Comunicação Segura na Composição de IDSs e seus Custos

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

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

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

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

Abstract. *The intrusion detection systems are usually designed to work in local area networks. These systems do not foresee the integration with other intrusion detection tools, reducing the scope of the analysis. This paper presents a model for secure communication between components of intrusion detection systems in large-scale environments. The introduced model supports the end-to-end security by using standards and the Web Services technology. A comparison between the involved communication technologies is also presented.*

Resumo. *Os sistemas de detecção de intrusão geralmente são projetados para funcionar em redes locais. Tais sistemas não prevêem a integração com outras ferramentas de detecção de intrusão, reduzindo o escopo da análise. Este artigo apresenta um modelo para a comunicação segura entre os componentes de sistemas de detecção de intrusão em ambientes de larga escala. O modelo introduzido suporta a segurança fim a fim, utilizando padrões de segurança e a tecnologia de Web Services. Uma comparação dos custos das tecnologias de comunicação envolvidas também é apresentada.*

1. Introdução

Os sistemas de detecção de intrusão (*Intrusion Detection Systems - IDS*) possuem um papel importante na segurança das redes de computadores, auxiliando a equipe de administração da rede a detectar possíveis violações ou tentativas de violação dos sistemas computacionais. Contudo, a maioria dos sistemas de detecção de intrusão existentes atua apenas em redes locais, não possuindo uma estrutura para compartilhar informações de segurança com partes de uma mesma organização ou outras organizações em ambientes de larga escala. A presença de *firewalls*[1] é outro fator que dificulta a interoperabilidade destes sistemas de detecção. Desta forma, as informações coletadas em uma rede local ficam restritas à própria rede, sendo inviável o compartilhamento e a correlação de eventos de segurança, limitando o escopo da análise.

Em [Brandão et al. 2006a] foi apresentado um modelo para composição de IDSs utilizando IDSs completos existentes ou apenas partes destes. Para a concretização das

*Este trabalho foi apoiado pelo CNPq, processo de número 550114/2005-0.

[1]Programas que filtram o tráfego da rede.

composições de IDSs faz-se necessário um modelo para a comunicação segura e interoperável entre seus elementos. Este artigo apresenta o modelo desenvolvido para a comunicação segura e interoperável entre os elementos de detecção de intrusão em ambientes de larga escala. A aplicação deste modelo permite a integração segura de elementos de detecção de intrusão distribuídos entre sub-redes de uma mesma organização ou entre organizações distintas para formar IDSs de larga escala.

O modelo proposto é aplicado em sistemas de larga escala (por exemplo, organizações virtuais [Park et al. 2003]) e faz uso de padrões de segurança e da tecnologia de *Web Services* [Booth et al. 2004] para conseguir a interoperabilidade nas integrações de IDSs. Entre as principais contribuições deste trabalho, podemos citar: o modelo de comunicação desenvolvido que permite integrar diversos formatos de registro de alerta em um formato único padrão, possibilitando a integração e correlação de eventos de segurança de diversas ferramentas existentes; a inclusão de mecanismos de segurança para a interação entre elementos de detecção de intrusão; e um estudo do impacto do uso de algumas tecnologias na comunicação segura entre elementos de IDS.

O artigo está organizado da seguinte forma: na seção 2 discutimos alguns dos principais trabalhos nesta recente área de pesquisa. Na seção 3, descreve-se brevemente o modelo de composição de IDSs utilizando *Web Services*, no qual o modelo de comunicação proposto está baseado. A seção 4 apresenta a proposta de comunicação segura para o modelo de composição de IDSs. Na seção 5 são apresentados os testes realizados para validação do modelo de comunicação segura e verificação dos custos computacionais. Ao final, traçamos algumas considerações sobre o modelo proposto, bem como possíveis trabalhos futuros.

2. Trabalhos Relacionados

A literatura relacionada à detecção de intrusão distribuída em ambientes de larga escala é recente. Podemos destacar exemplos, como [Leu et al. 2005, Tolba et al. 2005], nos quais são criados IDSs distribuídos utilizando a tecnologia de grades computacionais. Destacamos também o projeto *Intrusion Detection Force* [Teo et al. 2003], que define uma infra-estrutura com entidades e grupos formados pelas mesmas em diversas localidades (redes), que coletam e compartilham as informações de segurança, provendo a integração e a análise dos dados de diferentes localidades. Porém estas soluções de IDSs de larga escala não utilizam um formato padrão para mensagens, dificultando a interoperabilidade entre os IDSs. Em [Bass 2004], alertas de segurança gerados em formatos nativos são transformados em um formato único, não padronizado, adotado apenas neste sistema.

Um esforço para padronização de formatos, baseados na linguagem XML (eXtensible Markup Language) [Bray et al. 2000], para a troca de informações de segurança está sendo feito pelo IETF (*Internet Engineering Task Force*). Como resultado, foi especificado o formato IDMEF (*Intrusion Detection Message Exchange Format*) [Debar et al. 2006] que, apesar de não ter sido padronizado ainda, vem ganhando popularidade nos meios científico e comercial. O *Snort* [Roesch 1999], o DOMINO [Yegneswaran et al. 2004], a família de IDSs STAT [Vigna et al. 2000] e a proposta de [Park et al. 2003] usam ou estendem o formato IDMEF para atender as suas necessidades. Contudo, a segurança das comunicações não é prioridade nestes IDSs.

Outro IDS que utiliza o IDMEF é o *Prelude* [Zaraska 2003], que é um IDS

híbrido[2], no qual um ou mais gerentes recebem, estabelecem correlações e ainda podem retransmitir alertas gerados por sensores de diversos tipos, distribuídos em uma rede de computadores. Infelizmente, os alertas IDMEF não são transmitidos na linguagem XML, como define o padrão, optando por um formato nativo.

