Identificação de Cenários de Intrusão pela Classificação, Caracterização e Análise de Eventos gerados por Firewalls

Fábio Elias Locatelli, Fabiane Dillenburg, Cristina Melchiors, Luciano Gaspary

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

{locatelli,fabianed,cristina,paschoal}@exatas.unisinos.br

Abstract. *Despite neglected by most security managers due to the low availability of tools, the content analysis of firewall logs is fundamental (a) to measure and identify accesses to external and private networks, (b) to assess the historical growth of accesses volume and applications used, (c) to debug problems on the configuration of filtering rules and (d) to recognize suspicious event sequences that indicate strategies used by intruders in an attempt to obtain non-authorized access to stations and services. This paper presents an approach, accompanied of a tool, to classify, characterize and analyze events generated by firewalls. The proposed approach explores the case-based reasoning technique, from the Artificial Intelligence field, and visualization mechanisms in order to identify and emphasize possible intrusion scenarios. The paper also describes the validation of both our approach and tool carried out based on real logs generated along one week by the university firewall.*

Resumo. *Apesar de negligenciada pela maioria dos gerentes de segurança em função da pouca disponibilidade de ferramentas, a análise de conteúdo de logs gerados por firewalls mostra-se fundamental para (a) mensurar e identificar os acessos à rede privada e à externa, (b) acompanhar o crescimento histórico do volume de acessos e as aplicações utilizadas, (c) depurar problemas de configuração de regras de filtragem e, sobretudo, (d) reconhecer seqüências de eventos suspeitas que indiquem estratégias utilizadas por intrusos para tentar obter acesso não autorizado a estações e serviços. Este artigo apresenta uma abordagem, acompanhada de uma ferramenta, para classificação, caracterização e análise de eventos gerados por firewalls. A abordagem explora a técnica de raciocínio baseado em casos, da Inteligência Artificial, e mecanismos de visualização para identificar e realçar possíveis cenários de intrusão. O artigo descreve, ainda, a validação da abordagem e da ferramenta realizada com base em logs reais gerados ao longo de uma semana pelo firewall da universidade.*

1. Introdução

Uma das alternativas mais utilizadas pelas empresas como medida de proteção contra ataques é o *firewall*. Ele é caracterizado por ser uma barreira de segurança entre duas redes; sua maior função é bloquear todo o tráfego não autorizado oriundo de uma rede a outra. Alguns dos principais *firewalls* existentes hoje no mercado são o Firewall-1 [CheckPoint 2003], o Gauntlet [Secure Computing 2003], o Iptables [Netfilter 2003], o Guard [ISS 2003] e o Symantec Enterprise Firewall [Symantec 2003a].

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: (a) mensurar e identificar os acessos à rede privada e à externa (ex: serviços mais e menos requisitados, estações que ocupam mais ou menos banda, principais usuários); (b) acompanhar historicamente o crescimento do volume de acessos e as aplicações utilizadas; (c) depurar problemas de configuração de regras de filtragem e, sobretudo, (d) 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*.

Este artigo apresenta uma abordagem, acompanhada de uma ferramenta, para classificação, caracterização e análise de eventos gerados por *firewalls*. O artigo descreve, ainda, a validação da abordagem e da ferramenta realizada com base em *logs* reais gerados ao longo de uma semana pelo *firewall* da universidade. As principais contribuições científicas do trabalho se desdobram em três: (a) a abordagem permite identificar, através do agrupamento de eventos relacionados, seqüências de ações executadas a partir de, ou contra, uma determinada estação ou serviço; (b) com o apoio da técnica da Inteligência Artificial denominada raciocínio baseado em casos, a abordagem fornece condições para que cenários de intrusão possam ser modelados como casos; assim, sempre que seqüências semelhantes se repetirem, a abordagem é capaz de identificá-las e notificar o gerente; (c) a abordagem propõe o uso da técnica de visualização denominada *árvore hiperbólica* [Lamping 1996] para apresentar *snapshots* sintéticos dos eventos gerados pelo *firewall* em um determinado período.

