Criação e Gerenciamento de Composições de IDSs*

José Eduardo M. S. Brandão[1,2], Joni da Silva Fraga[1], Paulo Manoel Mafra[1]

[1]Laboratório de Controle e Micro Informática (LCMI)
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476, CEP 88040-900, Florianópolis - SC

[2]Instituto de Pesquisa Econômica Aplicada (IPEA)
SBS Q.1, Brasília - DF

{jemsb, fraga, mafra}@das.ufsc.br

Abstract. *This paper presents a new approach for building compositions of Intrusion Detection Systems (IDSs), based on the concept of services orchestration. This approach allows the creation of more cooperative, flexible and adequate IDSs to distributed systems, mainly those formed by wide-scale networks like Internet. IDSs and their components are combined using a service-oriented architecture model based on the Web Services technology. In order to provide the necessary interoperability among the elements of these composed systems, standards efforts, mainly those developed by IETF, W3C and OASIS are used. The paper introduces a services infrastructure that provides support for the creation and the management of IDS compositions.*

Resumo. *Este artigo apresenta uma nova abordagem para a construção de composições de Sistemas de Detecção de Intrusão (IDSs) baseada no conceito de orquestração de serviços. Esta abordagem permite a construção de IDSs mais cooperativos, flexíveis e adequados para ambientes heterogêneos, sobretudo aqueles formados por sistemas de larga escala que fazem uso da Internet. Os IDSs e seus componentes são combinados utilizando a arquitetura orientada a serviço, suportada pela tecnologia de Web Services. A interoperabilidade entre os diversos elementos de uma composição é obtida a partir do amplo emprego de esforços de padronização, sobretudo da IETF, W3C e OASIS. Este documento descreve uma infraestrutura de serviços e suportes proposta para a criação e operação destas composições de IDSs, focando na criação e gerenciamento das composições.*

1. Introdução

Os sistemas responsáveis pela execução de aplicações distribuídas não estão mais limitados ao escopo de uma rede ou um domínio administrativo único. Estes ambientes de larga escala formam o que é usualmente identificado como organizações virtuais. Dificilmente modelos tradicionais de sistema de detecção de intrusão (*IDS – Intrusion Detection System*) [Bace and Mell 2001] são capazes de desempenhar o monitoramento nestes ambientes de larga escala, envolvidos com uma grande diversidade de plataformas e aplicações. A evolução destes ambientes impõe a adaptação freqüente de IDSs a novas condições.

Este artigo detalha a proposta de uma nova abordagem, que chamamos de composi-

* Artigo financiado pelo CNPq, Projeto Nº 550114/2005-0

ções de IDSs. Estas composições de IDSs envolvem a combinação de diversos sensores e IDSs que coletam e analisam dados de forma distribuída e oferecem a flexibilidade da configuração dinâmica para atender novas situações, mesmo que temporárias. As composições de IDSs fazem uso extensivo de esforços de padronização e estão fundamentadas em uma infra-estrutura de serviços e suportes. A adoção destes padrões torna possível a interoperabilidade e a comunicação entre elementos de uma composição e, mesmo, entre IDSs completos. Os IDSs materializados a partir da infra-estrutura proposta seguem a arquitetura orientada a serviços suportada pela tecnologia de *Web Services* [W3C 2004], com o amplo uso de textos XML [Bray et al. 2004].

Este artigo traz as seguintes contribuições. Primeiro, é apresentada uma infra-estrutura para a composição de IDSs, baseada em Web Services, que é capaz de suportar a combinação de elementos de detecção de intrusão heterogêneos e distribuídos. Segundo, é introduzido o uso do conceito de orquestração de *Web Services* [Peltz 2003] para a criação e gerenciamento de composições dinâmicas de IDSs. Para isso, foi desenvolvido um procedimento genérico para a composição de IDSs, que é ilustrado através de um protótipo.

A próxima seção introduz a infra-estrutura que é a base da proposta. A seção 3 mostra como as composições de IDSs são criadas e gerenciadas. Detalhes de implementação e a avaliação da infra-estrutura proposta é descrita na seção 4 através de um protótipo e dos resultados obtidos com o mesmo. Na seção 5 são feitas algumas considerações sobre a nossa experiência com a abordagem de composição de IDSs usando *Web Services*. Na seção 6 é introduzido um breve resumo da literatura relacionada com o trabalho. Finalmente, na seção 7 são apresentadas as conclusões e algumas possibilidades de trabalhos futuros.

2. Infra-estrutura para Composição de IDSs

A infra-estrutura de serviços fornece suporte tanto para a integração de IDSs completos e independentes, quanto para a configuração de novos sistemas de detecção a partir de elementos de IDSs. São adotados como elementos básicos de um IDS aqueles definidos no modelo de detecção de intrusão do IETF [Wood 2002]: sensores, analisadores e gerenciadores.

A composição de sistemas de detecção de intrusão, a partir dos serviços do modelo proposto, deve atender às necessidades de ambientes fechados de médias e grandes empresas, mas principalmente as de ambientes abertos que fazem uso da Internet. As composições de IDSs, segundo o modelo, podem se estender por diferentes organizações, permitindo, por exemplo, o compartilhamento de alertas de segurança. Esta troca de informações pode estar sujeita a políticas que limitam o fluxo que sai de cada organização na comunicação entre elas. Para lidar com esta dificuldade, tanto os elementos de uma composição de IDSs, quanto a infra-estrutura de serviços são representados como *Web Services*.

Em geral, os IDSs são especializados e não são capazes de tratar com informações provenientes de diversos níveis e de diferentes ambientes. O modelo que propomos enfatiza o uso de elementos heterogêneos e distribuídos, permitindo a integração de ferramenta previamente existente, mesmo que de diferentes fabricantes. Portanto, a interoperabilidade é fundamental nas composições de IDSs. É necessário nestas composições que seus elementos compartilhem formas padronizadas de comunicação e de integração. O presente trabalho se concentra nos esforços dos organismos de padronização IETF[1], OASIS[2] e W3C[3].