Em [Yasinsac 2000] é discutida superficialmente a infra-estrutura necessária para a comunicação segura entre dispositivos de segurança. A especificação do formato IDMEF não define mecanismos de segurança para a troca de mensagens, porém, seus requisitos [Wood 2002] especificam os critérios de segurança que devem ser adotados para o transporte das mensagens. Estes critérios incluem a confidencialidade, integridade e o não repúdio das mensagens IDMEF. A forma mais comum de prover segurança às mensagens IDMEF é o uso do protocolo SSL (*Secure Socket Layer*), como faz o *Prelude*, por exemplo. Porém, este mecanismo não é suficiente para a segurança fim a fim das mensagens transmitidas entre domínios administrativos distintos e protegidos, nos quais os serviços de segurança não podem ser acessados diretamente por clientes externos. Nossa proposta visa exatamente oferecer segurança às composições de IDSs que atuam neste tipo de ambiente.

Infelizmente, todo mecanismo de segurança envolve custos e a análise de desempenho em IDSs não é uma tarefa trivial. Alguns trabalhos, como [Mutz et al. 2003] e [Antonatos et al. 2004] reportam os modelos de testes e as metodologias para *benchmarking* de sistemas de detecção de intrusão. Porém, tais modelos não estão preocupados com o desempenho ou os custos de comunicação. Em [Pereira et al. 2005] é apresentada uma comparação entre o uso do SNMP e o SMTP com SOAP para a notificação de eventos de gerenciamento, demonstrando que o uso do SOAP implica maior tráfego na rede.

3. Composição de IDSs usando Web Services

A composição de IDSs é uma nova abordagem para construção de sistemas de detecção de intrusão de larga escala. Para isso, IDSs completos ou elementos de detecção de intrusão interagem entre si para compor um sistema de monitoramento muito mais amplo e diversificado do que o obtido com o emprego dos IDSs tradicionais (distribuídos ou não). As composições podem ser criadas para atender as necessidades de segurança de uma única organização ou se estender por diferentes organizações que desejam compartilhar informações e alertas de segurança [Brandão et al. 2006a, Brandão et al. 2006b].

São adotados como elementos básicos de um IDS, aqueles definidos no modelo de detecção de intrusão do IDWG/IETF [Wood 2002]: sensores, analisadores e gerentes. IDSs completos podem agir como sensores ao enviarem dados a outros elementos da composição ou como analisadores, ao receberem e efetuarem a análise dos dados provenientes de elementos distribuídos (sensores). Como o conceito de Composição de IDSs comporta que um mesmo elemento de IDS possa ser integrado a mais de uma composição, o papel de cada elemento irá depender da forma como os mesmos são organizados em cada composição. Para permitir a interação entre os elementos de uma composição de IDSs foi criada uma infra-estrutura de serviços, seguindo a arquitetura orientada a serviços suportada pela tecnologia de *Web Services* [Booth et al. 2004], com o amplo emprego de XML [Bray et al. 2000]. A criação e o gerenciamento das composições utiliza o conceito de orquestração de serviços, conforme descrito em [Brandão et al. 2006b].

[2]IDSs que utilizam sensores de rede e de *host*.

Neste modelo de composição, os elementos de detecção de intrusão são apresentados para composição como *Web Services*. Para que uma composição seja criada são necessárias diversas informações sobre os elementos oferecidos, como suas características, localização, protocolos de comunicação e interfaces. Tais informações são apresentadas e acessadas através do Serviço de Registro e Pesquisa (SRP), que faz parte da infra-estrutura de serviços do modelo. O SRP está fundamentado na especificação UDDI (*Universal Description, Discovery and Integration specification*) [Clement et al. 2004]. As interfaces de cada *Web Service* são descritas em um formato processável, fornecido pela linguagem WSDL (*Web Services Description Language*) [Chinnici et al. 2006] e a localização desta descrição também é mantida no SRP. Para amenizar o problema do SRP ser um ponto único de falha no sistema, foi previsto a sua replicação.

O modelo de comunicação desenvolvido define elementos "serviços" a partir de partes dos IDSs convencionais, através de funções que formam o que chamamos de "Compatibilizador de Formatos", cujo papel principal é tratar com os formatos usados nas trocas de mensagens em composições de serviços. Estas funções, por exemplo, atuam na intermediação da comunicação entre partes usadas de IDSs convencionais e suas formas em *Web Services* que tornam as mesmas disponíveis para composições de IDSs.

No modelo, um Serviço de Segurança presta serviços que auxiliam na autenticação e no controle de acesso dos elementos envolvidos na composição. O Serviço de Segurança também está baseado na UDDI.

4. Comunicação Interoperável e Segura nas Composições de IDSs

Para prover um serviço de comunicação seguro entre elementos de IDSs, distribuídos em ambientes de larga escala como a Internet, fazemos uso de um conjunto de serviços que estão fundamentados em padrões e conceitos da tecnologia de *Web Services*. O uso desta tecnologia é viável nestes ambientes de larga escala e permite que se consiga a interoperabilidade e a integração entre IDSs. As comunicações nas composições podem envolver, em domínios administrativos separados, sensores e analisadores, sensores e gerentes, analisadores e gerentes, somente analisadores ou ainda, somente gerentes. Em todas estas comunicações são adotados mecanismos e padrões de segurança.

