Um Ambiente de Segurança Distribuído para a Integração de Firewalls com Sistemas de Detecção de Intrusão

José M. Bonifácio Jr[1], Edson S. Moreira[1], Adriano M. Cansian[2] e André C. P. L. F. de Carvalho[1].

[1]{*boni, edson, andre*}*@icmsc.sc.usp.br*
Instituto de Ciências Matemáticas de São Carlos - ICMSC/USP
Po Box 668 - CEP 13560-970 - São Carlos - SP - Brasil
[2]*adriano@ibilce.unesp.br*
UNESP - Universidade Estadual Paulista - IBILCE / S.J.Rio Preto
Po Box 136 - CEP 15970-001 - São José do Rio Preto - SP - Brasil

Resumo

Com o atual crescimento no uso de computadores e redes de computadores, aliado ao seu uso para atividades essenciais e comerciais, e a interligação de redes corporativas à Internet, segurança de computadores se torna um assunto chave nas mais diversas áreas. Paralelamente a este crescimento, é também observado um aumento nas tentativas de intrusão e ataques à redes de computadores. Este trabalho mostra um sistema de detecção de intrusão baseado em redes neurais e os resultados obtidos de um protótipo implementado. Este sistema de detecção será integrado à um firewall dentro de um ambiente de segurança distribuído. Esta integração visa complementar o firewall com as capacidades do sistema de detecção de inferir sobre o conteúdo das conexões permitidas pelo firewall.

Abstract

With the current growing in the use of computers and networks, their use for essential and commercial activities, and the connection of the corporate networks with the Internet, computer security has become a key issue in several different areas. It is observed too an increasing rate in attack attempts to the computer networks. This work show a intrusion detection system based in neural networks, as well as the results obtained from an implemented prototype. This intrusion detection system will be integrated with a firewall in a distributed security environment. This integration intends for adding to the firewall the capabilities that the intrusion detection system has to make inferences about the content of the connections allowed by the firewall.

Keywords: Intrusion Detection System, Neural Network, Distributed System

Introdução

Uma das principais características da sociedade moderna é o reconhecimento de que o poder é derivado da posse de informações. Como resultado, a posse, o manuseio e a proteção da informação tem se tornado um aspecto de crucial importância para a sociedade em geral. Observa-se uma crescente expansão do uso de computadores e sua utilização em redes. O armazenamento de informações sensíveis e sua transferência pela rede tornaram-se características primordiais em sistemas de computação modernos.

Por outro lado, atos de pirataria, tentativas de ataque e intrusões consumadas tem se tornado frequentes e envolvem um número crescente de computadores, [1][2] destinando-se geralmente, ao roubo, destruição ou alteração das informações. Frequentemente ocorrem situações onde os intrusos facilmente superam os mecanismos de proteção existentes [3]. Além disso, uma quantidade cada vez maior de atividades essenciais é realizada por intermédio das redes (principalmente através da Internet), e seu funcionamento correto e confiável torna-se de vital importância. Este cenário mostra a necessidade de técnicas especiais de segurança nos sistemas de computação modernos; técnicas que vão além da tradicional prática "*locking-up-the-doors*".

Cenário Atual

O estudo de segurança em redes de computadores é uma área de crescente interesse, uma vez que a rede é o meio pelo qual a maioria dos ataques ou intrusões em sistemas de computadores são realizados. Existem diversas tecnologias para se atacar este problema, dentre elas, firewalls, filtros, sistemas de monitoração e sistemas de detecção de intrusão. O uso destas tecnologias parte do princípio de que abandonar toda a gigantesca infra-estrutura de redes e computadores inseguros, ou torná-los seguros, é impossível ou financeiramente inviável.

A tecnologia de segurança de redes mais largamente utilizada é o firewall [17], porém ainda não existe nenhum sistema que garanta o estado-da-arte em matéria de proteção e que forneça um elevado grau de segurança enquanto permite uma certa flexibilidade e liberdade no uso dos recursos computacionais protegidos. Estes sistemas previnem o uso não autorizado de um recurso computacional usando algum tipo de mecanismo de controle de acesso, mas ainda existem diversos fatores que impeçam que atacantes eventualmente tenham acesso ao sistema. A maioria dos sistemas computacionais apresenta algum tipo de furo de segurança, permitindo que atacantes externos (ou mesmo usuários legítimos) tenham acesso à informações confidenciais. Mesmo um sistema supostamente seguro pode ser vulnerável à usuários internos abusando de seus privilégios ou se tornar comprometido através de práticas impróprias.

Uma vez que um ataque pode ser considerado inevitável, existe uma óbvia necessidade por mecanismos que possam detectar atacantes tentado entrar no sistema ou usuários legítimos fazendo mal uso de seus privilégios.