O artigo está organizado da seguinte forma: a seção 2 descreve trabalhos relacionados. Na seção 3 apresenta-se a abordagem proposta para classificar e caracterizar os eventos armazenados pelo *firewall*, bem como para identificar automaticamente cenários de intrusão. A seção 4 aborda a ferramenta desenvolvida e a seção 5, o estudo de caso realizado para validá-la. Por fim, a seção 6 encerra o artigo com considerações finais e perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Ao mesmo tempo em que existem algumas ferramentas para a análise de *logs*, elas carecem de funcionalidades importantes tais como a possibilidade de apresentar uma visão histórica de ocorrência dos eventos analisados, mecanismos que contribuam para a visualização facilitada dos mesmos, recursos para a sua análise e identificação automática de atividades suspeitas. Este é o caso, por exemplo, das ferramentas `Flatten` [Symantec 2001] e `Reptor` [Wankwood 2003]. A primeira é um programa proprietário, fornecido pela Symantec, que atua sobre o *log* do *firewall* Symantec Enterprise. O objetivo dessa ferramenta é criar *logs* estruturados a partir dos *logs* semi-estruturados desse *firewall* para que outras ferramentas ou *scripts* tenham facilidade para processar os dados. `Reptor`, por sua vez, é uma ferramenta distribuída sob a licença GPL que gera alertas e relatórios oriundos da análise do *log* diário do Symantec Enterprise Firewall. Essa ferramenta, contudo, opera apenas sobre os eventos estatísticos (que são os bem sucedidos), ignorando quaisquer ocorrências de eventos ligados à segurança. Além disso, limita-se a processar os eventos armazenados em um único dia, impedindo qualquer análise a médio e longo prazo.

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]. 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).

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. Abordagem para Classificar, Caracterizar e Analisar Eventos de Firewalls

Esta seção descreve a abordagem proposta pelo nosso grupo de pesquisa para classificar, caracterizar e analisar eventos gerados por *firewalls*. A abordagem se estrutura em duas partes independentes e complementares. A primeira, mais quantitativa, permite agrupar eventos armazenados pelo *firewall* com base em um ou mais elementos de agregação (filtros) definidos pelo gerente de segurança. Presta-se para os propósitos de classificação e caracterização de eventos. A segunda parte se propõe a analisar esses mesmos eventos e identificar, de forma automática, cenários de intrusão (com o apoio da técnica de raciocínio baseado em casos). A seção inicia com a descrição do formato típico de *logs* de *firewalls* (sub-seção 3.1) e prossegue com a apresentação da abordagem (sub-seções 3.2 e 3.3).

3.1. Formato típico de eventos gerados por *firewalls*

A abordagem proposta pode ser aplicada a *logs* de quaisquer *firewalls*, uma vez que tendem a ter formatos semelhantes (apesar das características e funcionalidades particulares que implementam). A figura 1 ilustra um conjunto de eventos recuperado de um *log* gerado pelo Symantec Enterprise[1], *firewall* sobre o qual foi validada a abordagem proposta. Os eventos são organizados em cinco campos:

- *Timestamp*: informa o horário da ocorrência do evento;
- *System name*: indica o nome do *firewall*, sendo importante em situações onde os eventos de dois ou mais *firewalls* são armazenados em um mesmo arquivo de *log* (omitido do exemplo);
- *Component*: determina o serviço ou a aplicação (ex: *kernel* ou *gwcontrol*) responsável pela geração do evento (omitido do exemplo);
- *Message number*: é um número de três dígitos que identifica o tipo do evento. De acordo com o exemplo ilustrado na figura 1, 347 indica uma varredura de portas, 201 consiste de um acesso negado e 121, um evento estatístico (registra informações para cada conexão bem sucedida autorizada pelo *firewall*);
- *Message text*: registra detalhes do evento, incluindo *hosts*, interfaces, portas, quantidade de dados enviados e recebidos, regras do *firewall*, entre outros.

Para a realização deste trabalho foi selecionado um subconjunto de seis tipos de eventos: *statistic*, *access denied*, *connection fail*, *packet not enabled on interface*, *port not allowed* e *port scan*. A escolha desse subconjunto se deu em função da grande freqüência com que aparecem no *log* e de serem significativos no contexto da gerência de segurança.

3.2. Classificação e caracterização de eventos

Conforme já mencionado na Introdução, cada evento gerado por um *firewall* registra informações importantes como o tipo de operação, endereços origem e destino, portas local e remota, entre outras. Algumas dessas informações se repetem em mais de um tipo de evento. Assim, torna-se possível agrupá-los, usando um ou mais elementos de agregação. Esta constitui a idéia central da primeira parte da abordagem.

Ao agrupar eventos que compartilham informações comuns, torna-se possível realizar uma série de contabilizações para (a) mensurar e identificar acessos à rede privada e à externa, incluindo ações maliciosas (como varreduras de portas e tentativas de acesso a serviços não autorizados), (b) acompanhar sua evolução ao longo do tempo, (c) depurar problemas de configuração de regras de filtragem, entre outras. A figura 1 oferece vários exemplos nesse sentido; alguns deles são comentados a seguir.

