Aplicando a Técnica de Raciocínio Baseado em Casos na Identificação de Cenários de Intrusão em *Logs* de *Firewalls*

Samir Lohmann, Cristina Melchiors, Luciano Paschoal Gaspary

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

samir.lohmann@ska.com.br, {cmelch, paschoal}@exatas.unisinos.br

Resumo. *As ferramentas de análise de logs de firewalls existentes atualmente são muito úteis na obtenção de certos diagnósticos de problemas de redes corporativas. Entretanto, falta a elas pro-atividade para que encontrem cenários de intrusão automaticamente. Este artigo apresenta os progressos feitos até o momento no desenvolvimento de um módulo para uma destas ferramentas (SEFLA) que, através da técnica chamada Raciocínio Baseado em Casos, analisa os eventos gerados por um firewall e identifica cenários de intrusão de forma automática. Espera-se também que tal módulo auxilie na correta configuração de regras de filtragem de firewalls.*

1. Introdução

Do ponto de vista da gerência de segurança, os *logs* de *firewalls* são ricos em informações, pois através deles pode-se identificar (a) suspeitas de ataques, (b) os serviços mais e os menos requisitados, (c) os *hosts* que ocupam mais e menos banda, (d) os principais usuários e (e) eventuais anomalias [Chapman e Zwicky, 1995; Symantec, 2001; Taylor, 2002].

Existem diversas ferramentas de análise de *logs* que permitem classificar, caracterizar, armazenar históricos e visualizar de forma amigável os eventos gerados por *firewalls* [Locatelli, 2003] Dentre elas, pode-se citar o Reptor [Wankwood, 2003] e o SEFLA [Locatelli, 2003]. Tais ferramentas possuem diferentes funcionalidades, mas as estatísticas e os eventos apresentados são meros espelhos do *log*. Embora sejam de grande valia para obter alguns diagnósticos, falta a elas pro-atividade, já que a identificação de cenários de intrusão é feita pelo usuário, através da análise manual dos eventos. Esta abordagem é pouco eficiente na detecção de comportamentos suspeitos, pois exige monitoramento constante por parte do gerente de segurança, sem o qual cenários de intrusão são simplesmente ignorados. Isto acontece devido ao grande volume de eventos gerados diariamente pelo *firewall*, e também porque se depende exclusivamente do usuário para detectar cenários de intrusão.

O objetivo deste trabalho é desenvolver um módulo para a ferramenta SEFLA que, através da técnica de Inteligência Artificial denominada Raciocínio Baseado em Casos (CBR – *Case-based Reasoning*) [Kolodner, 1993], analise os eventos gerados pelo *firewall* Symantec Enterprise e identifique cenários de intrusão de forma automática. Apesar de não ser um objetivo primário, espera-se também que a ferramenta seja capaz de auxiliar na correta configuração de regras de filtragem de *firewalls*.

Os estudos realizados até o momento apontam CBR como uma técnica adequada, pois tem a capacidade de identificar similaridade entre situações novas e situações já conhecidas armazenadas em sua base de conhecimento. Tal base armazena as situações antigas 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 [Locatelli et al, 2004].

O artigo está organizado da seguinte forma: a seção 2 explica os fundamentos de CBR e descreve uma proposta de modelagem para casos de intrusão. A arquitetura da ferramenta SEFLA com o novo módulo proposto é introduzida na seção 3. Por fim, na seção 4, são feitas considerações sobre problemas encontrados até agora e previstos para o futuro.

2. Modelagem de Cenários de Intrusão através de Casos

Um caso é uma representação em baixo nível de um problema da vida real. As fases utilizadas na solução de problemas em um CBR típico são o casamento e a adaptação. O casamento é a fase na qual as novas situações que chegam ao sistema (doravante denominadas casos correntes) são comparadas aos casos armazenados no sistema. Se nesta fase for encontrado algum caso armazenado que possua uma similaridade mínima com um caso corrente, parte-se para a adaptação, onde a solução do caso armazenado é adaptada ao caso corrente e, em seguida, avaliada. Atualmente, o protótipo não conta com as fases de adaptação e avaliação.

