Uma Abordagem para Gerência Baseada na *Web* utilizando *Web Services* para configuração de Regras no Mecanismo de Filtro de Pacotes Netfilter

Marcio Pohlmann[1], Weldson Queiroz de Lima[2]

[1]Curso de Pós-Graduação em Gerência e Aplicação em Redes de Computadores – Universidade Luterana do Brasil (ULBRA) - Martinho Lutero, 301 – 96.505-595 – Cachoeira do Sul – RS – Brazil

[2]Curso de Pós-Graduação em Gerência e Aplicação em Redes de Computadores
Analista de Redes do Serviço Federal de Processamento de Dados - SERPRO
Mestre em Engenharia Elétrica pela UFRN

marcpohl@gmail.com, weldsonlima@yahoo.com.br

Resumo: *Este artigo descreve a composição de um novo modelo de gerenciamento distribuído para a configuração de regras em um sistema de filtro de pacotes de forma segura por meio da tecnologia Web Services. Para validar esta abordagem, foi proposta a configuração de todos os serviços necessários e a criação de uma aplicação Web para inferência de regras via navegador em um servidor com Netfilter IPTables, utilizando as bibliotecas NuSOAP e gSOAP.*

1. Introdução

As atividades de gerenciamento têm agregado, ao longo dos tempos, um número cada vez maior de serviços e aplicações. Nestas entidades, as tarefas de configuração, controle de falhas, monitoração de desempenho, segurança e contabilização geram informações úteis para tomadas de ações e intervenção do gerente de redes.

Todavia, dados de gerência, quando não fiscalizados, constituem uma ameaça à estrutura dos componentes interconectados, pois agentes maliciosos de posse de informações da rede podem comprometer seus sistemas e serviços. No quesito segurança lógica, encontramos aplicações que separadamente visam prevenir, detectar e reagir a determinados eventos da rede, como o mecanismo de *firewall* com a técnica de filtro de pacotes, uma solução para controle dos fluxos e conexões de rede [Zwicky 2000].

Aplicações distribuídas de gerenciamento baseado na *Web* têm sido disponibilizadas para equipamentos e serviços *Internet*, originando tecnologias na composição de serviços para arquiteturas distribuídas e sistemas heterogêneos. Neste sentido, *Web Services* tem se mostrado uma interessante solução no desenvolvimento de componentes de software de serviços interativos na *Web*, tais como processos de e-commerce, redes de suporte a grids, inteligência artificial e gerenciamento de redes. [Vianna 2006] e [Coulouris 2005]

Na ótica da segurança, entretanto, a distribuição do controle e configuração de serviços de *Internet* através de uma interface de navegador necessita de atenção às informações de gerência, pois as mensagens trocadas entre cliente e servidor devem

estas ser conhecidas somente entre as duas entidades. Em muitos casos, aplicativos de gerência baseada na *Web* não fornecem mecanismos seguros nas transações entre gerente e objeto(s) gerenciado(s).

O objetivo deste trabalho é apresentar o desenvolvimento teórico e prático de um novo modelo de gerenciamento para a configuração de regras em um sistema de filtro de pacotes de forma segura através de *Web Services*. Para validação desta abordagem, foi criada uma aplicação *Web* para inferir regras via navegador no filtro de pacotes Netfilter, utilizando a interação das bibliotecas NuSOAP e gSOAP.

2. Fundamentação Teórica

O gerenciamento de redes, é descrito por [Lima 2005] como o ato de iniciar, monitorar e modificar a operação das funções que apóiam as necessidades dos usuários de redes de computadores. Esta atividade é fundamental para o crescimento ordenado, funcionamento satisfatório e distribuição uniforme dos recursos de rede. Para [Kurose 2003], o gerenciamento de rede exige a capacidade de monitorar, testar, consultar, configurar e controlar o *hardware*, o *software* e os componentes de uma rede.

A ISO (*International Standards Organization*) definiu cinco áreas de gerenciamento de redes, descritas por [Lopes 2003] como gerência de configuração, gerência de falhas, gerência de desempenho, gerência de segurança e gerência de contabilidade. Uma arquitetura de gerenciamento de redes clássica é composta basicamente por entidade gerenciadora (gerente), dispositivo gerenciado (com agente) e protocolo de gerenciamento, conforme [Kurose 2003], [Dantas 2002] e [Lopes 2003]. A solução para padronização de gerenciamento mais conhecida e utilizada é baseada no protocolo SNMP (*Simple Network Management Protocol*), por ser concebido em um período de demanda de gerenciamento de dispositivos, com ampla aceitação pela comunidade e sua simplicidade [Kurose 2003].