Exemplo 1. Para determinar o total de dados enviados e recebidos para as conexões FTP é preciso agrupar os eventos que sejam do tipo estatístico (121) e que possuam o campo protocolo com o valor *ftp* (proto=ftp). Esse agrupamento resulta nos eventos 12 e 13. A contabilização dos totais desejados se dá pela soma dos valores associados aos campos *sent* e *rcvd* (resultando 41.731 e 577 bytes).

Exemplo 2. Inconsistências ou incorreções na configuração de regras de filtragem podem ser detectadas com agrupamento similar. Considere que a política de segurança da organização estabelece que o serviço FTP, disponível na estação 10.200.160.161, não deva ser acessado por *hosts* externos (fora da faixa 10.200.160.X). O agrupamento apresentado no

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

exemplo 1 realça dois eventos, 12 e 13, que confirmam o descumprimento de tal política, uma vez que os dois acessos são oriundos de estações com prefixo 66.66.77.X.

Exemplo 3. A identificação dos *hosts* de onde partiu o maior número de varreduras de portas é obtida com o agrupamento dos eventos do tipo 347, que resulta no sub-conjunto {1, 2, 3, 4, 5, 6, 7, 8, 9}. Desses eventos, quatro indicam varreduras partindo da estação 66.66.77.77 e cinco da estação 66.66.77.90.

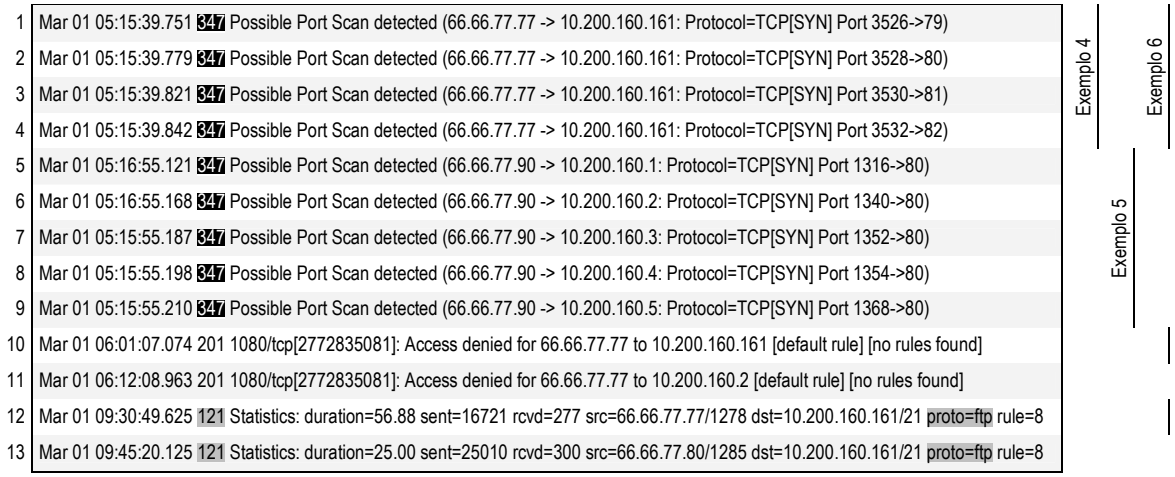


Figura 1. Conjunto real de eventos extraídos de um *log* e suas relações

Seguindo o mesmo raciocínio, outros elementos de agregação (ou uma combinação deles) podem ser empregados com o propósito de identificar, dentre as conexões feitas através do *firewall*, os máximos e mínimos em relação a protocolos utilizados, *hosts* que transmitiram dados entre as redes, *hosts* e portas acessadas, a quantidade de *hits* referente a eventos como varreduras de portas e acessos negados, as estações que mais sofreram e as que mais dispararam varreduras e as portas mais sondadas.

3.3. Análise automática de eventos

Além de uma análise mais quantitativa, onde diversas contabilizações são possíveis, a nossa abordagem permite a identificação automática de possíveis cenários de intrusão, que se realçam a partir da observação de um conjunto de eventos mais elementares. Na figura 1 podem ser destacados três comportamentos suspeitos, detalhados abaixo.

Exemplo 4. 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.

Exemplo 5. 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 6. 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 segunda parte da abordagem, detalhada nesta sub-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.

3.3.1. 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.