[1] http://www.ietf.org
[2] http://www.oasis-open.org
[3] http://www.w3c.org

Uma composição de IDSs pode ser permanente ou temporária, sendo formada para coletar dados de determinados sensores, pesquisar diversas bases de dados de eventos ou para compartilhar informações sobre um ataque em andamento. A composição dinâmica permite a adaptação a situações novas em um ambiente distribuído de larga escala.

A segurança dos próprios IDSs é obviamente um ponto crítico para qualquer sistema de monitoramento. Para tal, é necessário o uso de mecanismos que possam garantir as propriedades de segurança das próprias informações trocadas ou manipuladas nestas composições distribuídas de IDSs.

2.1 Suporte de Serviços para a Composição de IDSs

A composição de IDSs envolve o uso de um conjunto de serviços e suportes que são apresentados em uma forma simplificada na Figura 1. Os elementos de uma composição de IDSs são vistos como serviços e formam, na figura, o nível mais alto da estratificação que chamamos de aplicação.

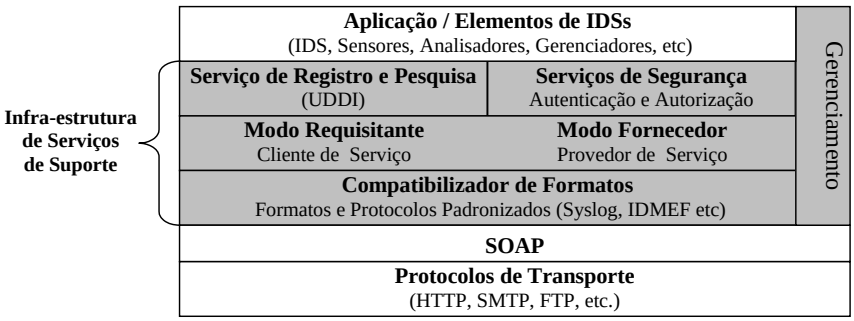

Figura 1 – Infra-estrutura para Composição de IDSs

As informações referentes aos serviços prestados por elementos de composições de IDSs, necessárias nas interações com os mesmos, são acessadas através do Serviço de Registro e Pesquisa (SRP), disponível na infra-estrutura proposta. Tal serviço está fundamentado na especificação UDDI (*Universal Description, Discovery and Integration specification*) [OASIS 2004a]. A descrição no SRP de um elemento de detecção de intrusão, na forma de *Web Service*, além de conter a identificação do serviço e a sua localização, precisa fornecer informações relacionadas ao acesso do serviço e às políticas que o governam. Suas interfaces são descritas em um formato processável, fornecido pela linguagem WSDL (*Web Services Description Language*) [W3C 2005]. A localização da descrição destas interfaces também é mantida no SRD.

Qualquer elemento de uma composição de IDSs pode funcionar no modo requisitante (cliente), no modo fornecedor (servidor) ou em ambos os modos. As interações seguem o modelo cliente/servidor, trocando *Requests* e *Replies*. O fornecimento de serviços entre elementos de composições de IDS irá depender de seus papéis: dependendo de suas atividades, as respostas dos provedores (modo fornecedor) não são imediatas a uma requisição e muitas vezes não são únicas. Por exemplo, quando um evento suspeito é identificado por um sensor, este como prestador de serviço envia ao cliente do serviço uma mensagem contendo um alerta de segurança. Este monitoramento pode ter um tempo determinado de duração, após o qual o serviço deixa de ser prestado, como, por exemplo, na primeira ocorrência do evento. Mas também pode durar indefinidamente, gerando inúmeras notificações. Na infra-estrutura proposta, os elementos de composição de IDSs se comunicam através da notificação de eventos e alertas. Portanto as operações de *Requests* e *Replies* são usadas para ativar e configurar a notificação de eventos e alertas entre os elementos da composição (ver seção 4.1).

Para definir elementos “serviços” a partir de partes dos IDSs convencionais é necessário introduzir um nível de funções que formam o que chamamos de “Compatibilizador de Formatos” (ver Figura 1), cuja principal atividade é tratar com os formatos usados nas trocas de mensagens das composições de serviços. Estas funções, por exemplo, atuam também na intermediação da comunicação entre partes usadas de IDSs convencionais (sem suporte *web*) e os *Web Services* que os disponibilizam para as composições. Este nível envolve ainda a configuração dos elementos serviços e a segurança das mensagens (estabelecimento do contexto de segurança das mensagens, cifragem e assinaturas das mensagens XML, etc). Conforme ilustrado na Figura 2, mensagens são enviadas por elementos serviços, utilizando os formatos suportados para as comunicações da composição. Estas interações estão baseadas em formatos padrões, como o IDMEF (*Intrusion Detection Message Exchange Format*) [Debar et al. 2006]. Uma vez formatadas nos padrões especificados, as notificações codificadas em SOAP [W3C 2003] são assinadas e cifradas usando as recomendações da especificação *WS-Security* [OASIS 2004c] para garantir a segurança fim a fim com o *XML-Encryption* [Reagle 2002][Imamura et al. 2002] e o *XML-Signature* [Eastlake et al. 2002][Reagle 2000].

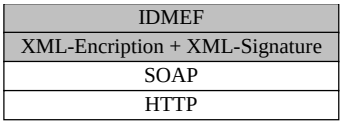

Figura 2 - Encapsulamento de Mensagens de Detecção de Intrusão

2.2 Serviço de Registro e Pesquisa

O Serviço de Registro e Pesquisa (SRP) é peça fundamental de nossa proposta. Os elementos que podem ser usados em composições de detecção de intrusão são registrados e descritos no SRP antes de serem oferecidos como *Web Services*. Uma vez disponibilizados e devidamente registrados, os serviços destes elementos podem ser localizados para que interajam com outros serviços na composição.

O SRP é baseado na especificação UDDI. Esta especificação usa a abordagem de registro [W3C 2004] com ênfase na criação de domínios administrativos para o armazenamento de informações. A especificação define também mecanismos para a associação de diversos servidores UDDI, provendo a escalabilidade necessária ao SRP.