Em concordância, [Dantas 2002] afirma que o protocolo é apenas um dos elementos do modelo de gerenciamento, que é composta ainda por uma estação de gerenciamento, agentes e uma base de informações de gerenciamento (*Management Information Base* - MIB). Atualmente, o SNMP dispõem de três versões nomeadas SNMP, SNMPv2 (adição de novas mensagens e melhoria da comunicação gerente-gerente) e SNMPv3 (provendo serviços de autenticação, privacidade e controle).

Em contrapartida, novos modelos de gerenciamento de redes vêm sendo desenvolvidos, visando atender serviços específicos ou diferenciados não suportados pelo SNMP, numa lista não exaustiva que inclui agentes móveis, gerenciamento baseado em políticas (*Policy-based Network Management* - PBNM), gerenciamento baseado na *Web* e *Web Services*. Este último é descrito por [Lima 2005] como sendo um conjunto de módulos de software capazes de realizar operações que podem ser encontradas e invocadas remotamente através da *Internet* por meio de protocolos de comunicação, transportado sob protocolos largamente utilizados na rede mundial de computadores.

Esta tecnologia surgiu de um consórcio de empresas lideradas pela Microsoft e IBM, tornando-se hoje um padrão do W3C (*World Wide Web Consortium*). A

comunicação com *Web Services* é baseada, segundo [Lima 2005] nos protocolos SOAP (*Simple Object Access Protocol*) ou RPC-XML, o que garante simplicidade e padronização nas trocas de mensagens, que tem como base o XML (*eXtentisible Markup Language*). Desta forma, os *Web Services* garantem a interoperabilidade e a intercomunicação entre sistemas diferentes devido a utilização de uma linguagem simples (XML), onde os dados são encapsulados e transportados sobre protocolos como HTTP (*HyperText Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*) ou FTP (*File Transfer Protocol*).

A arquitetura básica de um *Web Services*, de acordo com [Champion 2002], é composta por um cliente (*Service Requestor*), que solicita o serviço Web para um servidor (*Service Provider*), podendo fazer uso ainda de um catálogo de serviços disponibilizados na *Internet* (*Discovery Agencies*), num contexto denominado arquitetura orientada a serviços (SOA - *Service Oriented Architecture*).

Por outro lado, dada a dimensão dos campos de abrangência da gerência de segurança, atualmente esta gestão constitui uma nova área da ciência da computação, que compreende tópicos como criptografia, mecanismos de controle de acesso, detecção de intrusão, proteção de perímetros, programação segura, infra-estrutura de chaves públicas, análise de riscos, planos de continuidade de negócio, auditoria e políticas de segurança.

A segurança, de acordo com a norma [NBR 2001] e [Stallings 1999], é dividida em 3 faces: segurança física, segurança operacional e segurança lógica. A segurança física tem como objetivo a promoção de um ambiente de trabalho seguro e confortável, tendo como base o controle de condições ambientais (iluminação, ergonomia, temperatura, ruído ambiental), meio de acesso (perímetro, entradas, áreas críticas) e imprevistos (incêndios, inundações, falta de energia, sinistros). A segurança operacional destaca a normalização dos meios para instauração da segurança, através de planejamento (análise de riscos e políticas de segurança – através de *Network Attached Storage* ou *Storage Area Network*), execução (normas / procedimentos – como a NBR 17999 - e plano de contingência) e acompanhamento por meio de auditoria. A segurança operacional deve prover garantias em três aspectos, conforme [Stallings 1999]:

• **Ataques à segurança**: ações que comprometem a disponibilidade, confidencialidade, integridade e autenticidade podendo ser ativos (com a modificação de mensagens, *replay* de mensagens, mascaramento das informações ou DoS – *Denial of Service*) ou passivos (por meio de *release* de índices das mensagens ou análise de tráfego de informações);