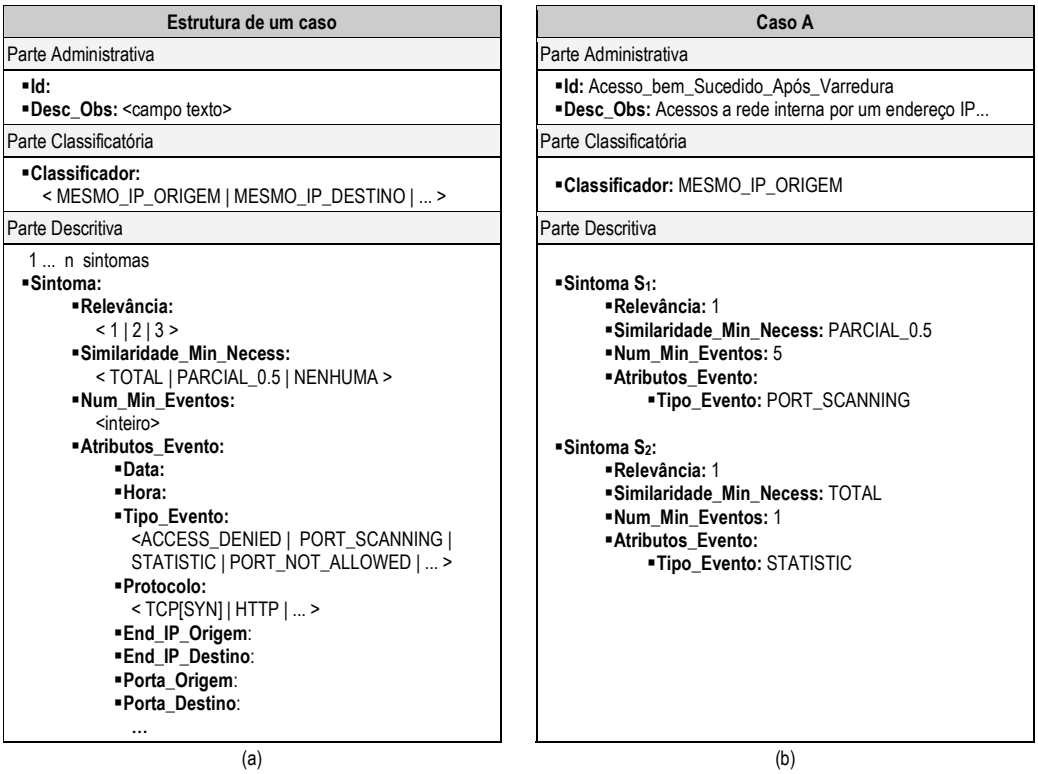


Figura 2. Modelagem de cenários de intrusão e atividades suspeitas como casos

3.3.2. Processos 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. Na versão atual, inicialmente foram modelados apenas tipos de similaridade de atributos que assumem casamento total (1) ou nenhum (0). 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 também apenas um atributo definido (tipo de evento). 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).

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 $((1 \times 0{,}8) + (1 \times 1))/2 = 0{,}9$, 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 (em nossa abordagem, 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 sub-seção, a atitude suspeita pode ser apresentada como *Acesso_bem_Sucedido_Após_Varredura* detectada para o endereço IP origem 66.66.77.77.

4. A Ferramenta SEFLA (Symantec Enterprise Firewall Log Analysis)

Para validar a abordagem apresentada na seção anterior foi desenvolvida a ferramenta SEFLA (Symantec Enterprise Firewall Log Analysis). Esta seção descreve a arquitetura e algumas das principais funcionalidades da ferramenta, incluindo os mecanismos de visualização de eventos empregados (baseados em *árvores hiperbólicas*).

4.1. Arquitetura

A ferramenta foi desenvolvida em ambiente Linux, utilizando as linguagens de programação Perl e PHP, o servidor Apache, o banco de dados MySQL e o *toolkit* Treebolic. A figura 3 ilustra a arquitetura de SEFLA, 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). A partir de um navegador *web* o gerente de segurança interage com o núcleo da ferramenta, que foi implementado em um conjunto de *scripts* PHP (3, 4). Essa interação permite (a) definir configurações de processamento (ex: tamanho dos históricos em dias e tipos de eventos a serem analisados), (b) recuperar relatórios, (c) realizar consultas e visualizar resultados, (d) observar alertas de cenários de intrusão ou atividades suspeitas e (e) verificar detalhes de eventos específicos. Para tal, a base de dados é sempre consultada ou atualizada (5).

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).