Um exemplo de deficiência de firewalls ocorre quando um usuário legítimo, ou um atacante que tenha conseguido acesso, executa operações que explorem furos de segurança na rede interna ao firewall. Como estas conexões serão legítimas para o firewall, ele não terá meios de identificar o ataque. Neste caso é necessário o uso conjunto de um sistema de detecção de intrusão, que tenha acesso aos dados da conexão e que possa inferir sobre seu comportamento, distinguindo conexões normais de conexões que representem uma possível situação de ataque.

Sistemas de detecção de intrusão [18] podem ser divididos em dois tipos básicos: os baseados em host [4-8] e os baseados em redes [9-11]. Os sistemas baseados em host usam *audit trails* (geralmente *deamons* rodando nas máquinas) para detectar um comportamento suspeito baseado nas ações que ocorrem na máquina, enquanto que os baseados em rede constróem seu próprio conjunto de informações usando o tráfego da rede capturado diretamente do meio de comunicação.

Um sistema de detecção de intrusão baseado em rede foi desenvolvido no ICMSC-USP. Tal sistema [12-15] faz uso de redes neurais artificiais [20] para identificar um comportamento intrusivo nas conexões estabelecidas. O uso de redes neurais é uma proposta inovadora na medida que permite um alto grau de adaptabilidade ao sistema. Novas técnicas de intrusão

podem ser facilmente adicionadas ao sistema retreinando-se a rede neural com os novos padrões de ataque. Outra característica deste sistema é a capacidade de generalização da rede neural que pode identificar um padrão intrusivo mesmo que nunca tenha tido contato com este tipo de ataque.

A proposta deste trabalho é de que o uso conjunto de um sistema de proteção, como um firewall, com um sistema de detecção de intrusão, pode prover um alto nível de segurança a um sistema computacional, uma vez que os dois sistemas possuem características complementares. Tal integração se dará através de um sistema de gerenciamento de acesso distribuído. Este sistema terá, além das características de processamento distribuído, modularidade, escalabilidade, tolerância a falhas, balanceamento de carga, gerenciamento remoto, comunicação segura entre os diversos módulos e portabilidade, dentre outras.

Será apresentado a seguir um overview sobre firewalls com enfoque nos seus pontos fortes fracos, um overview do sistema de detecção de intrusão, um modelo do sistema de gerenciamento de acesso, seus requisitos e características e por fim uma análise dos benefícios da integração entre firewalls e sistemas de detecção de intrusão.

Firewalls

O propósito de um firewall é prevenir o acesso de usuários não autorizados à recursos computacionais em uma rede privada e, possivelmente, impedir exportação não notificada e não autorizada de informações confidenciais.

Na configuração de um firewall, as principais decisões relacionadas à segurança são frequentemente ditadas pela política da organização ou corporação. Especificamente, as decisões devem ser tomadas fazendo-se um compromisso entre o nível de segurança desejado e a flexibilidade e facilidade de uso dos recursos computacionais que o firewall se destina a proteger, levando-se em conta também os investimentos disponíveis.

Existem duas abordagens básicas na configuração de um firewall:

- *O que não é expressamente proibido é permitido*
- *O que não é expressamente permitido é proibido*

A escolha de uma abordagem define toda a política de configuração e uso do firewall e de toda a rede e usuários por ele protegida. A primeira abordagem implica que o administrador do sistema deve prever que tipos de ações os usuários, ou pessoas externas, possam fazer que infrinjam a política de segurança. No segundo caso, o firewall é projetado para bloquear tudo e os serviços devem ser permitidos caso a caso após um cuidadoso estudo de necessidades e riscos. Esta política geralmente afeta diretamente os usuários, que passam a tratar o firewall como um estorvo. É uma abordagem mais segura e conservadora com base no fato de que o que não é conhecido pode ser perigoso.

A adoção de uma ou outra abordagem envolve o tipo de organização, as necessidades dos usuários, o grau de segurança desejado, o propósito desejado e diversos outros fatores. A segunda abordagem pode ser bem aceita em empresas com normas rígidas e bem definidas, mas pode-se tornar altamente imprópria em instituições de ensino e pesquisa, uma vez que ela restringe em demasia o uso de recursos computacionais. A não ser que se tenha uma equipe muito bem preparada e que reconheça a fundo todas as diferentes necessidades dos usuários do sistema e reflitam isto na configuração, o firewall pode ser tornar um inconveniente e mesmo prejudicar o andamento de pesquisas.

Um firewall pode apresentar diversos problemas. Uma pequena brecha na configuração ou algum serviço que tenha sido esquecido ou que tenha sua importância negligenciada é suficiente para que se tenha um furo na segurança que intrusos poderão explorar. Este perigo ainda pode ser potencializado, uma vez que o administrador, ao usar um firewall, dará uma importância relativamente menor às configurações individuais das máquinas.