• **Serviços de segurança**: atividades que garantam a segurança em sistemas de informação e na transferência de dados nas organizações, objetivando conter os ataques de segurança. Potencialmente, temos uma classificação dos serviços de segurança particionada nas seguintes áreas: confidencialidade, autenticidade, integridade, não-repúdio, controle de acesso e disponibilidade;

• **Mecanismos de segurança**: meios desenhados para detectar, prevenir ou recuperar informações em caso de ataques de segurança. Eles podem ser divididos em aplicações de defesa de vias de acesso (VPN´s – *Virtual Private Network*), gerência de controle de acesso (ACM - *Access Control Management* – através de *Network Adress Translation* ou Criptografia) e sistemas de detecção de intrusão (IDS - *Intrusion Detection System*).

A complexidade de gerenciamento de segurança não permite o uso de um único sistema para controle de todas as entidades envolvidas neste processo. Sendo assim, aplicações de gerência distribuída baseadas na *Web* têm sido disponibilizadas em equipamentos e servidores para facilitar as tarefas de gerenciamento e configurações. Para serviços *Internet*, citamos aplicações como o Webmin (gerenciador integrado de serviços de rede) e BiFrost (interface para configuração do Netfilter IPTables). Estas ferramentas, entretanto, não costumam fornecer segurança no envio e recepção de informações, como também não permitem a configuração dos parâmetros em múltiplos objetos gerenciados, tal como a inserção de regras em vários sistemas de *firewall*. Contudo, este artigo pretende expor uma abordagem desenvolvida para configuração de regras de forma segura através de *Web Services* em sistemas de filtro de pacotes com o Netfilter IPTables.

3. Metodologia

Como ponto de partida, foi especificado um modelo de gerenciamento aplicado em um cenário de testes, contendo a topologia utilizada e disposição dos elementos da rede, como pode ser observado no tópico 4 deste trabalho.

A seguir, foram definidos as ferramentas para a composição de um protótipo visando à validação do modelo. Nesta instância, fez-se uso de várias tecnologias e ferramentas. As máquinas virtuais, através do *software* VMware Workstation 5.5.1, permitiram a implementação de 3 máquinas virtuais, sendo duas contendo o sistema operacional Linux Slackware 10.2 e a última contendo o Microsoft Windows 98, apenas para testes de compatibilidade entre os sistemas heterogêneos. Ainda, foram utilizados os seguintes *softwares* e ferramentas:

• Apache 1.3.33, PHP 4.4.0 e MySQL 4.1.14: usados na hospedagem da página cliente, responsável pelo envio dos dados digitado pelo usuário ao *Web Services* em gSOAP;

• Biblioteca NuSOAP 1.94: utilizada neste contexto enviar das mensagens SOAP à aplicação C/C++/gSOAP;

• Biblioteca gSOAP: mantém, através de suas funções, um servidor *Web* reduzido que recebe os dados enviados pelo cliente NuSOAP;

• Netfilter IPTables 1.3.3: constitui neste projeto um sistema de *firewall* do tipo filtro de pacotes, com suas regras manipuladas por meio das chamadas do *Web Services*;

• Linguagem C/C++ e compilador gcc 3.3.6: utilizado juntamente com a biblioteca gSOAP para concepção do *Web Services*.

Por fim, foram realizados testes de programação utilizando as bibliotecas NuSOAP e gSOAP, instalação das aplicações para simulação de clientes/servidores nas máquinas virtuais e captura/análise dos pacotes gerados na interação entre os elementos.

4. Modelagem do Ambiente e Cenário de Aplicação

A modelagem do ambiente de especificação é baseada numa corporação que possui uma rede privada protegida por um sistema de *firewall*, conforme definido na Figura 1, que separa a mesma da *Internet*. O método de filtro de pacotes é empregado para controle dos fluxos trocados entre as duas redes.

A configuração deste *firewall* é feita através de uma página *Web*, como base para uma arquitetura distribuída, onde objeto gerenciado e gerente podem estar operando em servidores e locais diferentes. Os dados inseridos na página são enviados a um servidor *Web Services*, que também pode estar separado dos demais serviços, aplicados dinamicamente e de forma segura.

