulta em 0, pois esses eventos não são do tipo PORT_SCANNING. Considerando agora a similaridade do sintoma S_2, há quatro atributos definidos (tipo de evento, hora, bytes enviados > bytes recebidos e protocolo). No cálculo da similaridade de cada evento do caso corrente 1 em relação ao sintoma S_2, os eventos 1 a 4, 10 e 11 resultam em 0, enquanto que a similaridade do evento 12 resulta em 1 (campo *Tipo_Evento* igual a STATISTIC, *Hora* igual a MADRUGADA, bytes enviados (16.721) > bytes recebidos (277) e *Protocolo* igual a ftp).

Após o cálculo da similaridade dos eventos em relação a um sintoma, os eventos são ordenados pela sua similaridade. Os *n* eventos com similaridade maior são utilizados então para o casamento do sintoma, onde *n* indica o número mínimo de eventos necessário para haver similaridade total daquele sintoma (modelado no caso por *Num_Min_Eventos*). A similaridade do sintoma é calculada pelo somatório da similaridade desses *n* eventos dividido por *n*. Se a similaridade resultante para um sintoma é menor que a similaridade mínima definida para aquele sintoma no caso armazenado (modelado por *Similaridade_Min_Necess*), a comparação daquele caso corrente com o caso armazenado é interrompida, e o caso corrente é descartado. Lembrando o exemplo anterior, no sintoma S_1 a ordenação dos eventos pela sua similaridade resulta em {1, 2, 3, 4, 10, 12}. Como nesse sintoma o número mínimo de eventos para casamento total é 5, a similaridade do sintoma S_1 será calculada por (1 + 1 + 1 + 1 + 0)/5 = 0,8. Como a similaridade mínima estipulada no caso para o sintoma S_1 é 0,5, este sintoma é aceito e o processo continua, calculando a similaridade dos outros sintomas no caso, no exemplo, o sintoma S_2. Considerando agora o sintoma S_2, que tem número mínimo de eventos 1, a similaridade é calculada por (1)/1 = 1. Com similaridade 1, o sintoma S_2 também é aceito.

Por fim, após o cálculo do casamento de todos os sintomas presentes no caso armazenado, é feito o casamento do caso corrente com o caso armazenado. Esse cálculo é realizado considerando a similaridade dos sintomas e sua relevância, através da fórmula abaixo. Referenciando mais uma vez o caso corrente 1 e caso A, o grau de casamento final (isto é, casamento do caso corrente 1 com o caso armazenado A) será ((1 × 0,8) + (1 × 1))/2 = 0,9, ou seja, de 90%. Nesse exemplo, ambos sintomas possuem mesma importância (*Relevância*), mas atribuir pesos diferentes pode ser necessário em outras situações.

$$\frac{\sum_{i=1}^{ns} r_i \times sim\_sintoma_i}{\sum_{1}^{ns} r_i}$$, onde *ns* é o número de sintomas do caso armazenado, *ri* é a relevância do sintoma *i* e $sim\_sintoma_i$ é a similaridade do sintoma *i*.

Quando o grau de similaridade do caso corrente com um caso armazenado é maior que um valor pré-definido (por exemplo, valor 7,5), o caso corrente é selecionado como suspeito, indicando uma situação que deve ser informada ao gerente de segurança. Quando um caso é selecionado, alguns parâmetros adicionais são instanciados com dados do caso corrente, num

processo de adaptação, a fim de ser possível apresentar ao gerente uma visão clara e rápida do problema identificado. Um exemplo é a instanciação do atributo endereço IP origem para os casos em que o classificador corresponde a MESMO_IP_ORIGEM, como no caso A. Usando essa instanciação, no exemplo do caso corrente 1 comentado ao longo desta seção, a atitude suspeita pode ser apresentada como *Upload Suspeito* detectada para o endereço IP origem 66.66.77.77.

4. O Ambiente CIACE (Case-based Intrusion Alert Correlation Environment)

Para validar a abordagem apresentada foi desenvolvido o ambiente CIACE (Case-based Intrusion Alert Correlation Environment). Esta seção descreve a arquitetura e algumas das principais funcionalidades do ambiente.