Na UDDI, estruturas do tipo *tModel* definem tipos de *Web Services*, protocolos utilizados pelos mesmos ou categorias de sistemas. As interfaces do serviço, descritas em documento WSDL, são obtidas a partir das URLs disponíveis nos *tModels*.

Em geral, para *Web Services*, a localização e detalhes de implementação não são relevantes. Porém, no caso de serviços relacionados à detecção de intrusão, tais informações podem ser indispensáveis para a composição dinâmica de IDSs. Isto ocorre, por exemplo, quando é preciso localizar um sensor em um ponto específico de uma rede. Informações sobre o elemento de detecção de intrusão e sua localização são incluídas no registro do serviço na UDDI utilizando a classe *Analyzer* do IDMEF.

Para facilitar a localização de determinado tipo de elemento para as composições de IDSs, desenvolvemos uma classificação para os mesmos de acordo com seus papéis. Os elementos sensores seguem taxonomia proposta em [Alessandri et al. 2001]. No caso dos analisadores e IDSs monolíticos, usamos em suas classificações as mais recentes taxonomias de IDSs [Axelsson 2000][Debar et al. 2000][McHugh 2001], combinadas de acordo com a arquitetura e o método de detecção adotado. Maiores detalhes sobre as estruturas de armazenamento e o Serviço de Registro e Pesquisa podem ser obtidos em [Brandão et al. 2006].

2.3 Serviço de Segurança

O Serviço de Segurança que trata da autenticação e do controle de acesso dos elementos envolvidos na composição nesta primeira versão estão também baseados na UDDI. Os mecanismos de segurança são baseados na especificação UDDI, tanto os usados para a autenticação dos operadores e administradores quando dos registros e ativação das composições, como para a privacidade e integridade (controle de acesso) das informações dos elementos de IDSs disponíveis no SRP. As mensagens de registro e de busca são encapsuladas em SOAP, utilizando os mecanismos de segurança previstos na especificação *WS-Security*.

As chaves públicas e os contextos de segurança a serem usados nas comunicações entre os elementos da composição, seguem a especificação *WS-Security* e são obtidas na UDDI junto com o registro destes elementos através do Serviço de Segurança. No registro do elemento, uma estrutura *bindingTemplate* armazena a chave pública ou indica sua localização. As chaves públicas e chaves privadas são baseadas no modelo X.509 [ITU-T 1993].

3. Orquestração de Serviços

Uma composição de IDSs é efetivada a partir de sua representação que é dada através de seu registro na UDDI. Este registro é feito na forma de um *tModel*. Documentos indicados em *tags* do tipo *overview_Doc* contidas no *tModel* irão identificar e descrever a composição. Entre estes documentos está o WSDL que descreve as interfaces da composição e o XML que descreve as interações e a organização dos elementos da composição, conforme ilustrado na Figura 3. A partir destes documentos, a composição poderá ser implementada.

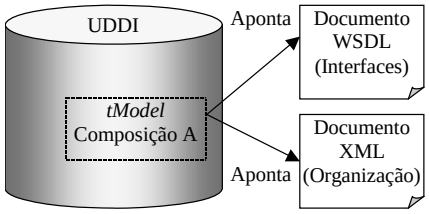

Figura 3 - Registro de Composições

Os termos "orquestração" (*orchestration*) e "coreografia" (*choreography*) são normalmente usados para descrever aspectos da criação de composições de *Web Services* [Peltz 2003]. Ambos os termos são empregados para representar processos de negócios. Um processo de negócio é um conjunto de passos parcialmente ordenados, cujo objetivo é atingir uma meta como, por exemplo, a construção de um IDS de larga escala. A orquestração descreve como *Web Services* podem interagir com outros *Web Services* internos e externos a um domínio administrativo. A orquestração inclui a lógica do negócio (ou seja, o comportamento desejado da composição) e a ordem das execuções definidas a partir de fluxos de controles que atravessam organizações e aplicações. A orquestração sempre representa o fluxo de eventos da composição a partir da perspectiva de uma das partes envolvidas. No caso específico da composição de IDSs, a visão é a do administrador responsável pela segurança. A coreografia [Austin et al. 2004], outro dos termos usados para composições, concerne às interações observáveis entre os serviços e seus usuários. Uma descrição da coreografia é um contrato multiparte que descreve, de um ponto de vista global, o comportamento observável externo entre múltiplos clientes. O "comportamento externo observado" é definido pela presença ou ausência de mensagens que são trocadas entre os Web Services e seus clientes. Coreografia e orquestração são formas complementares de compor Web Services.

O documento XML apontado por um *tModel*, no registro de uma composição na

UDDI irá representar uma orquestração. Ao invés da coreografia, a orquestração foi escolhida por prover a flexibilidade necessária para lidar com composições dinâmicas de IDSs. A orquestração permite que um administrador de segurança defina um fluxo de processo genérico que possa ser usado para criar uma composição e que não precisa ser alterado quando elementos são adicionados ou removidos, enquanto a coreografia carece de flexibilidade para acomodar tal evolução (se elementos são adicionados ou removidos, a coreografia precisa ser alterada). Contudo, o uso de coreografia não está descartado em uma futura investigação.

A Figura 4 ilustra uma orquestração simplificada da criação de uma composição de IDSs. Os passos internos são interpretados por um motor (*engine*) de orquestração que os executará seqüencialmente ou em paralelo, conforme definido pelo administrador. Os serviços de suporte à composição são invocados na ordem estabelecida no fluxo do processo definido. Da mesma forma, o fluxo do processo estabelece a ordem em que os elementos da composição de IDSs são agregados à composição. O uso das ferramentas para criação e descrição de IDSs será discutido na seção 4.1. Na figura os passos paralelos representam, por exemplo, a invocação simultânea de elementos sensores.