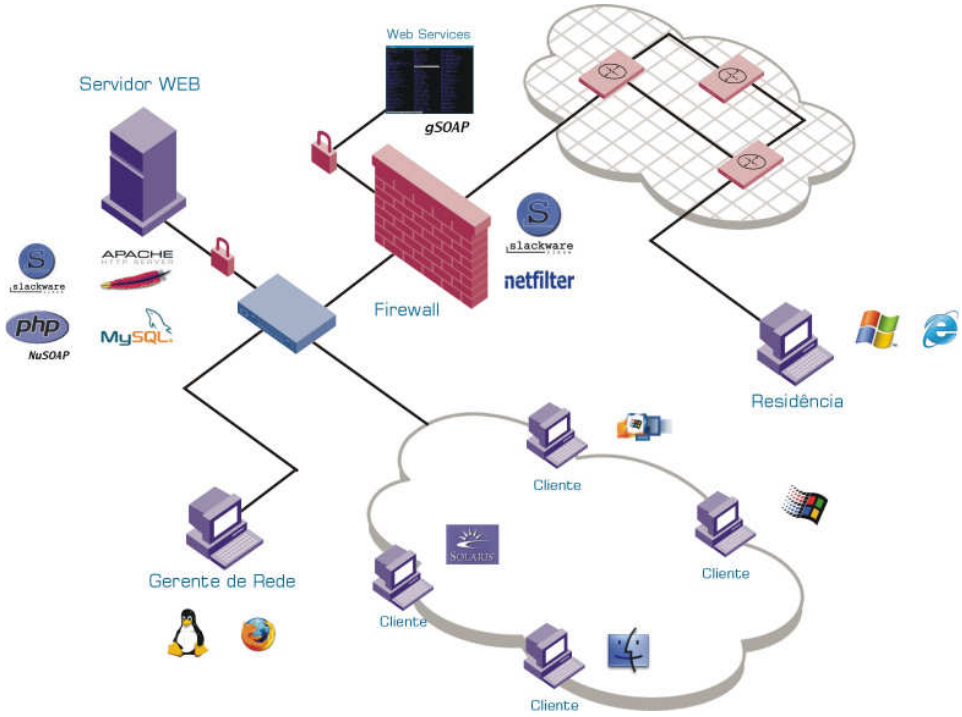

Figura 1. Modelo de gerenciamento e cenário de testes

Esta estrutura permite, por exemplo, que qualquer cliente de posse de um navegador (*browser*) realize alterações na configuração do sistema de *firewall* de forma segura, no escopo da *intranet* e mesmo da *Internet*. A segurança é garantida por meio de um sistema de senhas (armazenado em banco de dados) e o uso do protocolo HTTPS.

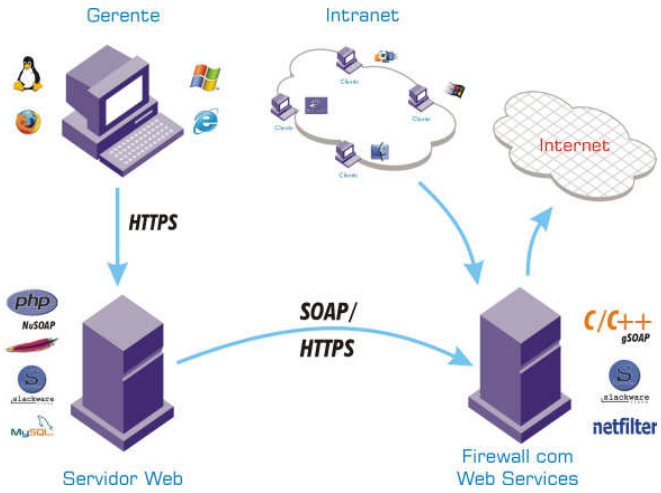

Figura 2: Esquema de funcionamento do sistema e interação dos elementos

Após a concepção da arquitetura utilizada, foi definido o método *screened host dual homed bastion host*, que conforme [Stallings 1999] e [Zwicky 2000] consiste de

um computador *firewall* com pelo menos duas interfaces de rede, o que pode ser observado na Figura 2, assim como a interação entre os elementos e protocolos utilizados. Situado entre a rede interna e a *Internet*, este *host* possibilita a triagem dos pacotes e conexões, realizada pelo NetFilter.