O modelo de comunicação e integração de IDSs proposto possui algumas vantagens sobre os outros modelos existentes. Neste é possível realizar a integração de qualquer elemento de IDS existente no mercado ou de desenvolvimento próprio. A comunicação ocorre de maneira segura e utilizando um formato de alerta padrão, como o IDMEF. O formato dos alertas gerados por cada IDS é convertido para este formato padrão, possibilitando a interação com composições de IDSs.

A figura 1 apresenta um exemplo de composição de IDSs em ambientes de larga escala. Este exemplo ilustra duas composições de IDSs (*CIDS1* e *CIDS2*) utilizando quatro domínios administrativos distintos (*Domínio A*, *Domínio B*, *Domínio C* e *Domínio D*). A composição *CIDS1* é formada pelo gerente GA1 do domínio A, pelo analisador AC1 e o sensor SC1 do domínio C e pelo sensor SD1 do domínio D. Este último sensor também faz parte da composição *CIDS2* que contém ainda: o sensor SD2 e o analisador AD1 do domínio D, o sensor SB1 e o gerente GB1 do domínio B. As setas, na figura, indicam o sentido das mensagens. Estas duas composições também trocam entre si mensagens SOAP através de seus gerentes.

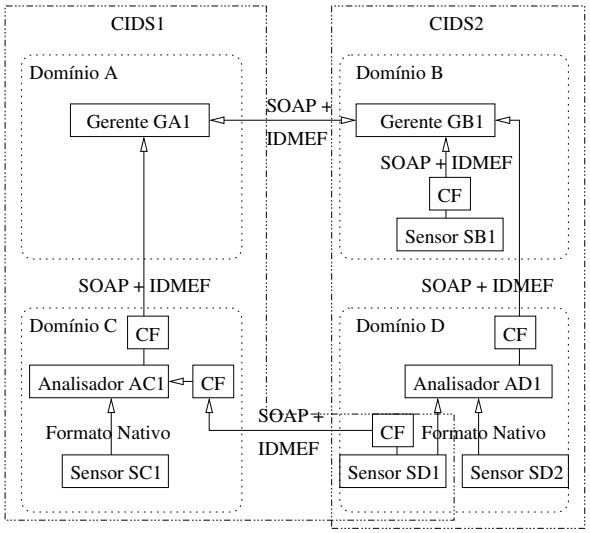


Figura 1. Exemplo de comunicação em composições de IDSs

4.1. Interoperabilidade na Troca de Mensagens

No nosso modelo, mensagens que envolvem comunicações entre componentes de domínios diferentes, pertencentes ou não à mesma composição, utilizam-se sempre de alertas no formato IDMEF encapsulados em mensagens SOAP. Estas mensagens são montadas a partir do que chamamos "Compatibilizador de Formatos" (*CF*). Estes "Compatibilizadores de Formatos" fazem a tradução das mensagens do formato nativo de qualquer sensor ou analisador para o formato IDMEF em XML[3] [Debar et al. 2006] encapsulado em SOAP. O *CF* permite a integração de diversos IDSs existentes através da tecnologia de *Web Services*. Os gerentes das composições são admitidos como sendo serviços *Web* no modelo proposto e, portanto, "entendem" o formato IDMEF em XML encapsulado em SOAP e não precisam fazer uso de *CFs*. Porém, comunicações envolvendo outros componentes de IDSs em outros níveis podem envolver *CFs* nos dois extremos de uma comunicação. No caso de surgir um novo formato de registro de alerta, que ainda não é suportado pelo *CF*, basta adicionar uma nova função ao código do *CF* para realizar a sua conversão.

A comunicação dentro de um domínio pode acontecer de duas maneiras: utilizando o formato nativo[4] ou através do Compatibilizador de Formatos. Na figura 1, por exemplo, o sensor SD1 pode comunicar-se com o analisador AD1 utilizando o formato nativo e com outro analisador externo ao seu domínio (analisador AC1) fazendo uso do seu *CF*.

Na comunicação entre os elementos de detecção de intrusão e o *CF* pode ser usado o formato IDMEF, ou algum outro formato (nativo do elemento) suportado pelo *CF*, com um canal de comunicação seguro utilizando SSL.

[3]O IDMEF em XML é o formato padrão, segundo o IDWG/IETF
[4]Formato próprio, adotado pelo sensor ou analisador de um IDS

4.2. Segurança na Comunicação

Neste modelo proposto, seguimos o padrão *WS-Security* [Nadalin et al. 2004] para assinatura (*XML Signature* [Reagle 2000, Eastlake et al. 2002]) e cifragem (*XML Encryption* [Reagle 2002, Imamura et al. 2002]) de mensagens encapsuladas em SOAP. Contudo, é necessário o uso de um repositório de certificados que contém as chaves públicas dos elementos de IDSs. Este repositório localiza-se no Serviço de Registro e Pesquisa (SRP), descrito na seção 3. Os certificados também podem ser mantidos em outros repositórios e sua localização é apontada no registro dos elementos de IDS armazenados no SRP.

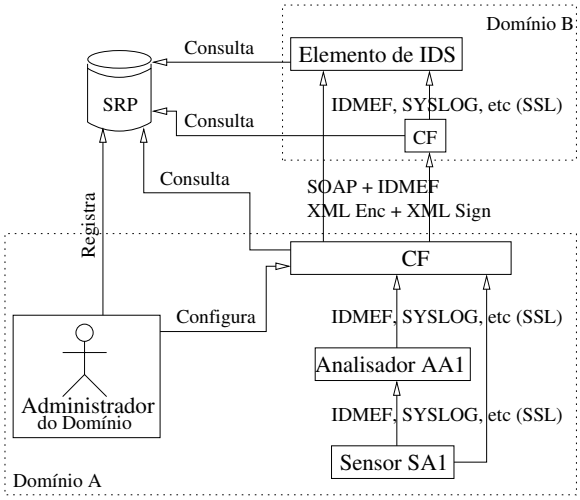


Figura 2. Detalhes do modelo de comunicação segura

A figura 2 especifica com alguns detalhes o modelo de comunicação proposto para as interações em composições de IDSs. Nesta figura, é ilustrado um domínio A com componentes de uma composição de IDSs onde um sensor e analisador fazem uso do Compatibilizador de Formatos[5] para se comunicarem com o elemento de IDS no domínio B. A comunicação entre o sensor e o analisador no domínio A é feita usando o formato nativo, suportado por ambos, ou através do *CF*.