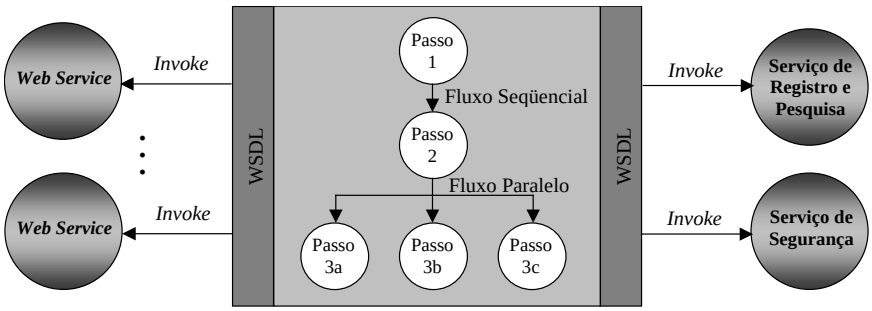

Figura 4 - Orquestração

Toda composição de IDSs possui um ou mais Serviços Gerenciadores que são os elementos que permitem a um administrador (ou responsável) interagir com toda a composição de IDSs no sentido da visualização de alertas e mesmo, para a reconfiguração do sistema de detecção. Os elementos de IDS, para as interações que envolvem reconfigurações e ativações a partir dos Serviços Gerenciadores, fornecem interfaces especificadas segundo o *Web Services Distributed Management* (WSDM) [OASIS 2004b]. A definição de interfaces segundo este padrão permite que qualquer serviço seja configurado e monitorado utilizando padrões de gerenciamento específicos para *Web Services*. Tais interfaces são agrupadas em habilidades específicas (*capabilities*). O WSDM fornece habilidades relacionadas à identificação, estado de funcionamento, disponibilidade do componente, configuração, métricas para monitoramento e notificação de eventos. Cada *Web Service* define em seu documento WSDL quais habilidades estão disponíveis. Por enquanto, apenas as habilidades de notificação de eventos e configuração são essenciais para os elementos da composição de IDSs.

4. Experimentos Realizados

Para validar a proposta, um protótipo da infra-estrutura de serviços foi implementado. A ferramenta *BEA Weblogic Workshop 8.1*[4], que provê um ambiente integrado de programação, um servidor *Web Service* e uma UDDI, foi usada na implementação deste protótipo. Também foram testadas as UDDIs *WSDP*[5] e *JUDDI*[6], além do servidor *Web Service Tom-*

[4] http://www.bea.com
[5] http://java.sun.com/webservices/downloads/webservicespack.html
[6] http://ws.apache.org/juddi/

Cat[7] e ferramentas de suporte para WSDM (*Muse*)[8] e SOAP (*Axis*)[9].

Para testar o protótipo foram usados nas composições dois IDSs conhecidos: o IDS de rede *Snort*[10] e o *Prelude-ids*[11]. Os sensores e analisadores da composição foram obtidos a partir destes IDSs. Ambos os IDSs tiveram seu código fonte adaptado para produzir mensagens no formato IDMEF padrão e atualizado (ver seção 6).

A transformação das partes de um IDS convencional em elementos "serviços" é feita neste protótipo por um adaptador que serve de *gateway* entre os protocolos originais destes IDSs convencionais e os protocolos da camada "Compatibilizador de Formatos".

4.1 Composição Implementada

A Figura 5 mostra o experimento montado. Aparecem nesta figura os sensores e analisadores obtidos a partir do *Prelude-ids* e do *Snort* e as composições que testamos. Em uma rede "A", dois serviços sensores (baseados no *Snort*) são configurados para enviarem notificações de eventos a um serviço analisador (baseado no *prelude-ids*). Este serviço analisador correlaciona as notificações recebidas dos sensores e gera novos alertas de acordo com os critérios configurados pelo serviço gerenciador. Na rede "B", um serviço sensor (*Snort*) envia suas notificações de eventos diretamente ao serviço gerenciador.

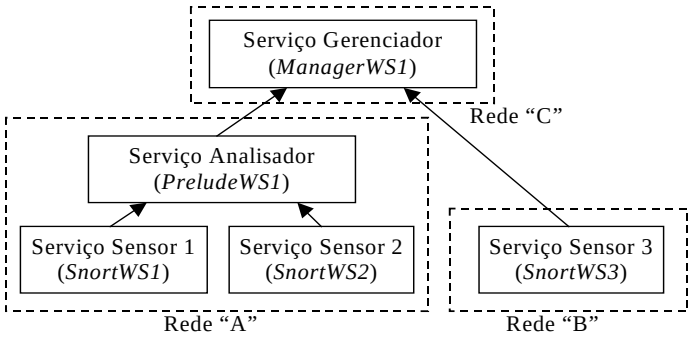

Figura 5 – Composições de teste

O Serviço Gerenciador usado nos nossos experimentos foi implementado na linguagem Java utilizando recursos para criação de Web Services oferecidos pela ferramenta *BEA Weblogic Workshop*. A interface do serviço gerenciador segue o formato de um portal *web*.

Existem atualmente diversas linguagens e ferramentas que auxiliam na orquestração de *Web Services* [Peltz 2003][Wang et al. 2004]. Neste trabalho foi adotado o BPEL4WS (*Business Process Execution Language for Web Services*) [Andrews et al. 2003][OASIS 2005a], também chamado de BPEL. Esta escolha deve-se, principalmente, à disponibilidade de ótimas ferramentas no mercado e à vasta documentação disponível.

O BPEL4WS visa a composição de *Web Services* em um ambiente distribuído, principalmente entre múltiplas organizações. Sua especificação provê uma gramática baseada em XML para descrever a lógica de controle necessária para coordenar uma composição. Esta gramática pode ser interpretada e executada por motores de orquestração que são controlados por uma das partes envolvidas. Os blocos de construção dos processos BPEL4WS descrevem atividades executadas dentro de uma composição. Há atividades básicas e atividades

[7] http://jakarta.apache.org/tomcat/
[8] http://ws.apache.org/muse
[9] http://ws.apache.org/axis
[10] http://www.snort.org/
[11] http://www.prelude-ids.org/

estruturadas. As atividades básicas são instruções de interação entre os *Web Services*. As atividades estruturadas descrevem o fluxo de execução do processo.

