Uma Abordagem para Detecção de Ataques Distribuídos e de Múltiplas Etapas baseada na Composição de Serviços *Web* voltados à Segurança

Leonardo Lemes Fagundes, Luciano Paschoal Gaspary

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

llemes@unisinos.br, paschoal@exatas.unisinos.br

Abstract. *This paper presents an architecture based on web services for distributed, multi-stage attack detection. The architecture provides a uniform mechanism to communicate with different security services and allows events generated by these services to be correlated, as well as countermeasures to be executed.*

Resumo. *Este artigo apresenta uma arquitetura baseada em web services para detecção de ataques distribuídos e de múltiplas etapas. A arquitetura oferece um mecanismo uniforme de comunicação com diferentes serviços de segurança e possibilita a correlação de eventos gerados pelos mesmos, bem como a execução de ações de contramedidas.*

1. Introdução

Entre os ataques a redes de computadores mais destrutivos e difíceis de detectar estão aqueles que ocorrem em diversas etapas, realizados de forma distribuída, ao longo de um intervalo de tempo. Essas etapas ou ataques intermediários possuem objetivos específicos e, em alguns casos, podem ser realizadas de diferentes maneiras e em uma ordem temporal qualquer [Cheung et al., 2003]. A natureza distribuída e, muitas vezes, concorrente desses ataques, aliada à possibilidade de execução de uma mesma etapa de diferentes maneiras, fazem com que a sua detecção represente uma tarefa bastante complexa.

No intuito de desenvolver soluções capazes de minimizar as chances de um intruso obter sucesso em suas atividades, muitos esforços já foram realizados, sobretudo na área de detecção de intrusão. Essa área tem sido foco de uma grande variedade de recentes projetos de pesquisa [Cuppens, 2002; Debar, 2001]. Entretanto, algumas limitações ainda permanecem, entre elas: (*i*) a inexistência de um mecanismo para representação de cenários de ataque que permita especificar um fluxo de eventos (grafo) que compõem uma intrusão, (*ii*) a falta de uma maneira uniforme e precisa para integrar, independentemente do domínio administrativo, diferentes serviços de segurança (exemplo: sistemas de detecção de intrusão e analisadores de *logs*) com ferramentas de correlação de alertas e (*iii*) a ausência de mecanismos para execução de contramedidas que dispensem a intervenção direta de um profissional.

Este trabalho tem por objetivo projetar, implementar e avaliar uma arquitetura de natureza distribuída que (*i*) ofereça um mecanismo uniforme de comunicação com diferentes serviços de segurança, (*ii*) possibilite a subscrição por eventos de interesse junto a esses serviços e (*iii*) permita a correlação desses eventos, visando à detecção de ataques de múltiplas etapas.

Os benefícios esperados com o uso dessa arquitetura desdobram-se em três. O primeiro benefício constitui-se na possibilidade de definir, de maneira flexível, os cenários de ataques e de uso incorreto dos recursos disponíveis. O segundo benefício se refere à viabilidade de identificar, prematuramente, invasões que estejam em curso. Por fim, o terceiro benefício diz respeito a execução automática de procedimentos de contenção em relação às ações indesejáveis, assim que as mesmas forem detectadas.

A arquitetura proposta se baseia na utilização de uma especificação denominada *Web Services Notification*, cuja função é padronizar uma abordagem para que serviços *web* implementem a notificação de eventos utilizando o padrão *publish/subscribe* baseado em tópicos [Graham et al., 2004].

2. Arquitetura Proposta

A arquitetura proposta é composta pelos seguintes componentes: estação de gerenciamento, serviços de notificação, motor de detecção e serviços de contenção. A Figura 1 ilustra um esquema da arquitetura e as interações realizadas entre os componentes que fazem parte da mesma.