No domínio A, introduzimos o papel do Administrador do Domínio que possui, entre outras, a função de registrar (operação Registra na figura 2) no SRP os seus elementos de detecção de intrusão, que serão acessados através de seus respectivos *CFs*. Uma vez registrados, estes elementos ficarão disponíveis para o uso em composições de IDSs. O serviço de registro vê estes elementos e seus *CFs* como Serviços *Web*. Outra função do administrador do domínio é configurar (operação Configura na figura 2) o Compatibilizador de Formatos (*CF*), informando em qual formato ele receberá as mensagens. O administrador obtém junto ao Serviço de Segurança um *token* que será usado para executar o processo de composição de serviços (orquestração) e interagir com os elementos de detecção de intrusão que serão adicionados à composição.

Durante a criação de uma composição é necessário que se estabeleçam associações entre *CFs* de elementos de IDSs de domínios diferentes. Ao ser configurado para atuar

[5]Na figura 2, o *CF* representa um papel e não um único componente compartilhado. Cada elemento possui seu próprio CF.

em uma composição, o elemento recebe informações dos elementos com os quais irá interagir. Entre estas informações estão os apontadores que serão usados para obter os certificados de seus parceiros, como faz o *CF* do domínio A (na figura 2). Usando a comunicação segura, mensagens SOAP, assinadas e cifradas são enviadas do domínio A para outro domínio (Domínio B, no caso da figura 2).

No exemplo desta figura, é mostrada a comunicação entre o sensor SA1 do domínio A e o *Elemento de IDS* no domínio B, que na estrutura de um IDS pode ser um analisador ou um gerente. Se o elemento em B for um gerente, a comunicação do sensor SA1 com este elemento deve fazer uso de um *CF* no domínio A. No outro lado da comunicação, no domínio B, não será necessária a recuperação de padrões nativos a partir da mensagem SOAP encapsulando o IDMEF. O gerente, que em nosso modelo é totalmente baseado em *Web Service*, recebe mensagens SOAP sem a necessidade de um *CF* no domínio B. Para se habilitar às comunicações seguras, o gerente deverá também consultar ao SRP (operação Consulta) para obter o certificado do emissor da mensagem e, então, poderá receber, decifrar, verificar assinatura e processar as mensagens recebidas.

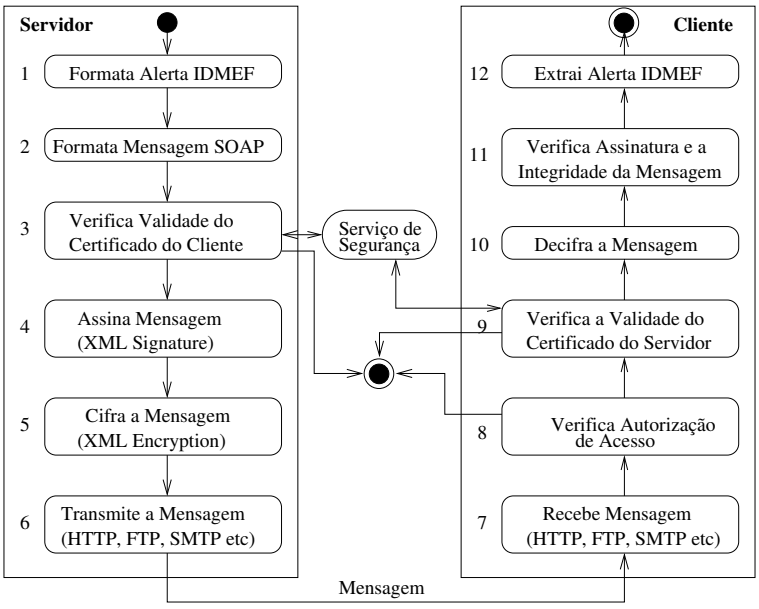


Figura 3. Processo de Comunicação Segura

O processo de comunicação segura é ilustrado na Figura 3. Neste processo, o elemento de detecção de intrusão servidor envia alertas a um elemento de IDS cliente. O elemento servidor formata o alerta IDMEF (1) e o encapsula em uma mensagem SOAP (2). Em seguida, verifica se o certificado do cliente é válido (3). O certificado do cliente é obtido pelo servidor no momento de sua ativação na composição, conforme será descrito na próxima seção. Se o certificado perdeu sua validade ou se o tempo entre a última verificação do mesmo for maior do que o definido pelo administrador, o Serviço de Segurança é consultado para a obtenção de um certificado válido. De posse do certificado do cliente, o servidor assina a mensagem usando o padrão *XML-Signature* (4). Uma vez assinada, a mensagem é cifrada usando o padrão *XML-Encryption* (5). A mensagem é então transmitida usando um protocolo de transporte padronizado (HTTP, FTP, SMTP etc) (6). O cliente, ao receber a mensagem (7), verifica se o servidor que a enviou possui

autorização para interagir com o cliente (8). Se a mensagem não for autorizada, ela é descartada. Como ocorre com o servidor, o cliente verifica a validade do certificado do servidor, obtido durante sua ativação na composição (9). Caso seja necessário, um novo certificado é obtido junto ao Serviço de Segurança. De posse do certificado do servidor, o cliente decifra a mensagem (10) e verifica se a mesma é autêntica e está íntegra (11). Finalmente, é extraído da mensagem o alerta IDMEF (12).