4.1. Arquitetura

O ambiente foi desenvolvido em ambiente Linux, utilizando as linguagens de programação Perl e PHP, o servidor Apache e o banco de dados MySQL. A figura 3 ilustra a arquitetura de CIACE, incluindo seus componentes e a interação entre eles. O módulo *parser* é responsável por processar os arquivos de *log* (1) e inserir os principais atributos de cada evento (ex: tipo de operação, endereços origem e destino, portas local e remota, entre outras) na base de dados (2).

Cada tipo de evento é armazenado em uma tabela distinta. Alguns atributos, por serem comuns a dois ou mais tipos de eventos, se repetem nas tabelas correspondentes. Esse esquema foi adotado em detrimento a um normalizado porque nesse último, para cada evento inserido na base de dados, seriam necessárias, em média, seis consultas e sete inserções (comprometendo o desempenho da fase de processamento).

A partir de um navegador *web* o gerente de segurança interage com o núcleo do ambiente, que foi implementado em um conjunto de *scripts* PHP (3, 4). Essa interação permite incluir, remover e atualizar casos na base de casos exemplo (3, 4, 6), conforme detalhado na seção 3, e configurar parâmetros de funcionamento da máquina de raciocínio (3, 4, 9). A identificação de cenários de intrusão é realizada automaticamente à noite após a ferramenta popular a base de dados com os eventos do log do dia corrente (módulo parser). Nesse momento, a máquina de raciocínio busca no banco de dados os eventos de interesse (8) e os confronta com os casos exemplo (7). Sempre que um novo comportamento suspeito é identificado, o módulo inclui um alarme no banco de dados (8), que se tornará visível ao gerente de segurança.

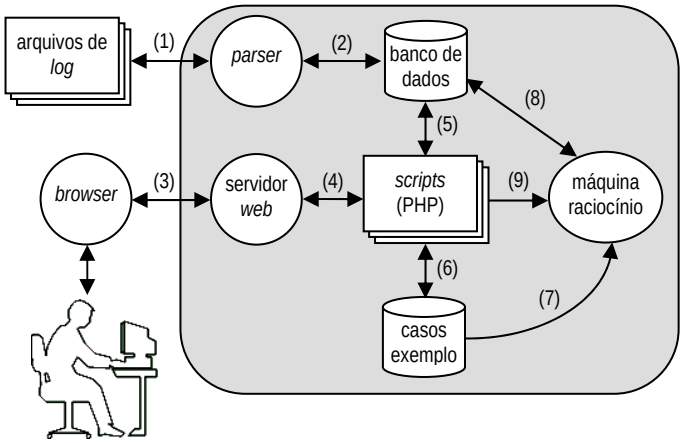

Figura 3. Componentes da arquitetura de CIACE e suas relações

4.2. Funcionalidades

A figura 4 ilustra algumas das funcionalidades oferecidas pelo ambiente CIACE. Em (a) é possível visualizar o formulário usado para inclusão de um novo caso ou edição de um já existente. Em (b) são ilustrados os alertas gerados por CIACE, após a execução do mecanismo de raciocínio. Como é possível observar, junto a cada alerta é informado o seu grau de similaridade com o caso armazenado correspondente. Por fim, em (c) é apresentado um resumo de todos os alertas. Nesse caso, para cada tipo de alerta são informados o número de ocorrências e a similaridade média.

O ambiente fornece, adicionalmente, uma interface de consulta que permite recuperar, mesclando todos os tipos de eventos, aqueles que satisfazem os critérios definidos pelo gerente de segurança (ex: mesmo endereço origem e/ou destino, porta local e/ou remota). Essa funcionalidade habilita a descoberta de novos casos que podem ser usados pelo gerente de segurança para alimentar a base de casos exemplo, em um processo denominado aprendizado.