Uma vez descrita (via *BEA Weblogic Workshop*), a composição pode ser implementada a partir de serviços *web* ou exportada para o formato BPEL, junto com a descrição WSDL. Estes documentos são registrados e representam, respectivamente, as interfaces da composição e o fluxo de execução da mesma (ver seção 3). A partir destes documentos é possível recuperar (construir) a composição de IDSs e sua correspondente orquestração.

4.2 Procedimento Geral para Implantação de Composições

O procedimento geral desenvolvido para a criação e a manutenção dinâmica de composições de IDSs é apresentado nesta seção. A Tabela 1 descreve os passos necessários para a criação de qualquer composição de IDSs utilizando a infra-estrutura desenvolvida. A Figura 6 mostra as operações básicas de uma orquestração, definidas por este procedimento, usando uma interface gráfica.

Tabela 1- Passos para a criação de composições de IDSs

Passo	Operação
1.	Localizar na UDDI o Serviço Gerenciador, de acordo com os parâmetros de busca especificados utilizando a operação *ClientRequestDiscovery*
2.	Executar a operação *ClientRequestCreate* para criar um registro da composição na UDDI.
3.	Inicializar o Serviço Gerenciador (*invoke*).
4.	Associar o Serviço Gerenciador à composição, usando a operação *ClientRequestRegistry*.
5.	Localizar na UDDI o Serviço Analisador, de acordo com os parâmetros de busca especificados utilizando a operação *ClientRequestDiscovery*.
6.	Executar a operação *ClientRequestSubscribe* no Serviço Analisador para subscrição dos alertas, enviando-os ao Serviço Gerenciador.
7.	Associar o Serviço Analisador à composição, usando a operação *ClientRequestRegistry*.
8.	Repetir os passos 5, 6 e 7 para cada um dos demais elementos envolvidos na composição (Sensores 1, 2 e 3).

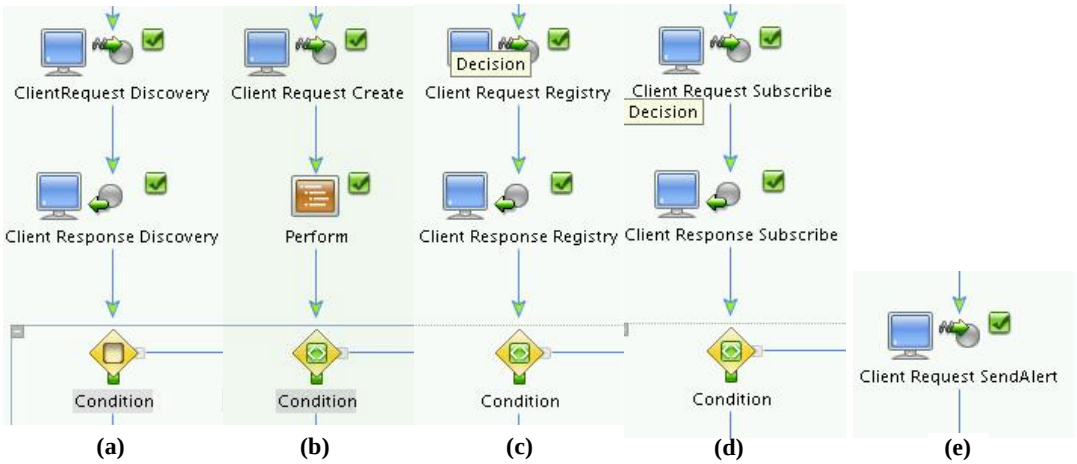

Figura 6 - Modelo Geral de Composição

A princípio não se conhece a localização dos serviços, apenas as funcionalidades desejadas dos mesmos. A Operação *ClientRequestDiscovery* (Figura 6a) é responsável por executar a busca por serviços na UDDI utilizando o Serviço de Registro e Pesquisa, de acordo com parâmetros estabelecidos na operação. Nos testes realizados no protótipo foram usados dois parâmetros: as características do elemento de detecção de intrusão descritas na estrutura *categoryBag* (seção 2.2); e o endereço da rede (ou sub-rede) em que o elemento deve estar operando (esta última contida na estrutura da classe *Analyzer*).

A operação *ClientRequestCreate* (Figura 6b) é responsável pela requisição para a criação do registro da composição. A operação *Perform* faz a verificação dos dados e a inserção do registro da composição na UDDI. As operações *ClientRequestRegistry* e *ClientResponseRegistry* (Figura 6c) são responsáveis respectivamente pela requisição e resposta da publicação de um serviço na UDDI. Os registros destes serviços podem ou não ser inseridos na UDDI, seguindo políticas de segurança pré-estabelecidas.

As operações *ClientRequestSubscribe* e *ClientResponseSubscribe* (Figura 6d), por sua vez, fazem a subscrição de serviços. Elas são baseadas, respectivamente, nas operações *SubscribeRequest* e *SubscribeResponse* das interfaces padrões da especificação WSN (*Web Service Notification*) [OASIS 2005b]. Por último, a operação *ClientRequestSendAlert* (Figura 6e), que é baseada na operação *Notify* do WSN, é utilizada pelos serviços já registrados na composição para enviar alertas à composição.

A orquestração de composições segue uma seqüência lógica e cronológica dos eventos representados na Figura 6. A partir de cada evento é possível continuar ou ir para o final do fluxo de execução (representado pelo *Condition*). Para executar a operação *ClientRequestSendAlert*, por exemplo, não é necessário executar os eventos anteriores, desde que eles já tenham sido executados pelo menos uma vez.

4.3 Testes do Protótipo

Os testes no protótipo foram realizados em um ambiente de produção do campus universitário, utilizando elementos de composição de IDSs localizados em redes distintas. O campus universitário possui milhares de computadores com dezenas de sistemas operacionais diferentes, distribuídos em diversas subredes interconectadas, com suas barreiras (como *firewalls*) e riscos de segurança. Portanto, a escolha do ambiente de testes reflete a heterogeneidade e a diversidade das redes de larga escala. Utilizando a composição descrita na Figura 5 foram realizadas simulações de ataques. Os testes obedeceram a seqüência indicada na Tabela 2.