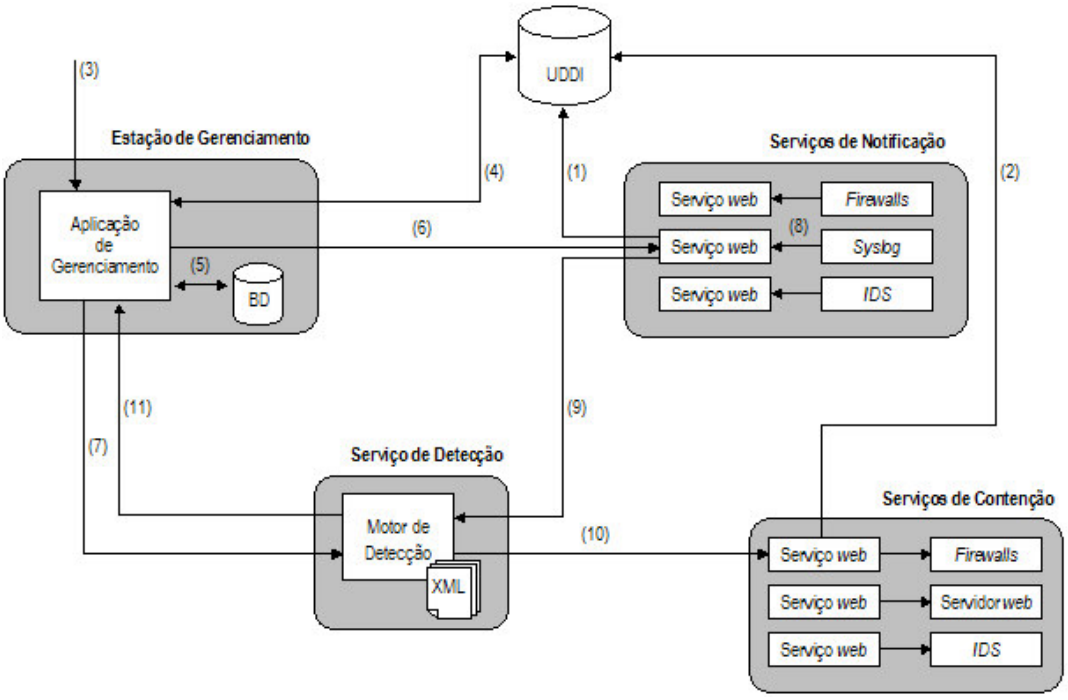

Figura 1. Visão geral da arquitetura

A análise da seqüência das interações realizadas entre os componentes dessa arquitetura resulta em um melhor entendimento sobre o funcionamento e o dinamismo da mesma. Inicialmente, os serviços de notificação e contenção (*scripts* capazes, por

exemplo, de remover usuários, bloquear acesso oriundos de determinados endereços e usuários) são publicados (fluxos 1 e 2 da figura) em um repositório de serviços *web*. Em seguida, o gerente de rede estabelece, via *browser*, uma conexão (3) com a estação de gerenciamento.

A estação de gerenciamento permite realizar consultas a repositórios de serviços (4) com o objetivo de se obter uma lista dos serviços de notificação e contenção de ataques disponíveis. As informações fornecidas por esses serviços são utilizadas na especificação dos cenários de intrusão que, posteriormente, são armazenados em uma base de dados local (5). A partir de um cenário de intrusão é gerada uma subscrição de tópicos (6) aos serviços de notificação. Outra tarefa importante diz respeito à instanciação das especificações (7), ou seja, a indicação (ao serviço de detecção) de quais cenários devem ser monitorados.

Assim que um serviço de notificação recebe uma subscrição vinda da estação de gerenciamento, se inicia o processo de monitoração dos tópicos de interesse. Tão logo seja identificada a ocorrência de um desses tópicos (8) o serviço de detecção é comunicado (9). Para cada novo evento recebido, o motor de detecção consulta os cenários de ataque monitorados (documentos em formato XML) na tentativa de detectar intrusões em andamento. Como resultado das consultas realizadas, podem ser executados os serviços de contenção (10) e enviados alertas (11) à estação de gerenciamento. Os alertas gerados seguem o *Intrusion Detection Message Exchange Format* (IDMEF) [Debar et al., 2004].

Conforme mencionado no parágrafo anterior, o motor de detecção monitora os cenários de ataques. Esses cenários devem ser especificados de forma a contemplar as diversas peculiaridades dos ataques de múltiplas etapas. Linguagens como STATL [Eckmann, 2002] e CAML [Cheung et al., 2003] pecam ao não oferecer um mecanismo para representação de informações obtidas a partir de diferentes fontes (heterogeneidade) e uma maneira ágil e ao mesmo tempo eficiente para modelar cenários de intrusões (flexibilidade). Com o objetivo de atender os requisitos supracitados propõe-se uma linguagem para definição de ataques de múltiplas etapas.

Essa linguagem está em fase de definição e será composta por duas notações, uma gráfica e outra textual. Os elementos básicos da especificação de um cenário de ataque são os estados e as transições. Os estados indicam o estágio atual do ataque e são representados por círculos. As transições são eventos que fazem com que a máquina de estados evolua de estado. Os eventos podem ser notificações (linha contínua) ou ações (linha tracejada).