4.3. Protótipo

Foi desenvolvido um protótipo, seguindo o modelo proposto para a comunicação segura e interoperável que permitisse a integração de IDSs. O protótipo está centrado no Compatibilizador de Formatos (*CF*), implementado fazendo uso da linguagem de programação Java, no qual os métodos para assinatura e cifragem de documentos XML são implementados seguindo os padrões *XML-Signature* e *XML-Encryption*. O *CF* se comunica com um sensor baseado no *Prelude*[Zaraska 2003].

Neste protótipo, utilizamos o protocolo HTTP para o transporte das mensagens SOAP, por ser o mais empregado em *Web Services* e por possibilitar o livre tráfego de mensagens em ambientes de larga escala através de servidores *Web*, mesmo na presença de *firewalls*. Na seção 5, avaliamos o desempenho do mecanismo de comunicação usando o protocolo HTTP com os mecanismos de segurança propostos.

A implementação de gerentes neste protótipo segue a arquitetura orientada a serviços, baseada na tecnologia de *Web Services* e na aplicação dos padrões de segurança *WS-Security* [Nadalin et al. 2004]. Para a hospedagem dos serviços, utilizou-se o servidor *Apache Tomcat* versão 5.5.12 e a ferramenta *BEA Weblogic Workshop*[6] versão 8.1. No Serviço de Registro e Pesquisa (SRP) utilizamos a ferramenta *BEA WebLogic Server UDDI Registry*[7] que contém a implementação da UDDI versão 3.

5. Testes e Resultados Obtidos

Visando verificar a eficiência e os custos computacionais do modelo de comunicação interoperável e segura, foram realizados dois experimentos. No primeiro experimento foram feitos testes comparativos, em ambiente controlado, entre o modelo proposto e o modelo de comunicação nativo do *Prelude*. O segundo experimento envolveu a composição de IDSs entre organizações distintas, envolvendo ambientes reais e o uso da Internet. Em ambos os testes usamos o protótipo descrito na seção 4.3.

5.1. Testes em Ambiente Controlado

A figura 4 mostra a disposição das máquinas para os testes comparativos realizados no primeiro experimento. No ambiente proposto para os testes foram caracterizados dois domínios administrativos através de duas sub-redes (rede A e rede B, na figura 4). Neste ambiente de testes, um computador executa o sensor de rede do *Prelude* que está conectado nas duas redes (A e B). Um outro computador presente na rede B executa o gerente que receberá os alertas gerados pelo sensor. Outros dois computadores disponíveis na rede A executam ataques ao sensor. Para efetuar os ataques, utilizou-se a ferramenta

[6]http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/Weblogic/workshop
[7]http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/server

Idswakeup [8], que utiliza a base de assinaturas do *Snort* para gerar ataques que serão identificados pelo IDS.

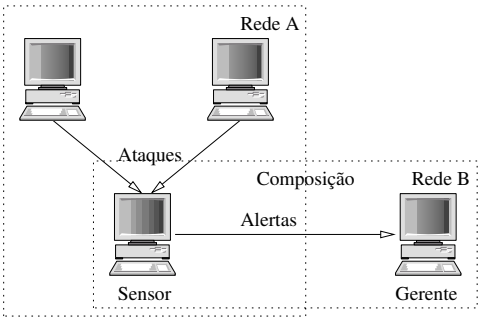


Figura 4. Disposição das máquinas

Na concretização do ambiente de testes do primeiro experimento foram ainda usados quatro computadores modelo AMD Athlon XP 2600+ contendo 512MB de memória RAM e placas de rede de 100Mbits/s. Foi instalado o sistema operacional Ubuntu Gnu/Linux versão 5.10.

Apesar de ilustrar as iterações típicas entre elementos de uma composição de IDSs, o ambiente de testes representado pela figura 4 não reflete um ambiente realmente de larga escala por dois motivos:

1. O sistema de comunicação nativo do IDS (cenário 1, descrito a seguir), por fazer uso do SSL, tem dificuldades para comunicação em ambientes de larga escala, sendo necessário a liberação de portas específicas em *firewalls*, dificultando a realização dos testes em tais ambientes;
2. Os resultados dos testes (tempos de transmissão, por exemplo), com o modelo proposto, seriam prejudicados por causa da alta latência das redes nestes ambientes e pela interferência de outros tipos de tráfego.

Contudo, os testes realizados são válidos para avaliar os custos de comunicação entre as tecnologias envolvidas em um ambiente considerado "ideal", desconsiderando a latência e outras dificuldades na comunicação em ambientes de larga escala.