Tabela 2 – Descrição dos Testes

1.	Foram registrados na UDDI diversos possíveis elementos para composição de IDSs
2.	Uma composição foi criada inicialmente na rede "A" com dois IDSs monolíticos baseados no Snort (*SnorWS1* e *SnortWS2*) agindo como sensores nesta rede e enviando seus alertas a um analisador baseado no *Prelude-ids* (*PreludeWS1*). O Serviço Gerenciador (*ManagerWS1*) recebe e apresenta as notificações ao administrador.
3.	Em um ataque simulado, o elemento *SnortWS1* (na rede "A") detecta uma possível tentativa de ataque procedente do *host* "10.2.13.122", localizado na rede "B".
4.	A composição é alterada para incluir um novo sensor localizado na rede "B". O serviço *SnortWS3* é localizado e ativado para coletar o tráfego proveniente do *host* suspeito e enviar os eventos diretamente ao *ManagerWS1*.
5.	Novas mensagens de alerta confirmam a tentativa de ataque a partir do *host* suspeito.
6.	Após as medidas administrativas para contenção do ataque, a composição volta à sua configuração original.

Nesta simulação, um usuário na console de um *host* suspeito realizou um ataque de *port Scan*[12] utilizado o software *Nmap*[13]. Na detecção e resposta a outros ataques, mais serviços poderiam ser ativados para localizar a origem e identificação de possíveis danos causados. Uma base de dados com eventos de auditoria também poderia ser pesquisada para identificar com maior precisão o usuário que originou o ataque.

5. Considerações sobre a composição de IDSs

O uso de composições de IDSs dinâmicas, suportadas pela infra-estrutura de serviços proposta neste texto, torna possível a construção de soluções mais adequadas para o monitora-

[12] Varredura de portas de um *host* em busca de vulnerabilidades de segurança.
[13] http://www.insecure.org

mento de segurança em redes de larga escala. O uso de *Web Services* permite que elementos de composições para detecção de intrusão possam ser acessados atravessando domínios de redes distintas ou, ainda, segmentadas. Usando esta infra-estrutura proposta, elementos para composição de IDSs podem ser implementados em ambientes de larga escala, registrados e ativados sob demanda. O uso da UDDI com uma classificação específica destes elementos torna a busca de serviços especializados mais fácil e precisa.

A cooperação entre empresas e organizações é outro ponto importante que pode ser viabilizado com a presente proposta. Elementos de detecção de intrusão podem ser disponibilizados entre parceiros para permitir a investigação de atividades suspeitas provenientes de suas diversas redes de computadores. Isto, porém, requer a especificação e monitoramento de políticas de segurança precisas entre as partes envolvidas.

Esta proposta favorece também a terceirização da tarefa de análise de alertas de segurança. Serviços analisadores mantidos por centros de resposta a incidentes de segurança ou empresas prestadoras de serviço podem ser ativados para receberem alertas e correlacioná-los com alertas recebidos de outros clientes. Como retorno, cada cliente pode obter alertas globais, estatísticas precisas sobre incidentes e parâmetros para configuração de suas redes.

Apesar da proposta enfatizar a integração dos diversos padrões adotados neste trabalho, isto na prática não é tão trivial. As várias organizações que definem os padrões nem sempre estão de acordo e especificações que seriam, aparentemente, complementares acabam sendo incompatíveis. Outra dificuldade está relacionada a trabalhos de padronização incipientes. De qualquer forma acreditamos que a proposta é um passo significativo para a detecção de incidentes em sistemas de larga escala.

6. Trabalhos Relacionados

Trabalhos envolvendo a detecção de intrusão distribuída em ambientes de larga escala e o uso de padrões de *Web Services* ainda são raros na literatura relacionada. Contribui para isto o fato que a detecção de intrusão em sistemas distribuídos de larga escala é uma área de pesquisa recente. O mesmo ocorre com as definições e padrões envolvendo *Web Services*.

Podemos afirmar que os sistemas de detecção de intrusão convencionais não são estruturados para a troca de informações entre diferentes organizações, mantendo as informações restritas ao escopo da organização na qual foram coletadas. Uma das poucas exceções é o IDF [Teo et al. 2003], que propõe mecanismos para a troca de informações sobre incidentes de segurança entre organizações através da Internet. Propostas recentes de IDSs distribuídos [Teo et al. 2003][Tolba et al. 2005][Leu et al. 2005], em geral, não usam formatos e protocolos padrões para a comunicação entre os elementos envolvidos na detecção. Contudo, a necessidade de se utilizar formatos comuns para a comunicação entre IDSs está presente na literatura. A proposta de Bass [Bass 2004] é um exemplo disso, onde notificações de segurança são geradas em formatos nativos e transformadas em um formato único. Porém, infelizmente, este formato único não segue os padrões existentes.

Esforços recentes de padronização relacionados à troca de informações de segurança estão sendo desenvolvidos, principalmente, pelo IETF, através dos grupos de trabalho IDWG[14] e INCH[15]. O IDWG está concluindo a especificação do *Intrusion Detection Message Exchange Format* (IDMEF) [Debar et al. 2006] e do *Intrusion Detection Exchange Protocol* (IDXP) [Feinstein et al. 2002]. Estas propostas de padronização visam a troca de in-

[14] http://www.ietf.org/html.charters/OLD/idwg-charter.html
[15] http://www.ietf.org/html.charters/inch-charter.html

formações entre IDSs e entre os elementos que os formam. O IDWG também define um modelo geral de detecção de intrusão, cujos elementos são a base da proposta apresentada neste documento. O INCH está trabalhando na troca de informações e estatísticas sobre incidentes de segurança entre grupos de resposta a incidentes de segurança (CSIRTs - *Computer Security Incident Response Teams*). Os requisitos [Keeni et al. 2006] e o modelo de dados (IODEF - *Incident Object Description Exchange Format*) [Danyliw et al. 2006] para implementação estão em fase de especificação. Todas estas especificações são baseadas na linguagem XML [Bray et al. 2004].