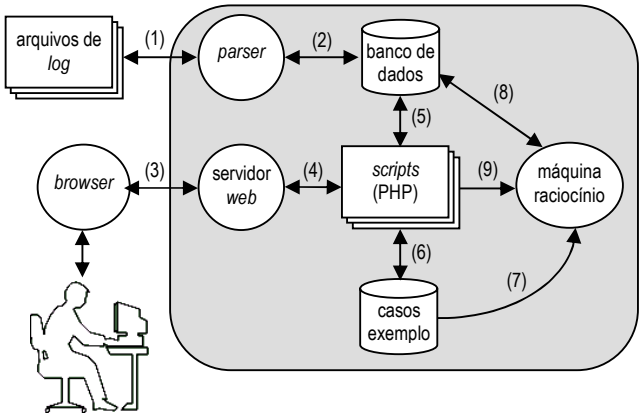


Figura 3. Componentes da arquitetura de SEFLA e suas relações

Através do navegador *web* o gerente de segurança também inclui, remove e atualiza casos na base de casos exemplo (3, 4, 6), conforme detalhado na sub-seção 3.3, e configura 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.

4.2. Principais funcionalidades

A figura 4a ilustra a interface gráfica da ferramenta, que está organizada em três partes principais: *Tool Configuration*, *Log Analysis Information* e *Event Correlation*. Em *Tool Configuration* é possível definir as configurações de processamento da ferramenta. *Log Analysis Information* provê funcionalidades ligadas à primeira parte da abordagem, descrita na sub-seção 3.2. As opções do menu secundário, à esquerda, são responsáveis por fornecer diversas visões sobre os eventos armazenados. Consultas podem ser realizadas por endereço origem ou destino, porta origem ou destino, protocolo, interface (ou uma combinação de atributos) sobre um mesmo tipo de evento (ou *statistic*, ou *access denied*, ou *port scanning*, etc). No centro da figura 4a é possível verificar o resultado de uma consulta que informa os *hosts* que mais realizaram acessos por entre as redes privada e externa no dia 28/09/2003. Já a figura 4b ilustra o número de eventos do tipo *access denied*, classificados por protocolo, observados no período de 28/09 a 04/10/2003.

Figura 4. Interface gráfica de SEFLA e suas funcionalidades

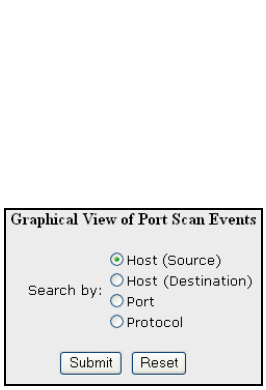
Em *Event Correlation* se encontram as funcionalidades relacionadas com a segunda parte da abordagem, detalhada na sub-seção 3.3. Para identificar, de forma interativa, cenários de intrusão como o ilustrado no exemplo 6, a ferramenta fornece uma interface de consulta que permite recuperar, mesclando 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 novas heurísticas (casos) que podem ser usadas pelo gerente de segurança para alimentar a base de casos exemplo, em um processo denominado *aprendizado*. Estão disponíveis também em *Event Correlation* recursos para visualizar os alarmes gerados sempre que cenários de intrusão são identificados.

4.3. Visualização de eventos

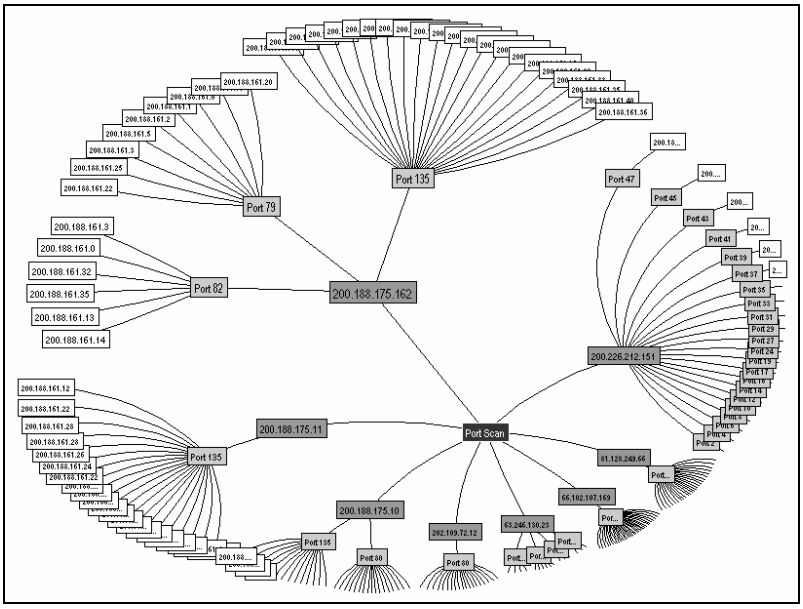
As consultas para recuperar eventos do *log* retornam, normalmente, uma quantidade expressiva de dados, o que dificulta uma melhor compreensão do resultado por meios tradicionais, como tabelas. Assim, buscou-se encontrar uma forma que permitisse não apenas a visualização desses dados, mas também uma maior exploração do usuário sobre a informação fornecida. A árvore hiperbólica apresentou-se como uma solução adequada [Lamping 1996]. Essa técnica representa hierarquias através de um *layout* radial definido no plano hiperbólico e depois mapeado para um disco 2D, possibilitando uma visão global das informações em um dado