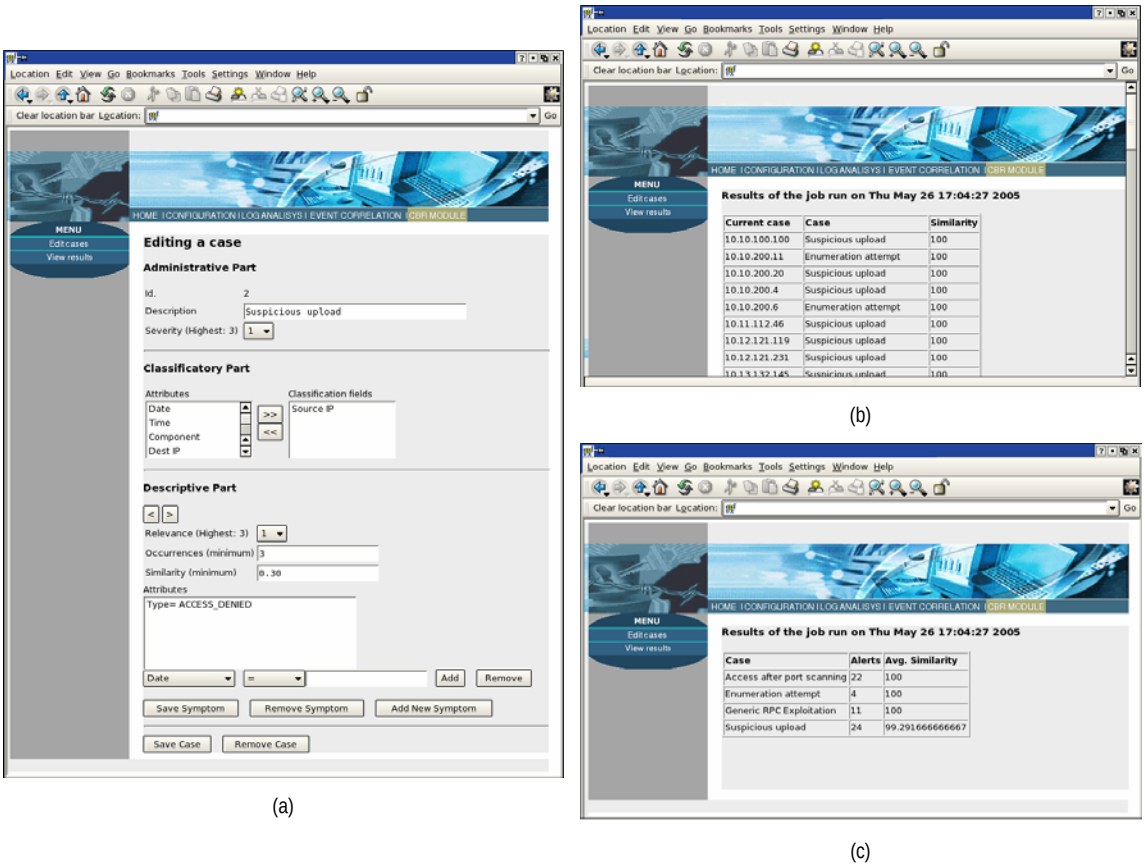

Figura 4. Interface gráfica de CIACE e suas funcionalidades

5. Avaliação Experimental

A avaliação da abordagem proposta compreendeu a realização de três experimentos: (*i*) modelagem de casos e sua detecção em logs teste e controle; (*ii*) ajuste de similaridade mínima dos casos; e (*iii*) ajuste dos parâmetros de um caso. O primeiro experimento teve por objetivo avaliar se cenários de intrusão definidos com base em um arquivo de *log* (teste) continuariam sendo identificados em arquivos de *log* posteriores (controle). Os dois últimos experimentos, por sua vez, visaram à identificação do impacto que mudanças na similaridade mínima dos casos e nos parâmetros de um caso particular podem provocar na relação entre detecção de cenários de intrusão *x* número de alertas gerados.