São poucas as experiências na literatura ou mesmo de produtos que se utilizam destes padrões emergentes. Um dos IDSs mais populares, o *Snort*, pode utilizar um *plugin* que o permite enviar alertas no formato IDMEF. O DOMINO [Yegneswaran et al. 2004] e a família de IDSs STAT [Vigna et al. 2003] estendem o formato IDMEF para atender suas necessidades. O uso de padrões nestes casos é limitado e as extensões realizadas não são completamente compatíveis com a especificação original.

Um IDS que emprega o modelo do IDWG e *Web Services* é descrito em [Park et al. 2003]. Infelizmente, o modelo usa apenas um método de detecção, centraliza a análise e, apesar de adotar o IDMEF internamente, não permite a integração com outros IDSs.

Um IDS que se aproxima da nossa proposta é o *Prelude-ids*. É um IDS híbrido que agrega e correlaciona alertas gerados por sensores de diversos tipos e fabricantes, distribuídos em uma rede de computadores. As mensagens são enviadas sobre conexões SSL. No *Prelude-ids*, é utilizado um modelo hierárquico de análise, reportando eventos a um ou mais gerenciadores. Como o grupo IDWG padroniza apenas o uso da linguagem XML para a formatação das mensagens, o formato nativo utilizado pelo *Prelude-ids* para a comunicação com os sensores seria incompatível com o padrão original do IDMEF. No *Prelude-ids* a análise de alertas IDMEF em XML só é possível *off-line*, pela importação de arquivos com o uso de um produto comercial. Portanto, fica comprometida a adoção de sensores baseados no IDMEF padrão, para envio de alertas ao *Prelude-ids* para detecção *on-line*. Apesar disso, o gerenciador do *Prelude-ids* é capaz de formatar alertas em XML, o que facilita seu uso como sensor ou analisador em IDSs de larga escala, como o que propomos.

Ao contrário dos IDSs apresentados nesta seção, nossa proposta utiliza apenas formatos padronizados originais para a comunicação, possibilitando a integração de qualquer sensor, analisador ou gerenciador a uma composição de IDSs. O *Prelude*-ids e o *Snort* são usados como analisadores e sensores em nosso protótipo. Para isso, foram feitas modificações para torná-los totalmente compatíveis com os padrões originais.

São muitas as propostas de solução para a criação e o gerenciamento de composições de *Web Services*. Temas como orquestração, coreografia, *workflow*, padrões e ferramentas utilizadas estão presentes por toda a literatura sobre o assunto. Uma boa visão sobre isso pode ser obtida em [Peltz 2003]. As análises dos problemas e soluções para composição de *Web Services* feita por [Wang et al. 2004] e para o gerenciamento de composições, feita por [Esfandiari e Tosic 2005], foi bastante útil para a escolha das ferramentas para orquestração. Segundo [Esfandiari e Tosic 2005], o gerenciamento de composições de serviços deve suportar todo o ciclo de vida das mesmas, incluindo a descoberta de serviços e contratação de serviços, o monitoramento dos requisitos da composição e, ainda, possíveis recomposições de serviços. Tal requisito foi levado em conta na nossa solução.

Novas propostas para IDSs distribuídos incluem o uso de grades computacionais (*Grids*), como os projetos GIDA [Tolba et al. 2005] e GIGS [Leu et al. 2005] distribuem a tarefa de análise dos dados coletados por sensores entre nodos de uma grade computacional.

A análise do mecanismo de gerenciamento de composições em escala global baseado em *Grids*, proposto por [Vambenepe et al. 2005], usando um *workflow* BPEL, também foi bastante útil para o desenvolvimento do processo de composição de IDSs. Apesar de não fazermos o uso de *Grids*, nossa proposta também pode ser aplicado em tal ambiente.

O uso de *Web Services* para a composição de IDSs foi proposto resumidamente em [Brandão et. al 2005], onde foram apresentados os objetivos básicos para a composição de IDSs, adotados na presente proposta. Mais detalhes sobre o modelo e, principalmente, o funcionamento do Serviço de Registro e Pesquisa no registro dos elementos de IDSs foram introduzidos em [Brandão et. al 2006]. Porém, não foram apresentados detalhes sobre a criação, registro e gerenciamento das composições. Tais questões são tratadas no presente documento.

A Tabela 3 ilustra as diferenças entre os IDSs distribuídos analisados e a nossa proposta, com relação à arquitetura, mecanismos de comunicação utilizados e à possibilidade de interoperabilidade com outros IDSs.

Tabela 3 - Interoperabilidade nos IDSs Distribuídos

IDS	Comunicação	Interoperabilidade com outros IDSs
IDF	Não Padronizada	Não
GIDA	Não Padronizada	Não
GIDS	Não Padronizada	Não
[Bass 2004]	Não Padronizada	Não
DOMINO	IDMEF alterado	Não
STAT	IDMEF alterado	Não
Prelude-ids	IDMEF alterado	Recepção no formato nativo ou IDMEF *off-line*
[Park et al. 2003]	IDMEF	Não
Composição de IDSs	**Diversos padrões (implementado IDMEF)**	**Sim**

7. Conclusões e Trabalhos Futuros

Neste trabalho foi apresentada uma nova abordagem para a composição dinâmica de sistemas de detecção de intrusão. O modelo permite a integração de elementos de IDSs ou mesmo de sistemas monolíticos para criar sistemas de detecção de intrusão distribuídos, em ambientes de larga escala. Para conseguir flexibilidade e interoperabilidade nestas composições dinâmicas, é proposta uma infra-estrutura baseada em serviços que deve servir de suporte a estas composições. Esta infra-estrutura está fortemente fundamentada na tecnologia de *Web Services* e em padrões para comunicação de alertas de segurança.

A infra-estrutura é apresentada neste texto como uma estratificação de serviços. Estes serviços contribuem nos vários níveis para viabilizar as composições de IDSs e tornar as comunicações entre seus elementos interoperáveis e seguras. Como contribuição adicional, podemos afirmar que outros padrões, além dos apresentados neste texto podem ser integrados facilmente nesta infra-estrutura de serviços.