contexto. Além disso, permite, por meio de mecanismos simples de navegação, uma mudança de perspectiva sobre os dados apresentados, podendo-se focar na região central da representação (árvore) as informações mais relevantes, enquanto o restante do diagrama tem seus nodos diminuídos de tamanho até serem suprimidos na borda do círculo.

A figura 5b ilustra a visualização resultante de uma agregação similar à realizada no exemplo 3, detalhada na sub-seção 3.2. Como é possível observar, a árvore representa um conjunto de eventos do tipo *port scan*, onde a raiz corresponde ao tipo de evento, os nós filhos são os endereços IP origem, no nível seguinte se encontram as portas destino e as folhas são os IPs destino. Vale salientar que essa representação gráfica destaca, muitas vezes, cenários de intrusão. Neste exemplo, identifica-se visualmente varreduras verticais e horizontais (mencionadas nos exemplos 4 e 5 da sub-seção 3.3).



(a)



(b)

Figura 5. Consulta por varreduras de portas e árvore hiperbólica resultante

5. Estudo de Caso

Como estudo de caso foi utilizada a rede acadêmica da Universidade do Vale do Rio dos Sinos, cuja infraestrutura apresenta aproximadamente 4.100 computadores conectados à rede e com acesso à Internet. A partir do *firewall* que se encontra na borda dessa rede, foram coletados arquivos de *log* durante o período de uma semana, de domingo a sábado. A ferramenta SEFLA foi populada com esses *logs* e através da análise dos relatórios obtidos foi possível classificar, caracterizar e analisar os eventos visando determinar o uso da rede e identificar cenários de intrusão e atividades suspeitas. A ferramenta foi instalada em uma estação IBM Pentium 4 modelo NetVista, processador de 1.8 GHz, 256 MB de memória RAM, 512 de memória *cache*, sistema operacional Linux distribuição Red Hat Linux 9.0, *kernel* 2.4.20.

A tabela 1 descreve o perfil de cada *log* e as características de processamento dos mesmos. Os maiores *logs* são os gerados entre a segunda e a sexta-feira. Dado o somatório do tamanho de todos os arquivos de *log* (13,05 GB) e considerando que desse volume foram processados 52,2% dos eventos (que correspondem aos seis tipos de eventos listados na sub-seção 3.1), pode-se verificar que o tamanho do *log* foi bastante reduzido ao ser inserido na base

de dados (resultou 4,5 vezes menor em relação ao original). Além disso, o tempo gasto para processar os 13,05 GB de *log* foi de 144,5 minutos (2 horas, 24 minutos e 30 segundos).

Tabela 1. Características de processamento dos arquivos de *log*

Data do *log*	Tamanho do *log* (em GB)	Eventos processados (em milhões)	Tempo de processamento (em minutos)	Tamanho acumulado da base de dados (em GB)
28/09/2003	0,69	1,30	8,1	0,15
29/09/2003	2,53	3,84	28,7	0,72
30/09/2003	2,55	3,79	28,4	1,27
01/10/2003	2,37	3,52	24,0	1,82
02/10/2003	2,28	3,58	23,6	2,31
03/10/2003	1,93	3,10	22,6	2,75
04/10/2003	0,70	1,40	9,1	2,92
Totais	**13,05**	**20,53**	**144,5**	**2,92**