5.1. Modelagem de casos e sua detecção em *logs* teste e controle

O ambiente CIACE foi utilizado para executar a abordagem proposta com base em dois conjuntos distintos de *logs*: o primeiro, denominado conjunto de teste, consistiu dos eventos gerados pelo *firewall* de borda usado na Universidade do Vale do Rio dos Sinos no período de 28/09/2003 a 02/10/2003; o segundo conjunto, denominado controle, correspondeu aos eventos gerados no período de 14/12/2003 a 18/12/2003. Em ambos os casos, os eventos utilizados foram aqueles registrados de domingo a quinta-feira.

Para realizar a avaliação o ambiente CIACE foi instalado em um computador Pentium III 500 MHz com 256 KB de memória cache, 128 MB de memória RAM e dois discos rígidos (IDE): um com capacidade de 10 GB contendo o sistema operacional e outro de 40 GB contendo os arquivos de *log* e de banco de dados. O computador foi configurado com o sistema operacional Linux Mandrake versão 9.2, o gerenciador de banco de dados MySQL versão 4.0.15 e o servidor web Apache versão 2.0.47.

Com base em averiguações manuais no *log* de teste foram identificados diversos cenários suspeitos, dos quais seis foram modelados como casos: *Upload Suspeito* (ilustrado na figura 2b), *Tentativa de Conexão SQL*, *Tentativa de Enumeração*, *Exploração Genérica de RPC*, *Exploração de Netbios* e *Ataque RPC e POP3*. Por restrição de espaço apenas os dois últimos são explicados a seguir.

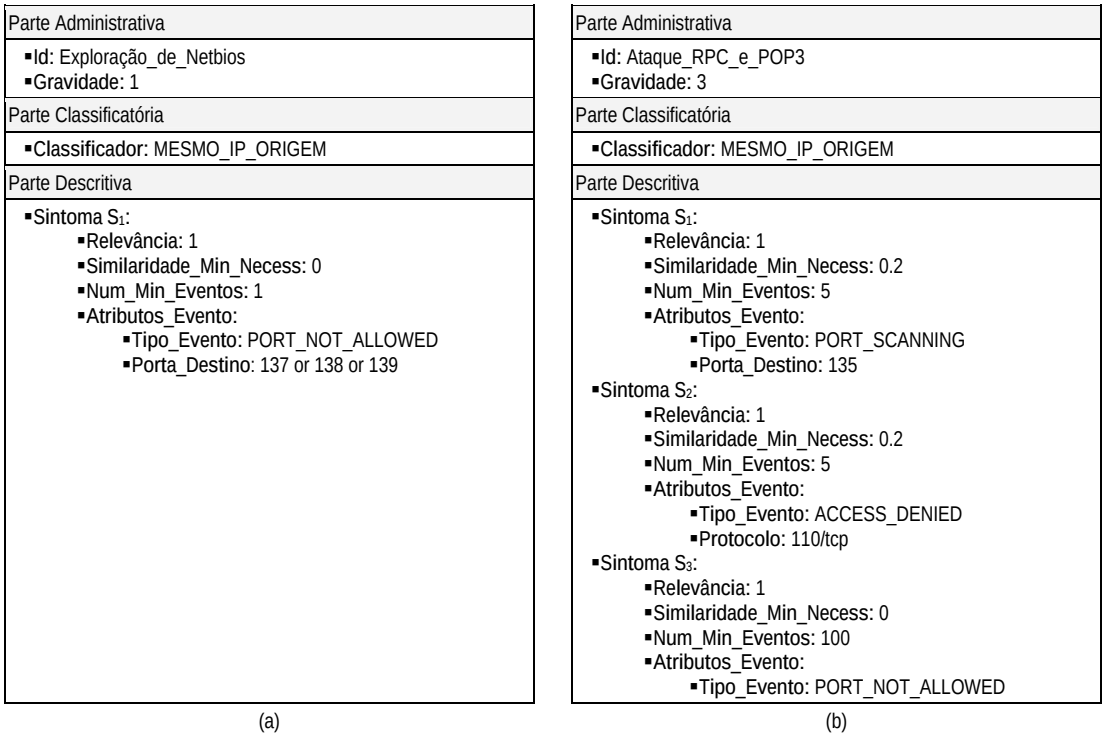

Figura 5. Casos *Exploração de Netbios* e *Ataque RPC e POP3*

O caso *Exploração de Netbios* (figura 5a) modela a identificação de tentativas de conexão às portas 137, 138 e 139. Essas portas são utilizadas pelos serviços que compõem o Netbios (*Network Basic Input Outupt System*), responsável por prover compartilhamento de arquivos no sistema operacional Windows. Os referidos serviços possuem vulnerabilidades que podem ser exploradas em ataques de negação de serviço e/ou para obtenção de acesso ao registro do Windows.