Como foco principal deste trabalho propomos a criação e o gerenciamento das composições utilizando o conceito de orquestração de *Web Services*. A organização dos elementos de detecção de intrusão e invocação dos serviços de suporte é descrita utilizando linguagem BPEL4WS e interpretada por um motor de orquestração.

Nos trabalhos futuros, pretendemos refinar o serviço de segurança. Consideramos também a aplicação da proposta em grades computacionais.

Referencias

Alessandri, D., et al. (2001). Towards a taxonomy of intrusion detection systems and attacks. MAFTIA Deliverable D3, EU Project IST-1999-11583 Malicious- and Accidental-Fault Tolerance for Internet Applications (MAFTIA). Version 1.01.

Andrews, T., et al. (2003). Business Process Execution Language for Web Services. Version 1.1 - 5 May 2003.

Austin, D., et al. (2004). Web services choreography requirements. W3c working draft 11.

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Department of Computer Engineering, Chalmers University of Technology, SE-412 96 Göteborg, Sweden.

Bace, R. and Mell P. (2001). Intrusion Detection Systems. NIST Special Publication on Intrusion Detection System.

Bass, T. (2004). Service-oriented horizontal fusion in distributed coordination-based systems. *IEEE MILCOM 2004*.

Bray, T., Paoli, J., and Sperberg-McQueen, C. M. (2004). Extensible markup language (XML) 1.0 (third edition)". Technical report, W3C.

Brandão, J. E., Fraga, J. S. , and Mafra, P. M. (2005). Composição de IDSs Usando Web Services. *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg2005)*. p. 339-342.

Brandão, J. E., Fraga, J. S. , and Mafra, P. M. (2006). A New Approach for IDS Composition. *In IEEE International Conference on Communications (ICC 2006)*, Istambul, Turquia. IEEE.

Danyliw, R., Meijer, J., and Demchenko, Y. (2006). The incident object description exchange format data model and xml implementation. Technical Report draft-inch-ietf-iodef-06.txt, IETF Extended Incident Handling WG.

Debar, H., Curry, D., and Feinstein, B. (2006). The intrusion detection message exchange format. Technical Report draft-ietfidwg-idmef-xml-16, IETF.

Debar, H., Dacier, M., and Wespi, A. (2000). A revised taxonomy for intrusion detection systems. *Annales des Telecommunications*, 55(7–8):361–378.

Eastlake, D., Reagle, J., and Solo, D. (2002). (extensible markup language) xml-signature syntax and processing. Request for Comments 3275, Internet Engineering Task Force.

Esfandiari, B. and Tosic, V. (2005). Towards a web service composition management framework. In proceedings of IEEE International Conference on Web Services (ICWS'05), pages 419–426. IEEE.

Feinstein, B., Matthews, G., and White, J. (2002). The intrusion detection exchange protocol (idxp). Technical Report draft-ietf-idwg-beep-idxp-07, IETF.

Imamura, T., Dillaway, B., and Simon, E. (2002). Xml encryption syntax and processing, w3c recommendation. Technical report, W3C.

ITU-T (1993). ITU-T recommendation x.509.

Keeni, G., Danyliw, R., and Demchenko, Y., (2006). Requirements for the Format for Incident Information Exchange (FINE). Technical Report draft-ietf-inch-requirements-08.txt, IETF.

Leu, F.-Y., et al.. (2005). Integrating grid with intrusion detection. In *AINA*, pages 304–309.

McHugh, J. (2001). Intrusion and intrusion detection. *Int. J. Inf. Sec.*, 1(1):14–35.

OASIS (2004a). UDDI version 3.0.2. OASIS UDDI Spec Technical Committee Draft.

OASIS (2004b). Web Services distributed management: Management using Web Services (muws 1.0) part 2 - Web Services distributed management: Management of Web Services (wsdm-mows) 1.0. OASIS Web Services Distributed Management (WSDM) TC.

OASIS (2004c). Web Services security: SOAP message security 1.0. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

OASIS (2005a) Business Process Execution Language for Web Services. Version 2.0 - Committee Draft, 01 September 2005.

OASIS (2005b). Web Services Base Notification 1.3. OASIS Web Services Notification (WSN) TC.

Park, S., Kim, K., Jang, J., and Noh B. (2003). Supporting Interoperability to Heterogeneous IDS in Secure Networking Framework. *APCC Communications*, 2(21-24):844 – 848.

Peltz, C. (2003). Web Services orchestration and choreography. *IEEE Computer*, 36(10):46–52.

Reagle, J. (2000). XML signature requirements. Request for Comments 2807, Internet Engineering Task Force.

Reagle, J. (2002). Xml encryption requirements. Note 04, W3C.

Teo, L., Zheng, Y., and Ahn, G.-J. (2003). Intrusion detection force: An infrastructure for internet-scale intrusion detection. In *First IEEE International Information Assurance Workshop (IWIA 2003)*, pages 73–88, Germany.

Tolba, M., et al. (2005). Gida: Toward enabling grid intrusion detection systems. *5 th IEEE International Symposium on Cluster Computing and the Grid*.

Vambenepe, W., et al. (2005). Dealing with scale and adaptation of global Web Services management. In proceedings of IEEE International Conference on Web Services (ICWS'05), pages 339–346. IEEE.

Vigna, G., Valeur, F., and Kemmerer, R. A. (2003). Designing and implementing a family of intrusion detection systems. In *ESEC/FSE-11: Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 88–97, New York, NY, USA. ACM Press.

W3C (2003). Soap version 1.2. W3C World Wide Web Consortium.

W3C (2004). Web Services Architecture. W3C Working Group Note 11.

W3C (2005). Web Services Description Language (WSDL) version 2.0 part 1: Core language. W3C Working Draft.

Wang, H., Huang, J. Z., Qu, Y., and Xie, J. (2004). Web Services: problems and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):309–320.

Wood, M. and Erlinger, M. (2002). Intrusion detection message exchange requirements. Technical Report draft-ietf-idwg-requirements-10, IETF.

Yegneswaran, V., Barford, P., and Jha, S. (2004). Global intrusion detection in the domino overlay system. In *NDSS*, San Diego, California, USA. The Internet Society.