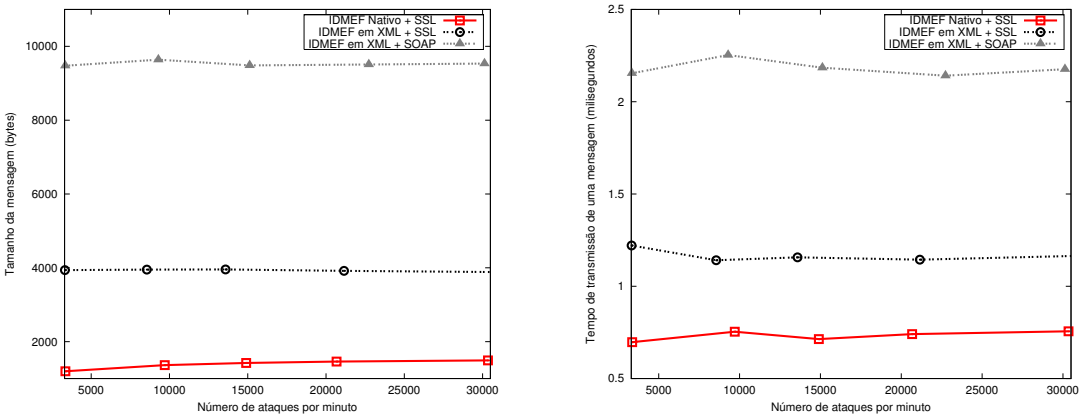
Na avaliação dos resultados são descritos três cenários de testes que permitiram obter as comparações desejadas:

- **Cenário 1** (testes com o sistema de comunicação nativo): os alertas são transmitidos através de um canal SSL, usando o formato de comunicação original do Prelude.
- **Cenário 2** (testes com o sistema de comunicação com IDMEF em XML): os alertas são convertidos para o formato IDMEF em XML e transmitidos ao gerente através de um canal SSL.
- **Cenário 3** (testes com o sistema de comunicação segura e interoperável): os alertas seguem o modelo de comunicação proposto, com mensagens SOAP protegidas segundo os padrões *WS-Security*, utilizando o algoritmo *tripledes-cbc* e chaves de 512 *bits*, garantindo, desta forma, a segurança das mensagens sem causar um impacto tão grande no desempenho.

[8] http://www.hsc.fr/ressources/outils/idswakeup

O objetivo dos testes foi avaliar os custos de desempenho associados ao uso de mecanismos de segurança propostos. Para isso, determinamos o tamanho médio das mensagens geradas por cada sistema de comunicação, identificamos o tempo e a capacidade de transmissão de alertas nos três cenários descritos acima além de verificar, em diferentes situações de ataque, os tempos para processar e gerar alertas nos três formatos.

Os testes foram realizados variando o número de ataques efetuados ao sensor. Foi utilizado o software *Tcpdump* para coleta de dados das interfaces de rede das máquinas. Através da análise destes dados, foram obtidos o tamanho médio das mensagens, o tempo médio de transmissão das mensagens e o tempo médio de processamento dos alertas.



(a) Tamanho médio das mensagens (b) Tempo médio para transmissão de mensagens

Figura 5. Testes comparativos: tamanho da mensagem e tempo de transmissão

A Figura 5(a) mostra um gráfico com o tamanho médio das mensagens utilizando os três formatos para a comunicação com o gerente: IDMEF nativo com SSL (*IDMEF Nativo + SSL*), IDMEF em XML com SSL (*IDMEF em XML + SSL*) e IDMEF em XML com SOAP (*IDMEF em XML + SOAP*). Neste gráfico observou-se que o tamanho das mensagens de alerta permanece relativamente constante em 9500 bytes para mensagens no formato IDMEF em XML com SOAP, 4000 bytes para mensagens no formato IDMEF em XML com SSL e 1500 bytes para mensagens no formato IDMEF nativo com SSL.

Em outra análise observou-se o tempo de transmissão entre o Sensor sob ataque e o Gerente que recebe os alertas correspondentes. Como esses dois computadores foram interligados por uma rede independente (rede B), esta comunicação não sofreu interferências e pode-se observar que o tempo de transmissão permaneceu constante. O *link* de 100Mbits/seg não ficou saturado mesmo quando o sensor sofreu muitos ataques. A Figura 5(b) ilustra o gráfico com os tempos de transmissão.

Em uma terceira análise, observou-se o tempo de processamento e resposta do sistema, ou seja, o tempo médio entre o início dos ataques e o instante em que as mensagens de alerta são enviadas. O tempo "ocioso" do sistema reduz até chegar no seu ponto de saturação. Quanto mais perto do ponto de saturação do sistema, mais alertas são processados por unidade de tempo e menor será o tempo médio para gerar um alerta. A Figura 6 mostra o gráfico desta análise. Este tempo de processamento depende do número de ataques que chegam e são detectados e do poder de processamento do sistema. Observou-se que o tempo é maior com poucos ataques acontecendo ou quando o sistema está saturado

(recebe mais do que sua capacidade de processamento). No caso do sistema desenvolvido que utiliza o padrão *WS-Security* para atender os aspectos de segurança em ambientes abertos, o ponto de saturação ficou em 15000 ataques por minuto. A partir deste valor, o número de ataques foi maior do que este sistema pode processar. Ao chegar em 30000 ataques por minuto este sistema entrou em colapso e parou. Tal falha ocorre devido ao estouro de memória na máquina virtual *Java,* configurada pelo Servidor *Tomcat*, que pode ser atenuada com depurações na configuração.