Já no caso *Ataque RPC e POP3* (figura 5b) é modelado um cenário de intrusão onde o atacante, além de tentar explorar vulnerabilidades do service RPC (sintoma 1), procura acessar o serviço POP3 (Post Office Protocol) e tem acesso negado (sintoma 2). Observa-se, adicionalmente, que o atacante executa centenas de tentativas de conexões em diversas portas, não obtendo sucesso (conforme modelado pelo sintoma 3).

Após ter povoado o ambiente com os casos recém enumerados, executou-se o mecanismo de raciocínio primeiro sobre o conjunto de teste e, posteriormente, sobre o conjunto de controle. Em ambas as execuções a similaridade mínima foi configurada com o valor 75%. Uma síntese dos resultados obtidos é ilustrada na tabela 1.

Tabela 1. Resultado do raciocínio

	Teste		Controle	
Casos armazenados	**Oc.**	**Sim.**	**Oc.**	**Sim.**
Upload Suspeito	15	96,6	24	99,29
Tentativa de Conexão SQL	1	100	0	-
Tentativa de Enumeração	8	100	4	100
Exploração Genérica de RPC	10	100	11	100
Exploração de Netbios	3	100	0	-
Ataque RPC e POP3	1	80	0	-

Oc.: num. ocorrências do caso; Sim.: similaridade média (%)

Tabela 2. Eventos presentes nos *logs*

Tipo de evento	Teste	Controle
STATISTIC	12.922.408	11.906.511
ACCESS_DENIED	1.760.325	4.023.992
CONNECTION_FAILED	292.436	31.145
PACKET_NOT_ENABLED	1.411	702
PORT_NOT_ALLOWED	90.703	79.867
PORT_SCANNING	58.605	96.332

Os casos armazenados mais gerais e com maior número de ocorrências, tais como *Exploração Genérica de RPC*, não tiveram alterações significativas no número de alertas gerados e na similaridade média dos casos encontrados. A exceção é o caso *Upload Suspeito*, para o qual foi gerado um número de alertas 62,5% maior no conjunto de controle em relação ao conjunto de teste. O elevado número de eventos do tipo ACCESS_DENIED encontrado no conjunto de controle explica essa diferença (tabela 2).

O caso *Tentativa de Enumeração* teve um número de ocorrências 50% menor no conjunto de controle. Esse caso possui um único sintoma, cujo atributo *Tipo_Evento* é PACKET_NOT_ENABLED. O conjunto de controle, de fato, apresentou a metade de eventos desse tipo em relação ao conjunto de teste, conforme apresentado na tabela 2.

Alguns casos (*Exploração de Netbios*, *Ataque RPC e POP3* e *Tentativa de Conexão SQL*) não foram observados no conjunto de controle. Atribui-se esse resultado ao fato de que logo que uma vulnerabilidade é identificada, os fabricantes de sistemas operacionais e antivírus disponibilizam correções e atualizações para eliminá-la. Em conseqüência disso, tentativas de ataques que exploram tais vulnerabilidades passam a acontecer com menor freqüência.

5.2. Ajuste de similaridade mínima dos casos

O segundo experimento consistiu na execução do mecanismo de raciocínio, povoado com os seis casos especificados, sobre o conjunto de teste, variando a similaridade mínima exigida para seleção de casos correntes em cada rodada (50%, 51%, 75%, 90% e 100%). Os resultados obtidos são resumidos na tabela 3.