5. Funcionamento do(s) protótipo(s) e Testes Realizados

Para prover troca de mensagens de forma segura entre NuSOAP e gSOAP, foram criadas algumas aplicações visando testar as tecnologias envolvidas, bem como estudar a estrutura de composição dos pacotes de dados de ambas as aplicações *Web Services*. Nesta tarefa, foi imprescindível a utilização do analisador de protocolos de rede Ethereal. Em cada conjunto de programas cliente/servidor criados, uma captura de *frames* era efetuada com o intuito de comparar a composição de dados.

Dentre as aplicações testadas, destacam-se: aplicação cliente/servidor NuSOAP, aplicação cliente/servidor gSOAP, aplicação cliente NuSOAP / servidor gSOAP. Esta última, por razões da implementação de desenvolvedores e métodos de execução distintos, dispensou o maior tempo em sua concepção. A comunicação entre as duas bibliotecas *Web Services* não foi realizada imediatamente com seus códigos originais, devido à incompatibilidade das mensagens produzidas por NuSOAP e gSOAP. Sendo assim, NuSOAP teve que ser alterada em vários pontos de seu código fonte, visando produzir o envelope SOAP compatível com gSOAP. Nesta atividade, cada alteração de NuSOAP era precedida de uma sondagem de pacotes, que era analisada e seguida de novas alterações de código. A tarefa perdurou até que as bibliotecas obtivessem uma linguagem comum. Vale ressaltar que NuSOAP foi escolhida para integrar o projeto por sua simplicidade, leveza e código PHP, que permitiu sua alteração e uso do interpretador para envio das regras e gSOAP, por seu baixo consumo de recursos computacionais e funções C++ que facilitaram a inserção dos parâmetros recebidos de NuSOAP no sistema operacional corrente e, mais precisamente, no Netfilter IPTables. O fluxo para execução da arquitetura, bem como endereços IP configurados nas interfaces de rede das entidades envolvidas, podem ser analisados na Figura 3.

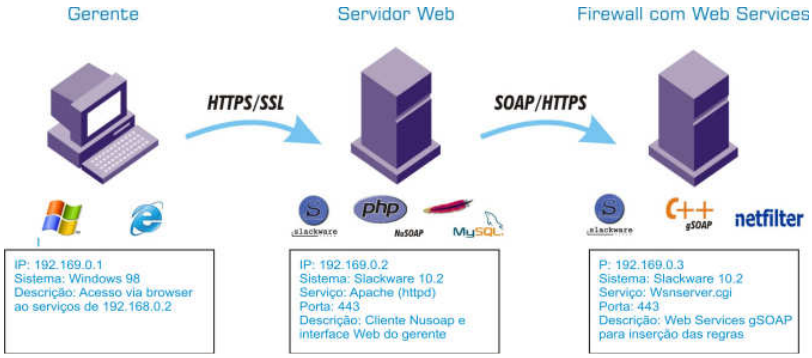

Figura 3. Fluxo de execução da inserção da regra IPTables via interface Web

Em termos gerais, a aplicação cliente NuSOAP construída em PHP envia uma string contendo uma regra IPTables ao software provedor desenvolvido em C++ e gSOAP, que atende requisições na porta 8080. No lado servidor, a função system() do C++ executa o comando recebido do cliente. No entanto, ainda não haviam sido inseridos quaisquer mecanismos para garantia de segurança na troca das mensagens entre as entidades. Para isso, foi realizado um último conjunto de testes provendo a

comunicação segura entre as aplicações produzidas com PHP / NuSOAP e C++ / gSOAP. Neste sentido, utilizou-se as funcionalidade de openssl, um *toolkit* de código livre para implementação do protocolo SSL, nas formas das bibliotecas libssl (que contém os mecanismos SSL) e libcrypto (que constitui dos algoritmos de segurança).

6. Testes Conclusivos e Análise dos Resultados

Após concluídas as etapas anteriores foi produzida uma página *Web* com login e senha para digitação dos comandos, conforme ilustra a Figura 4.