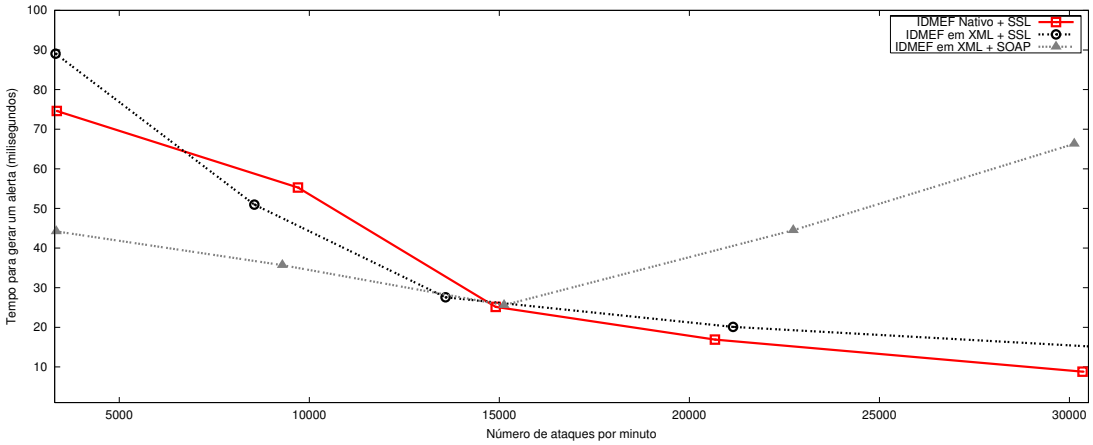


Figura 6. Tempo médio para reagir e gerar um alerta

5.2. Teste em ambiente de larga escala

Foram aplicados testes com o protótipo desenvolvido em um ambiente de larga escala, visando comprovar o funcionamento do modelo nestes ambientes. Nestes testes foram utilizadas duas redes dispersas geograficamente, as redes do DAS (Departamento de Automação e Sistemas - UFSC) e do IPEA (Instituto de Pesquisa Econômica Aplicada - DF). A figura 7 apresenta a disposição das máquinas nestas redes. Neste teste, o sensor (A) localizado na rede do IPEA, que recebe ataques provenientes do computador (C) localizado na rede do DAS. O gerente (B), localizado na rede do DAS, recebe e processa os alertas gerados pelo sensor (A).

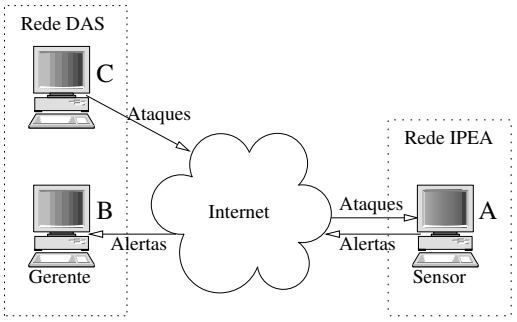


Figura 7. Disposição das máquinas

O sensor recebeu, identificou e gerou alertas de ataques originados do computador (C). Como resultado desses testes, o tempo médio de transmissão das mensagens de alerta

foi de 4,403 milisegundos. Neste tempo foi desconsiderado o atraso médio entre as redes, que é de 51,126 milisegundos. Fazendo uma comparação com os resultados obtidos anteriormente, observamos um acréscimo de 100% no tempo de transmissão de uma mensagem, o que é razoável para um ambiente como a Internet. Os tempos de transmissão máximo e mínimo de uma mensagem foram respectivamente 8,719 e 4,023 milisegundos. Estas variações são consideradas normais para ambientes de larga escala. Os outros quesitos analisados (tamanho médio das mensagens e tempo médio para gerar um alerta) não apresentaram variações quando comparados aos testes anteriores pois não dependem do ambiente de comunicação. Os outros modelos analisados previamente (cenários 1 e 2 da seção 5) não conseguem se comunicar nestes ambientes e portanto não foram testados.

5.3. Considerações sobre os experimentos

Utilizando o protocolo nativo ou o IDMEF em XML, o sistema apresenta melhor desempenho pois nestes casos, é feito *cache* em disco, adiando o seu ponto de saturação. Uma das razões para este desempenho inferior do protótipo desenvolvido está no uso do protocolo HTTP. Para cada mensagem SOAP a ser enviada é estabelecida uma nova conexão, enquanto no sistema nativo com o uso do SSL, uma única conexão é usada para o transporte de todas as mensagens entre os pares comunicantes.

O tamanho das mensagens SOAP é maior e os custos das conversões de formatos também se fazem sentir. Primeiro o alerta é convertido do IDMEF binário usado pelo *Prelude* para o formato padronizado IDMEF em XML. Depois esta informação passa pelo encapsulamento da mensagem SOAP, com a inclusão da respectiva assinatura. Com isto, o tempo para transmissão de uma mensagem utilizando o modelo proposto é maior pois, além da mensagem ser maior, existe a necessidade de uma nova conexão TCP para cada mensagem de alerta a ser transmitida. O sistema proposto satura mais rápido pois o processamento para converter as mensagens, assiná-las e cifrá-las é maior. Contudo, o uso de *Web Services*, IDMEF e padrões de segurança permite a integração de diversos sistemas de segurança em redes distintas, além da transposição de *firewalls* e barreiras que existem entre domínios administrativos diferentes, o que não é possível usando outros mecanismos de comunicação e segurança.

O resultado do segundo experimento comprovou o funcionamento deste modelo em ambientes distribuídos e de larga escala, fazendo uso da Internet. Analisando a escalabilidade do sistema desenvolvido, é possível afirmar que a adição de mais elementos na composição não é um grande problema. Contudo, o número de alertas por minuto que este sistema pode processar é um fator limitante (Figura 6).

6. Conclusões

Este artigo descreve um modelo de comunicação segura para a integração de IDSs em ambientes de larga escala. Neste texto explicitamos as características do modelo, descrevemos um protótipo e realizamos testes sobre o modelo, analisando os custos de comunicação envolvidos. O modelo de comunicação desenvolvido visa manter a segurança das mensagens, utilizando padrões e garantindo a segurança fim a fim das mesmas, permitindo a interoperabilidade entre os diversos sensores, analisadores e gerentes da composição. Este novo modelo possibilita ainda a integração de IDSs, permite a troca de informações de segurança entre organizações e permite a criação de sistemas

distribuídos em ambientes de larga escala, o que não é possível utilizando os sistemas IDSs nativos isoladamente. Este trabalho possibilita a integração de IDS existentes para formar um sistema maior, mais complexo e preciso para identificação, por exemplo, de ataques distribuídos aos sistemas computacionais.