Parte Administrativa		
Descrição	**Descrição do Caso de Intrusão**	
Gravidade	[1 , 2 , 3]	
Parte Classificatória		
Classificação	[Mesmo_IP_Origem , Mesmo_IP_Destino , Mesma_Porta_Destino , Diferentes_Portas_Destino , Intervalo_de_x_horas ...]	
Parte Descritiva		
Sintoma n		
Relevância	[1 , 2 , 3]	
Similaridade_Mínima	[1 , 0.5 , 0]	
Número_Min_Eventos		
Atributos do Evento		
Nome	Operador	Valor
Nome_do_Atributo	[Igual , Diferente , Maior_que , Menor_que , ...]	Valor_do_Atributo
Outros Atributos		

Figura 1 – Estrutura de um caso de intrusão

A figura 1 ilustra a estrutura de um caso. Cada caso é composto por uma parte administrativa, que possui a descrição do caso e a gravidade do mesmo, que pode variar de um até três (do menos para o mais grave). A parte classificatória especifica quais são os atributos utilizados no agrupamento de eventos para formar os casos correntes. O agrupamento é explicado na seção 3. Por fim, a parte descritiva contém os sintomas do caso. Cada sintoma tem, obrigatoriamente, uma relevância (de um a três, do menos para o mais relevante), a similaridade mínima necessária (total, parcial ou nenhuma) e o número mínimo de eventos necessários que um sintoma de um caso corrente deve ter para se dizer que equivale ao sintoma do caso armazenado. Além desses atributos obrigatórios, há uma série de atributos opcionais, preenchidos de acordo com cada sintoma. Para cada atributo, pode-se especificar um operador (Igual, Diferente, Maior_que, Menor_que, Intervalo) e um valor, que pode ser fixo ou então uma máscara. No caso do atributo Hora, por exemplo, pode-se especificar, além de um valor fixo (como 12:00), uma máscara como "Madrugada: 00:00 até 6:00". Certos operadores só poderão ser utilizados em atributos cujo tipo de dados permita tal operador (ex: Maior_que só pode ser usado em campos numéricos).

Parte Administrativa		
Descrição	**Acesso bem-sucedido após varredura**	
Gravidade	3	
Parte Classificatória		
Classificação	Mesmo_IP_Origem	
Parte Descritiva		
Sintoma 1		
Relevância	1	
Similaridade_Mínima	1	
Número_Min_Eventos	1	
Atributos do Evento		
Nome	Operador	Valor
Tipo	Igual	PORT_SCANNING
Sintoma 2		
Relevância	1	
Similaridade_Mínima	1	
Número_Min_Eventos	1	
Atributos do Evento		
Nome	Operador	Valor
Tipo	Igual	STATISTIC

Figura 2 – Acesso bem-sucedido após varredura

Parte Administrativa		
Descrição	**Upload suspeito**	
Gravidade	2	
Parte Classificatória		
Classificação	Mesmo_IP_Origem	
Parte Descritiva		
Sintoma 1		
Relevância	1	
Similaridade_Mínima	0.5	
Número_Min_Eventos	3	
Atributos do Evento		
Nome	Operador	Valor
Tipo	Igual	ACCESS_DENIED
Sintoma 2		
Relevância	1	
Similaridade_Mínima	1	
Número_Min_Eventos	1	
Atributos do Evento		
Nome	Operador	Valor
Tipo	Igual	STATISTIC
Protocolo	Igual	FTP
Bytes Enviados	Maior_que	Bytes Recebidos

Figura 3 – Upload Suspeito

Nas figuras 2 e 3, dois exemplos de casos são apresentados. No caso ilustrado na figura 2, um usuário executou uma varredura de portas e logo depois conseguiu acesso a algum computador da rede, já que foi gravado um evento estatístico (uma conexão foi realizada com sucesso). Já no caso da figura 3, o usuário obteve alguns acessos negados (o que pode indicar que está tentando explorar senhas fracas) e depois fez um *upload* (já que teve uma conexão para protocolo FTP e enviou mais dados do que recebeu). Isto indica que este usuário pode estar transferindo um arquivo com código malicioso para depois executá-lo.

3. Arquitetura da Ferramenta

SEFLA é uma ferramenta que possibilita estruturar arquivos de *log* do *firewall* Symantec Enterprise (SEF) alimentando, através de um *parser*, um banco de dados MySQL. Como este banco de dados é um arquivo estruturado (ao contrário do *log* original), é possível classificar, caracterizar, armazenar históricos e visualizar de forma amigável os eventos gerados pelo SEF. A interface com o usuário é feita através de *scripts* PHP hospedados em um servidor *web*, que executa os *scripts* e envia para o usuário páginas HTML. A figura 4 ilustra a arquitetura atual de SEFLA.

O módulo apresentado neste trabalho utiliza a mesma base de dados, porém possui um repositório de casos de intrusão conhecidos, além de um mecanismo de agrupamento, recuperação e seleção, que identifica cenários de intrusão e gera alarmes para o gerente de segurança, quando encontrar casos suspeitos. Os novos componentes são mostrados em destaque na figura 4. Os casos armazenados são criados e mantidos pelo gerente da rede, de acordo com o perfil de uso da rede.