A figura 6 ilustra algumas descobertas, mais quantitativas, realizadas com o apoio da ferramenta SEFLA. Em (a) é apresentado o fluxo de dados entre a rede privada e a externa. Como é possível observar, o protocolo HTTP é o mais utilizado, seguido de TCP/1500 (utilizado por ferramenta de *backup*), FTP, SMTP e HTTPS. O total de *bytes* transferidos entre as redes, de pelo menos 30 GB de segunda a sexta-feira, é outra informação que merece ser destacada. No que tange as varreduras de portas, foram analisados os dados do dia em que elas mais ocorreram, ou seja, domingo (figura 6b). As cinco estações de onde partiu o maior volume de sondagens possuem o mesmo prefixo (200.188.175.X). Em identificando esse comportamento hostil, requisições partindo dessa rede deveriam ser cuidadosamente analisadas (ou bloqueadas pelo *firewall*). A figura 6c, em contrapartida, destaca as estações que mais foram alvo de varreduras ao longo da semana em análise. Ainda na análise de varreduras de portas, a figura 6d ilustra o histórico da porta mais sondada. Segundo o estudo realizado sobre os *logs*, a porta destino 135 representou 90% do total de sondagens no período de sete dias. A porta 135 é geralmente usada pela plataforma Windows para iniciar uma conexão RPC (*Remote Procedure Call*) com um computador remoto. As varreduras ilustradas são provavelmente decorrentes dos vírus W32.Blaster.Worm e W32.Welchia.Worm descobertos, respectivamente, em 11/08/2003 e 18/08/2003. Esses *worms* são caracterizados por explorar a vulnerabilidade RPC do DCOM (*Distributed Component Object Model*) do Microsoft Windows agindo através da porta TCP 135 para efetuar ataques de DoS (*Denial of Service*) [Symantec 2003b].

Além da análise descrita acima, os eventos coletados pelo *firewall* ao longo da semana também puderam ser analisados do ponto de vista da identificação automática de cenários de intrusão. Um dos cenários identificados foi o de varreduras de portas (com comportamento similar ao observado nos exemplos 4 e 5), que se repetiu com alta freqüência nos eventos do *log*. Uma instância desse cenário corresponde à varredura representada por 24 eventos do tipo *port scan* de um mesmo IP origem (200.226.212.151) para um mesmo IP destino (200.188.160.130), observados a uma hora e quarenta e oito minutos de domingo. Esse cenário foi identificado na abordagem como similar ao caso *Varredura*, com similaridade de 100%, já que a ocorrência envolveu mais de cinco eventos do tipo *port scan* oriundos de um mesmo endereço IP origem (sintoma estabelecido para o caso). Outro cenário reconhecido em diversos momentos foi o que apresenta varreduras seguidas de acesso bem sucedido partindo da mesma estação origem, representado no caso *Acesso_bem_Sucedido_Após_Varredura* (figura 2b). É importante salientar que, para o conjunto reduzido de casos modelados, a abordagem se mostrou eficaz na identificação de todas as instâncias dos mesmos junto ao *log* utilizado como fonte de dados. Estamos ampliando, atualmente, a base de casos para poder avaliar a generalidade da abordagem proposta.