Os resultados obtidos com os testes mostraram que o modelo de comunicação desenvolvido teve um desempenho inferior ao sistema de comunicação nativo do IDS quando está sob intenso ataque, porém o desempenho do sistema é satisfatório.

Em trabalhos futuros, pretende-se avaliar a viabilidade do uso de outros protocolos como o SMTP no lugar do HTTP para transporte das mensagens SOAP, bem como modificar o *CF* para adicionar o suporte a novos formatos de mensagens através de *plug-ins*, sem precisar modificar o seu código fonte.

Referências

Antonatos, S., Anagnostakis, K., Polychronakis, M., and Markatos, E. (2004). Performance analysis of content matching intrusion detection systems. In *In Proceedings of the 4th IEEE/IPSJ Symposium on Applications and the Internet (SAINT)*.

Bass, T. (2004). Service-oriented horizontal fusion in distributed coordination-based systems. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 615– 621, Monterey, CA.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., and Ferris, C. (2004). Web Services Arquiteture. URL `http://www.w3.org/TR/ws-arch/`.

Brandão, J. E., da Silva Fraga, J., and Mafra, P. M. (2006a). A new approach for ids composition. In *Proceedings of the 2006 IEEE International Conference on Communications (ICC), Istambul*.

Brandão, J. E. M. S., Fraga, J. S., Mafra, P. M., and Obelheiro, R. R. (2006b). A WS-Based Infrastructure for Integrating Intrusion Detection Systems in Large-Scale Environments. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, volume 4275 of *Lecture Notes in Computer Science*, pages 462–479, Montpellier, France. Springer Berlin / Heidelberg.

Bray, T., Paoli, J., Sperberg-McQueen, C., and Maler, E. (2000). Extensible markup language (xml). URL `http://www.w3.org/TR/REC-xml`.

Chinnici, R., Moreau, J.-J., Ryman, A., and Weerawarana, S. (2006). Web services description language (wsdl) version 2.0 part 1: Core language. Technical report, W3C. URL `http://www.w3.org/TR/wsdl20/`.

Clement, L., Hately, A., von Riegen, C., and Rogers, T. (2004). Oasis uddi spec technical committee. URL `http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf`.

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

Eastlake, D., Reagle, J., and Solo, D. (2002). Xml-signature syntax and processing. rfc 3275. Technical report, National Institute of Standards and Technology.

Imamura, T., Dillaway, B., and Simon, E. (2002). Xml encryption syntax and processing. Technical report, Department of the Navy XML Registry.

Leu, F.-Y., Lin, J.-C., Li, M.-C., Yang, C.-T., and Shih, P.-C. (2005). Integrating grid with intrusion detection. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 304–309.

Mutz, D., Vigna, G., and Kemmerer, R. (2003). An Experience Developing an IDS Stimulator for the Black-Box Testing of Network Intrusion Detection Systems. In *Proceedings of the 2003 Annual Computer Security Applications Conference*, Las Vegas.

Nadalin, A., Kaler, C., Hallam-Baker, P., and Monzillo, R. (2004). Web services security: Soap message security 1.0. URL `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-se%curity-1.0.pdf`.

Park, S.-K., Kirn, K.-Y., Tang3, J.-S., and Noh, B.-N. (2003). Supporting interoperability to heterogeneous ids in secure networking framework. In *Asia-Pacific Conference on Communication (APCC)*, pages 844–848.

Pereira, E. D., Granville, L. Z., Almeida, M. J., and Tarouco, L. M. (2005). Avaliação de suporte de notificações utilizando snmp e web services em uma arquitetura de correlação de eventos distribuída. In *Anais do 23o Simpósio Brasileiro de Redes de Computedores (SBRC)*, pages 1–14. SBC.

Reagle, J. (2000). Xml signature requirements. request for comments 2807. Technical report, Massachusetts Institute of Technology Laboratory for Computer Science.

Reagle, J. (2002). Xml encryption requirements. note 04. Technical report, Massachusetts Institute of Technology Laboratory for Computer Science.

Roesch, M. (1999). Snort - lightweight intrusion detection for networks. In *Proceedings of USENIX LISA, Berkeley*, pages 229–238.

Teo, L., Zheng, Y., and Ahn, G. (2003). Intrusion detection force: An infrastructure for internet-scale intrusion detection. In *Proceedings of the first IEEE International Workshop on Information Assurance*.

Tolba, M. F., Abdel-Wahab, M. S., Taha, I. A., and Al-Shishtawy, A. M. (2005). Toward enabling grid intrusion detection systems. In *Proceedings of the 5th IEEE International Symposium on Cluster Computing and the Grid*, pages 537–540.

Vigna, G., Eckmann, S., and Kemmerer, R. (2000). The stat tool suite. In *Proceedings of Information Survivability Conference and Exposition, IEEE Press*, pages 46–55.

Wood, M. (2002). Intrusion detection message exchange requirements. URL `http://www.ietf.org/internet-drafts/draft-ietf-idwg-requirements-10.txt%`.

Yasinsac, A. (2000). Active protection for secure security services. Technical Report TR–000101, Department of Computer Science, Florida State University.

Yegneswaran, V., Barford, P., and Jha, S. (2004). Global intrusion detection in the domino overlay system. In *Network and Distributed System Security Simposium*.

Zaraska, K. (2003). Prelude ids: Current state and development perspectives. URL `http://www.prelude-ids.org/download/misc/pingwinaria/2003/paper.pdf`.