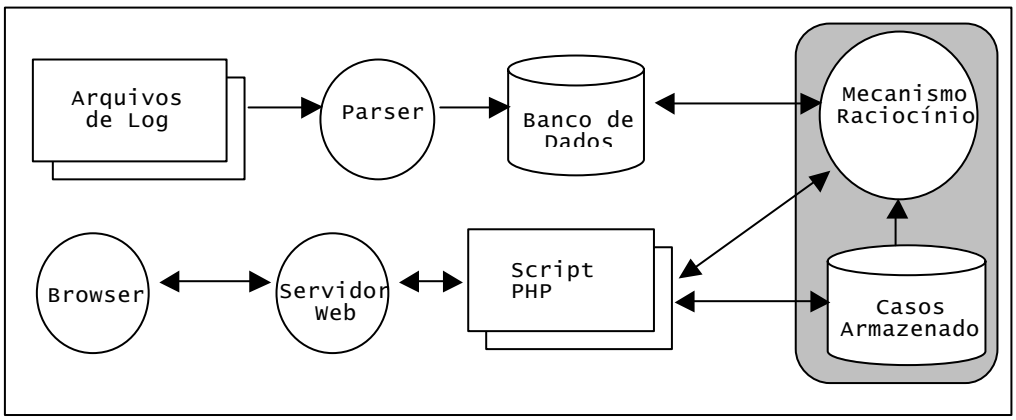

Figura 4 – Arquitetura proposta

Há uma diferença fundamental entre os sistemas CBR citados na literatura e o proposto neste trabalho. Em um CBR tradicional, os problemas chegam prontos de um ambiente externo. Como exemplo, pode-se citar solicitações de suporte a *helpdesk*, e IDSs baseados em CBR, onde cada pacote que chega à rede é analisado. O protótipo proposto não recebe problemas prontos. É necessário, primeiramente, agrupar os eventos do *log* do *firewall*. O critério de agrupamento é obtido a partir do classificador dos próprios casos armazenados, o que faz com que cada caso corrente seja comparado com todos os casos armazenados. O produto desse agrupamento é um conjunto de casos correntes. As fases do CBR aqui proposto são:

a. Agrupamento: a partir dos atributos do classificador de cada caso armazenado, todos os eventos do *log* são agrupados, e cada grupo corresponde a um caso corrente. Por exemplo, se o classificador for Mesmo_IP_Origem, todos os eventos cujo Endereço_IP_Origem for 200.244.90.5 formam um caso corrente.

b. Recuperação: os casos correntes obtidos no passo anterior serão comparados com todos os casos cujo classificador for o mesmo pelo qual foram agrupados. Em seguida, a similaridade com cada um destes casos é calculada. A comparação e o cálculo são realizados através do algoritmo proposto em [Locatelli et al, 2004].

c. Seleção: Se no passo anterior foram encontrados um ou mais casos correntes que alcançaram ou ultrapassaram a similaridade mínima necessária de algum caso armazenado, o caso corrente com maior similaridade será selecionado e um alerta será gerado ao administrador da rede.

4. Considerações

Com base em estudos preliminares e não conclusivos realizados até o momento, foram percebidos alguns problemas que precisam ser tratados, seja neste trabalho, seja por trabalhos futuros. Em primeiro lugar, o algoritmo necessário para atingir os objetivos aqui propostos é caro computacionalmente, o que é agravado devido ao tamanho dos *logs* analisados, que pode chegar a vários *gigabytes*. Apesar do protótipo não ter compromisso com desempenho (pois não é um IDS e sim, um sistema de apoio ao gerente de segurança), espera-se que o processamento dos *logs* aconteça no menor tempo possível para que os dados analisados sejam tão recentes quanto possível.

Em segundo lugar, os estudos realizados até o momento indicam que determinadas ações de atacantes não são registradas pelo *firewall*. Como exemplo, pode-se citar alguns tipos de varreduras camufladas, as quais não chegam a estabelecer uma conexão TCP. Como um trabalho futuro, sugere-se uma abordagem mista, na qual não só os *logs* do *firewall* são analisados, mas também dados de outras fontes, como *logs* de Sistemas de Detecção de Intrusão (IDS).

Referências

Chapman, D. B., Zwicky, E. D. (1995) "Building Internet Firewalls", O'Reilly & Associates.

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

Locatelli, F. E. (2003) "Uma Ferramenta baseada na Análise de Logs para Classificação, Caracterização e Correlação de Eventos Gerados pelo Symantec Enterprise Firewall", Trabalho de Conclusão de Graduação, Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos (UNISINOS).

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: XXII Simpósio Brasileiro de Redes de Computadores, Gramado, p. 851-864.

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

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

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