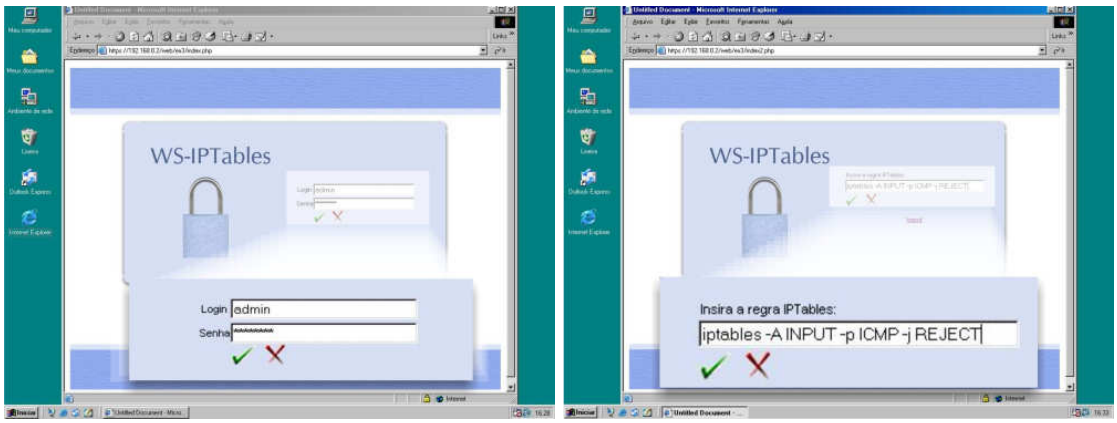

Figura 4. Interface de autenticação e aplicação das regras via navegador

Para finalizar, foi efetuada uma última captura de *frames*, para análise da estrutura dos pacotes sondados. O resultado da interação das 3 máquinas virtuais, desde a entrada no sistema, autenticação e a aplicação da regra estão destacadas na Figura 5, podendo ser analisada através da seqüência de conexões e troca de mensagens. Em situações normais, os dados poderiam ser facilmente identificados em pacotes HTTP/XML/SOAP, entretanto observa-se apenas a presença de *frames* TLS (*Transport Layer Security*), ilegíveis a um suposto atacante.

31	12.527089	192.168.0.2	192.168.0.1	TCP	32768 > https [SYN] Seq=0 Len=0 MSS=1460 TSV=199308 TSER=0 WS=0
32	12.530191	192.168.0.1	192.168.0.2	TCP	https > 32768 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=955711 TSER=199308
33	12.530595	192.168.0.2	192.168.0.1	TCP	32768 > https [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=199309 TSER=955711
34	12.577544	192.168.0.2	192.168.0.1	SSLv2	Client Hello
35	12.578697	192.168.0.1	192.168.0.2	TCP	https > 32768 [ACK] Seq=1 Ack=127 Win=5792 Len=0 TSV=955716 TSER=199313
36	12.585935	192.168.0.1	192.168.0.2	TLS	Server Hello,
37	12.586044	192.168.0.2	192.168.0.1	TCP	32768 > https [ACK] Seq=127 Ack=1449 Win=8688 Len=0 TSV=199314 TSER=955717
38	12.586277	192.168.0.1	192.168.0.2	TLS	Certificate, Server Key Exchange, Server Hello Done
39	12.586417	192.168.0.2	192.168.0.1	TCP	32768 > https [ACK] Seq=127 Ack=1775 Win=11584 Len=0 TSV=199314 TSER=955717
40	12.587684	192.168.0.2	192.168.0.1	TLS	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
41	12.594543	192.168.0.1	192.168.0.2	TLS	Change Cipher Spec, Encrypted Handshake Message
42	12.595033	192.168.0.2	192.168.0.1	TLS	Application Data
43	12.632945	192.168.0.1	192.168.0.2	TCP	https > 32768 [ACK] Seq=1834 Ack=922 Win=7932 Len=0 TSV=955722 TSER=199315
44	12.686310	192.168.0.1	192.168.0.2	TLS	Application Data
45	12.686480	192.168.0.1	192.168.0.2	TLS	Encrypted Alert
46	12.686747	192.168.0.2	192.168.0.1	TLS	Encrypted Alert
47	12.687441	192.168.0.1	192.168.0.2	TCP	https > 32768 [FIN, ACK] Seq=2436 Ack=922 Win=7932 Len=0 TSV=955727 TSER=199315
48	12.687777	192.168.0.2	192.168.0.1	TCP	32768 > https [RST, ACK] Seq=959 Ack=2437 Win=14480 Len=0 TSV=199324 TSER=955727
49	12.687814	192.168.0.1	192.168.0.2	TCP	https > 32768 [RST] Seq=2436 Len=0
50	12.713365	192.168.0.2	192.168.0.3	SSLv3	Application Data
51	12.733525	192.168.0.2	192.168.0.3	TCP	https > 1027 [FIN, ACK] Seq=378 Ack=524 Win=6432 Len=0
52	12.734177	192.168.0.3	192.168.0.2	TCP	1027 > https [ACK] Seq=524 Ack=379 Win=8383 Len=0