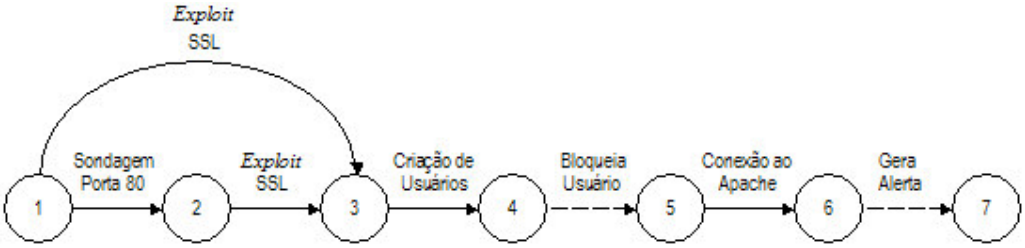

Figura 2. Representação gráfica de um cenário de ataque

A Figura 2 ilustra um exemplo de cenário de ataque a servidores *web* Apache usando a notação proposta. Esse ataque ocorre basicamente da seguinte forma. Um atacante executa um *exploit* (que explora um *buffer overflow* em um módulo SSL) e obtém acesso ao *Shell* de um servidor *web* Apache. A partir daí é criado um usuário para garantir ao invasor acesso a este servidor, mesmo após a correção da vulnerabilidade.

Para exemplificar o funcionamento do processo de detecção, assuma que a partir da estação de gerenciamento foram realizadas as seguintes subscrições: "Sondagem Porta 80" e "*Exploit* SSL" encaminhadas a um sistema de detecção de intrusão baseado em rede (ex.: o Snort), "Criação de Usuário" subscrição enviada à um sistema de detecção de intrusão baseado em *host* (ex.: o Tripwire) e, "Conexão ao Apache" subscrição feita à um serviço de análise de *logs* do Apache. Supondo que os três primeiros eventos subscritos já tenham sido notificados, a máquina de estados se encontra no estado 4 a partir do qual é realizada uma contramedida (Bloqueia Usuário) na tentativa de conter esse ataque. Caso, ainda assim, seja notificada uma conexão ao servidor Apache é gerado um alerta e alcançar-se-á o estado 7.

3. Considerações Parciais

Este artigo apresentou uma arquitetura para detecção de ataques distribuídos e de múltiplas etapas. No estágio atual a linguagem proposta para representação dos cenários de ataques está em fase final de especificação. Logo em seguida, iniciará a etapa de prototipação e aperfeiçoamento da arquitetura. O sistema de comunicação será baseado no *Pubscribe*[1], uma implementação em Java da especificação *Web Services Notification*.

Uma vez que o protótipo esteja finalizado, serão realizados testes para avaliar a capacidade da linguagem empregada na representação dos cenários de ataque e o desempenho da arquitetura. A etapa de avaliação será realizada sobre um ambiente a partir do qual serão reproduzidos diversos tipos de ataques, entre eles: negação de serviço distribuídos, tentativas de acesso indevido e ataques a servidores *web*.

4. Referências

Cuppens, F. and Miége, A. (2002) "Alert Correlation in a Cooperative Intrusion Detection Framework", Proceedings in IEEE Symposium on Security and Privacy, pp 187 – 200.

Cheung, S., Lindqvist, U. and Fong, W. M. (2003) "Modeling Multistep Cyber Attacks for Scenario Recognition", DARPA Information Survivability Conference and Exposition (DISCEX III), pp. 284 – 292.

Debar, H. and Wespi, A. (2001) "Aggregation and Correlation of Intrusion-Detection Alerts", Lecture Notes in Computer Science, Proceedings RAID, pp. 85 – 103.

Debar, H., Curry, D. and Feinstein, B. (2004) "The Intrusion Detection Message Exchange", IETF Intrusion Detection Exchange Format Working Group, Internet Draft.

Eckmann, T. S., Vigna, G., Kemmerer, A. R. (2002) "STATL: An Attack Language for State-based Intrusion Detection", Journal of Computer Security, vol. 10, n°. 2, pp. 71-104.

Graham, S., Nib lett, P., Chappell, D. (2004) Web Services Notification. Online: http://www.oasis-open.org/committees/documents.php?wg_abbrev=wsn (junho de 2005).

[1] Documentação disponível em: http://ws.apache.org/