Ao executar o mecanismo de raciocínio com 50% de similaridade mínima, observou-se que determinados casos, como *Upload Suspeito*, apresentaram um número elevado de alertas, com similaridade média bastante baixa (um pouco acima de 50%). Isto se deveu ao fato de o caso possuir dois sintomas, sendo que um deles – S_1, cinco *port scannings* – é característico de muitos casos correntes. Como a similaridade mínima foi configurada em 50%, todos os casos correntes que apresentaram esse sintoma foram selecionados como *Upload Suspeito*. Ao executar o mecanismo com 51% de similaridade mínima, percebeu-se que o problema de explosão de alertas não ocorreu. Assim, caso se deseje executar o mecanismo de raciocínio com

máxima tolerância a casos correntes pouco semelhantes aos armazenados, recomenda-se determinar uma similaridade mínima de 51%.

Tabela 3. Resultado do raciocínio usando diferentes similaridades mínimas

Similaridade mínima	50%		51%		75%		90%		100%	
Casos armazenados	**Oc.**	**Sim.**	**Oc.**	**Sim.**	**Oc.**	**Sim.**	**Oc.**	**Sim.**	**Oc.**	**Sim.**
Upload Suspeito	59.592	50,01	17	93,11	15	96,6	12	100	12	100
Tentativa de Conexão SQL	1	100	1	100	1	100	1	100	1	100
Tentativa de Enumeração	8	100	8	100	8	100	8	100	8	100
Exploração Genérica de RPC	10	100	10	100	10	100	10	100	10	100
Exploração de Netbios	3	100	3	100	3	100	3	100	3	100
Ataque RPC e POP3	3	75,33	3	75,33	1	80	0	-	0	-

Oc.: número de ocorrências do caso armazenado; Sim.: similaridade média (em %)

As execuções do mecanismo de raciocínio usando similaridades mínimas superiores a 75% mostram que o ambiente se torna pouco tolerante a casos correntes similares aos casos armazenados. Com similaridade mínima de 90%, todos os casos correntes selecionados apresentaram similaridade média de 100%.

5.3. Ajuste dos parâmetros de um caso

No terceiro experimento foram realizadas alterações nos parâmetros do caso *Upload Suspeito*. A cada parâmetro modificado, o mecanismo de raciocínio foi executado, de forma não cumulative, sobre o conjunto de teste. Em todas as execuções a similaridade mínima exigida para seleção de casos correntes foi 75%. Os ajustes de parâmetros efetuados foram os seguintes:

- *aumento da similaridade mínima de um dos sintomas*: a similaridade mínima do sintoma S_1 do caso foi aumentada para 1 (ou 100%);
- *diminuição da similaridade mínima de um dos sintomas*: o mesmo sintoma teve sua similaridade mínima diminuída para zero, o que na prática significa que qualquer caso corrente será selecionado desde que exista pelo menos um evento no *log* cujos atributos coincidam com os definidos no referido sintoma;
- *aumento da relevância de um dos sintomas*: a relevância do sintoma S_2 foi aumentada para 3, o que o tornou três vezes mais importante que o sintoma S_1, que teve sua relevância mantida com o valor 1.

A tabela 4 ilustra o número de casos correntes selecionados pelo mecanismo de raciocínio após o ajuste de cada um dos parâmetros referidos. À direita são apresentados os casos selecionados sem o respectivo ajuste (usando o caso original). Como é possível verificar, os ajustes de parâmetros dos casos podem levar a modificações no número de ocorrências observadas nos arquivos de *log*. No caso *Upload Suspeito*, por exemplo, o aumento da relevância do sintoma S_2 (*upload* registrado pelo evento STATISTIC) permitiu a identificação de dois casos adicionais (em relação à execução usando o caso original). Pretende-se, como trabalho futuro, ampliar esse experimento para todos os casos especificados, variando ainda mais as combinações possíveis, visando a um melhor mapeamento do impacto causado pela modificação desses parâmetros.