Figura 5. Captura de pacotes efetuada durante o acesso ao Web Services

7. Considerações Finais e trabalhos futuros

Administradores e gerentes de redes, em anos recentes, têm feito uso contínuo de estruturas e ferramentas na prática da gestão de redes e, neste sentido, várias aplicações para gerência distribuída via *Web* passaram a fazer parte do dia-a-dia dos gestores das infra-estruturas, por sua praticidade e robustez. Entretanto, atender de forma satisfatória as atividades de gerenciamento e garantias de segurança é tarefa árdua e complexa.

Agentes maliciosos, de posse de ferramentas não-intrusivas para levantamento de informações através da captura e análise do tráfego da rede podem obter facilmente dados da estrutura gerenciada.

A solução desenvolvida tem como ponto forte a heterogeneidade disponível através dos *Web Services* e a inserção de forma segura das regras no *firewall*, proporcionado pelo uso de SSL na troca de informações entre os pares (cliente *Web* / servidor PHP-NuSOAP e servidor PHP-NuSOAP / *firewall*-Netfilter-gSOAP). Destaca-se, ainda, a complexidade da tarefa de alterações de código em NuSOAP, que propiciou a troca de dados entre bibliotecas distintas e tecnologias de desenvolvimento diferentes.

Dentre os possíveis trabalhos futuros, destacam-se a melhoria do controle de acesso ao *Web Services* (com uso de protocolos AAA - *Authentication, Authorization* e *Accounting* ou PKI - *Public Key Infrastructure*), aperfeiçoamento da interface *Web* (semelhante ao Firewall Builder, de forma que o administrador não necessite conhecer a sintaxe do Netfilter IPTables), adequação do sistema para inserção de regras em múltiplos servidores, implementação de técnicas de programação segura, controle das chamadas ao sistema via comandos em C++ ou PHP, além de testes com o SE-Linux, um componente de tomadas de decisão de acesso ao Kernel do Linux e privilégios de sistema, desenvolvido sob os moldes da arquitetura MAC (*Mandatory Access Control*).

Referências

CHAMPION, M., et. al. (2002) “Web Services Architecture”. http://www.w3.org/TR/2002/WD-ws-arch-20021114/

COULOURIS, G., et. al. (2005) “Distributed Systems - Concepts and Design”. 4nd ed. Addison-Wesley.

DANTAS, M. (2002) “Tecnologias de Redes de Comunicação e Computadores”. Rio de Janeiro: Axcell Books.

KUROSE, J. F.; ROSS, K. W. (2003) Redes de Computadores e a Internet: uma abordagem top-down. 3. ed. São Paulo: Pearson Addison Wesley.

LIMA, W. Q. de. (2005) “Estado da arte do Gerenciamento de Redes de Computadores através de Web Services”. Porto Alegre: UFRGS, 2005. Trabalho Individual I (Doutorado), Instituto de Informática, Universidade Federal do Rio Grande do Sul.

LOPES, R. V., et. al. (2003) “Melhores Práticas para Gerência de Redes de Computadores”. Rio de Janeiro: Campus.

NBR ISO/IEC 17799 (2001) “Tecnologia da informação - Código de prática para a gestão da segurança da Informação”. Rio de Janeiro: Associação Brasileira de Normas e Técnicas.

STALLINGS, W. (1999) “Cryptography and network security: principles and practice” 2ª ed. Upper Saddle River: Prentice Hall.

VIANA, R. L., et. al. (2006) “Comparando Aspectos de Desempenho do Protocolo SNMP com Diferentes Estratégias de Gateways Web Services”. Anais do 24º Simpósio Brasileiro de Redes de Computadores (SBRC 2006). Curitiba.

ZWICKY, E. D., et. al. (2000) “Building Internet Firewalls”. 2nd ed. O'Reilly & Associates. Inc.