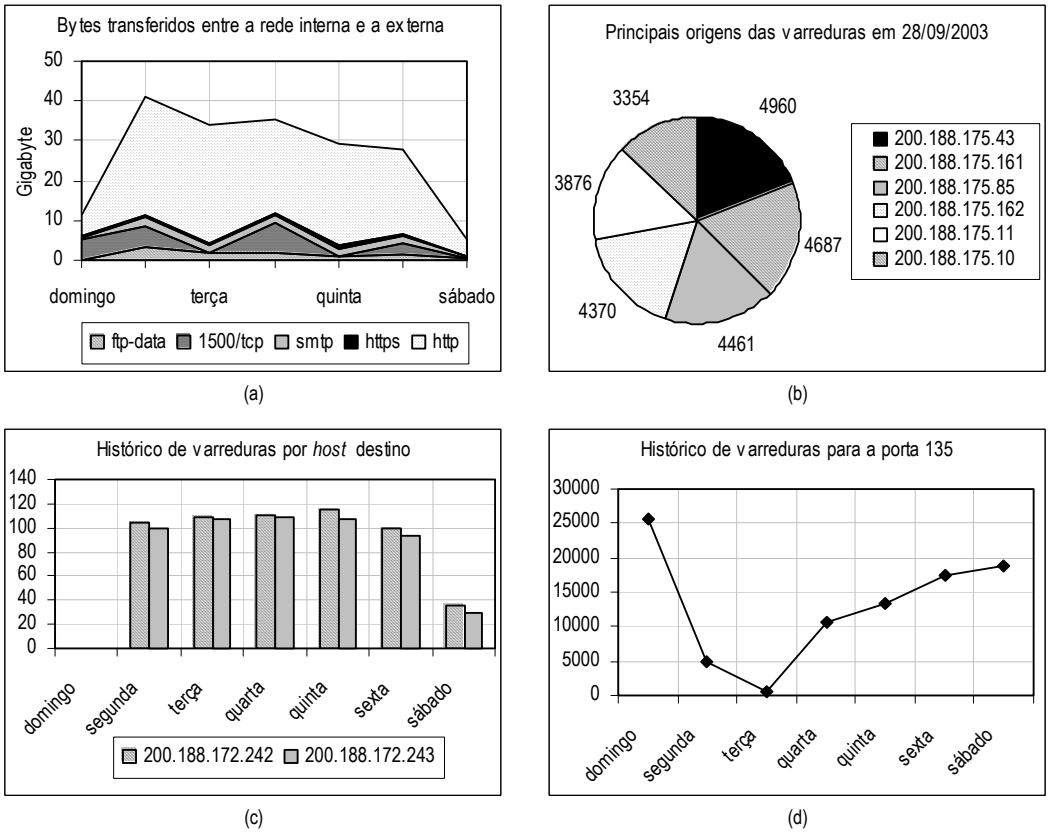


Figura 6. Algumas informações recuperadas com a utilização de SEFLA

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. Nesse contexto, apresentamos uma abordagem, acompanhada de uma ferramenta, para classificação, caracterização e análise de eventos gerados por *firewalls*. Vale salientar que a nossa abordagem não substitui outros sistemas, como os de detecção de intrusão, devendo ser utilizada em conjunto com os mesmos.

A organização da abordagem em duas partes permite manipular, de forma satisfatória, tanto informações quantitativas, quanto qualitativas. Por um lado, o mecanismo de agrupamento de eventos com base em um ou mais elementos de agregação revela características de uso da rede e de atividades maliciosas. Essas podem ser usadas para avaliar o cumprimento da política de segurança, policiar o uso de recursos (revendo regras de filtragem vigentes) e reconhecer origens e alvos de comportamentos hostis (visando sua proteção). Por outro lado, a segunda parte da abordagem – apoiada na técnica de raciocínio baseado em casos – provê o reconhecimento automático de seqüências de eventos que representam cenários de intrusão ou atividades suspeitas. Aqui, mais do que identificar e quantificar ações, se procura realçar as estratégias adotadas por invasores para obter acesso indevido a estações, serviços e aplicações.

O mecanismo de visualização incorporado à ferramenta é outro ponto a ser destacado. As consultas realizadas ao sistema, em geral, resultam grandes volumes de informações, cuja visualização ficaria bastante comprometida se fossem dispostos em tabelas convencionais. As árvores hiperbólicas, nesse contexto, mostraram-se adequadas principalmente para enfatizar pontos de convergência e divergência dos eventos em relação aos atributos comuns. A investigação de mecanismos adicionais de visualização, em especial para ilustrar a matriz de comunicações estabelecidas, constitui um dos próximos trabalhos a serem realizados.

Como pôde ser observado na seção 5, mesmo após o processamento e armazenamento dos eventos no banco de dados, o tamanho da base resultante é elevado (considerando que ela registra os eventos de apenas sete dias). Para que se possa obter estatísticas de longo prazo, propõe-se como um dos trabalhos futuros a síntese de informações essenciais sobre os eventos mais antigos em uma base reduzida (com o custo de se perder a possibilidade de detalhamento desses eventos).

Pretende-se também aprimorar a abordagem de raciocínio baseado em casos modelando novos tipos de similaridade de atributos que admitam similaridade parcial, a fim de aumentar a flexibilidade no casamento entre casos. Outro aspecto a ser tratado é a inclusão de mecanismos na abordagem que permitam modelar diferenças e relações para os vários eventos que podem fazer parte de um sintoma do caso.

Referências

CheckPoint (2003) “FireWall-1: The Most Comprehensive Network Security Platform on the Market”, http://www.checkpoint.com, February.

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.

ISS (2003) “RealSource Guard”, http://www.iss.net, March.

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

Lamping J. and Rao, R. (1996) “The Hyperbolic Browser: a Focus+Context Technique for Visualizing Large Hierarchies”, In: Journal of Visual Languages and Computing, v. 7, n. 1, p. 33-55.

Netfilter (2003) “Firewalling, NAT and Packet Mangling for Linux 2.4”, http://www.iptables.org, August.

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.

Secure Computing (2003) “Gauntlet Firewall”, http://www.securecomputing.com, November.

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., Network Security Essentials: Applications and Standards, Prentice-Hall, 2000.

Symantec (2003a) “Symantec Enterprise Firewall 7.0”, http://enterprisesecurity.symantec.com, February.

Symantec (2003b) “Symantec Security Response”, http://www.symantec.com.br, November.

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

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

Wankwood (2003) “Reptor”, http://www.wankwood.com/reptor, February.

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.