Tabela 4. Resultado do raciocínio ajustando parâmetros do caso *Upload Suspeito*

	Com ajustes		Sem ajustes	
Ajustes realizados	**Oc.**	**Sim.**	**Oc.**	**Sim.**
Aumento da similaridade mínima de um dos sintomas	12	100	15	96,6
Diminuição da similaridade mínima de um dos sintomas	15	96,6	15	96,6
Aumento da relevância de um dos sintomas	17	96,6	15	96,6

6. Conclusões e Trabalhos Futuros

Garantir a segurança das informações mantidas pelas organizações é um requisito básico para suas operações, uma vez que a cada ano o número de incidentes cresce de modo exponencial. Todavia, para proteger as organizações, considerando a quantidade e a complexidade crescente dos ataques realizados, é preciso contar com técnicas e ferramentas que apóiem a análise de evidências (disponíveis, por exemplo, em arquivos de log) e, conseqüentemente, permitam a identificação de cenários de intrusão ou atividades suspeitas.

Este artigo apresentou uma abordagem, baseada na técnica de Raciocínio Baseado em Casos, para identificar automaticamente, junto a arquivos de *logs* gerados por *firewalls*, seqüências de eventos que representam cenários de intrusão ou atividades suspeitas. A abordagem foi avaliada com dois conjuntos de *logs* distintos, abrangendo períodos de cinco dias. Os resultados obtidos apontam que CBR é uma técnica adequada para a identificação de cenários de intrusão. A calibragem (i) de similaridade para seleção de casos e (ii) de parâmetros como similaridade e relevância de sintomas pode levar a uma relação satisfatória entre detecção de cenários de intrusão e número de alertas gerados.

Uma das limitações da abordagem reside no fato de os *logs* de *firewalls*, em geral, não possuírem todas as informações necessárias para a detecção de determinados cenários de intrusão. Por exemplo, uma execução de código remoto ou acesso ao registro do Windows, que acontecem depois que o intruso já passou pela barreira do *firewall*, não podem ser detectados. Para eliminar essa limitação, pretende-se associar *logs* de *firewall* com outros dados históricos, tais como *logs* de IDSs e de sistemas operacionais.

Referências

Debar, H. and Wespi, A. (2001) “Aggregation and Correlation of Intrusion-Detection Alerts”, In: Recent Advances in Intrusion Detection, LNCS, v. 2212, p. 85-103.

Esmaili, M. et al. (1996) “Case-Based Reasoning for Intrusion Detection”, In: Computer Security Applications Conference, p.214-223.

Kolodner, J. (1993) Case-Based Reasoning, Morgan Kaufmann.

Locatelli, F. E., Dillenburg, F., Melchiors, C., Gaspary, L. P. (2004) “Identificação de Cenários de Intrusão pela Classificação, Caracterização e Análise de Eventos gerados por Firewalls”, In: Simpósio Brasileiro de Redes de Computadores, v. 2, p. 851-864.

Ning, P., Cui, Y. and Reeves, D. (2002) “Analyzing Intensive Intrusion Alerts via Correlation”, In: Recent Advances in Intrusion Detection, LNCS, v. 2516, p. 74-94.

Porras, P. A. , Fong, M. W. , and Valdes, A. (2002) “A Mission-Impact-Based Approach to INFOSEC Alarm Correlation”, In: Recent Advances in Intrusion Detection, LNCS, v. 2516, p. 95-114.

Schwartz, D., Stoecklin, S. and Yilmaz, E. (2002) “A Case-Based Approach to Network Intrusion Detection”, In: International Conference on Information Fusion, p.1084-1089.

Stallings, W. (2000) Network Security Essentials: Applications and Standards, Prentice-Hall.

Symantec Enterprise Firewall, Symantec Enterprise VPN, and VelociRaptor Firewall Appliance Reference Guide. Symantec, 2001.

Taylor, T. (2002) Security Complete, Sybex.

Yegneswaran, V., Barford, P. and Ulrich, J. (2003) “Internet Intrusions: Global Characteristics and Prevalence”, In: ACM SIGMETRICS Performance Evaluation Review, v. 31, n. 1, p. 138-147.