Outro grande problema com firewalls diz respeito à maus usuários (que tenham usernames e passwords válidos). Qualquer um que queira abrir uma conexão "teoricamente" legal terá permissão, a não ser em casos que todas as conexões da máquina, ou do domínio de origem tenham sido configuradas para serem filtradas pelo firewall. Um intruso, que consiga um username e um password, poderá abrir uma conexão de uma máquina externa e ter acesso à todo o sistema. A obtenção da senha pode-se dar por furos existentes, por engenharia social (quando um intruso engana alguém para conseguir uma senha, por exemplo), por descuido de um usuário legítimo que tenha escrito sua senha em algum lugar impróprio ou qualquer outro método. Uma vez dentro do sistema, ele poderá usar diversas técnicas disponíveis e bem conhecidas para então executar suas intenções.

Ainda que firewalls sejam uma alternativa considerada segura e largamente utilizada, seu uso não implica em segurança total. É necessário o uso de outras tecnologias atuando em conjunto para se aumentar o nível de segurança de uma rede privada.

Sistema de Detecção de Intrusão

O objetivo de um sistema de detecção de intrusão é prover capacidades extras a um sistema de segurança, como um firewall. Tal sistema deve observar as conexões consideradas seguras e permitidas pelo firewall procurando por um comportamento suspeito que indique uma possível intrusão ou tentativa de ataque. Como o firewall não tem meios de avaliar o conteúdo de tais conexões, ele se baseia apenas em suas regras de filtragem para decidir se uma conexão pode ou não ser aceita. Desta forma, as funcionalidades de um sistema de detecção se tornam de vital importância na medida em que provêem meios de inferir sobre o conteúdo das conexões permitidas e detectar as que apresentem um comportamento suspeito ou não condizente com a política de segurança implantada.

Tentativas de ataque acontecem de acordo com algumas técnicas de acesso e frequentemente o invasor está fisicamente fora do sistema sob ataque [2]. Os primeiros modelos de Sistemas de Detecção de Intrusão (SDI), projetados para computadores isolados, usam algoritmos básicos que incluem análise de funções multinomiais e aproximação de matrizes covariantes para detectar desvio do comportamento normal [4-5], tão bem como sistemas especialistas para detectar violação de políticas de segurança [8]. Os modelos mais modernos monitoram um grande número de redes de computadores e transferem a informação monitorada para ser processada em um equipamento central que emprega técnicas de sistemas distribuídos.

A maioria dos SDIs tem um processo auditor (*daemon*) em cada máquina, responsável por capturar ações de violação de segurança dentro da máquina. Sistemas baseados em redes, ao invés de utilizar pistas de auditoria, analisam o tráfego de pacotes dentro da rede[9] para detectar comportamento intrusivo [6].

O Sistema de Detecção de Intrusão do ICMSC

Uma das características inovadores deste sistema de detecção consiste em introduzir um agente de segurança capaz de detectar comportamento intrusivo em conexões estabelecidas. Este agente atua capturando e decifrando pacotes que são transmitidos através da rede sobre

monitoramento. Para fazer uma inferência sobre a condição de segurança das conexões, o agente emprega um sistema especialista e uma rede neural que irá prover um coeficiente de suspeita, o qual, baseado em informações intrusivas previamente registradas, dará uma idéia a respeito da severidade do ataque ou o grau de suspeita das atividades naquela conexão.

O sistema se baseia no fato de que uma intrusão pode ser detectada a partir de uma análise de modelos predeterminados, que são anômalos comparados com ações normais [6-7]. A grande maioria dos ataques são resultado de um pequeno número de ataques conhecidos, como relatados por equipes como o CERT [19].

O uso de redes neurais pode fornecer mecanismos para o reconhecimento de ataques, tão bem quanto uma capacidade de adaptação em resposta a mudanças nas técnicas de intrusão.

O agente é organizado em quatro módulos (Figura 1). Os módulos gerenciam o fluxo de pacotes e fornecem um vetor de estímulo para a rede neural. O nível mais baixo apenas captura um fluxo de dados na rede e passa os pacotes ordenados ao segundo módulo. O módulo seguinte consiste de 2 sub-módulos: módulo de pré-seleção de pacotes, e módulo do sistema especialista. O módulo de pré-seleção faz a monitoração e filtragem iniciais dos pacotes, que possam representar eventos de interesse, tais como que tipo de protocolo será monitorado ou que origens e destinos devem ser considerados. Os pacotes previamente filtrados então passam através de uma análise feita pelo sistema especialista. O sistema especialista usa as seguintes informações ao tomar decisões de quais conexões terão seus conteúdos monitorados: Quais os caminhos esperados de conexões: origem e destino (quais podem ser perigosos) e portas de origem e destino envolvidas; sensibilidade das máquinas e confiabilidade dos domínios; capacidades dos serviços e autenticação do nível de segurança. O sistema especialista ainda mantém registro de eventos passados que influenciarão em sua decisão. Eventos previamente registrados de um mesmo par origem-destino irão conferir um maior grau de severidade a novos eventos.

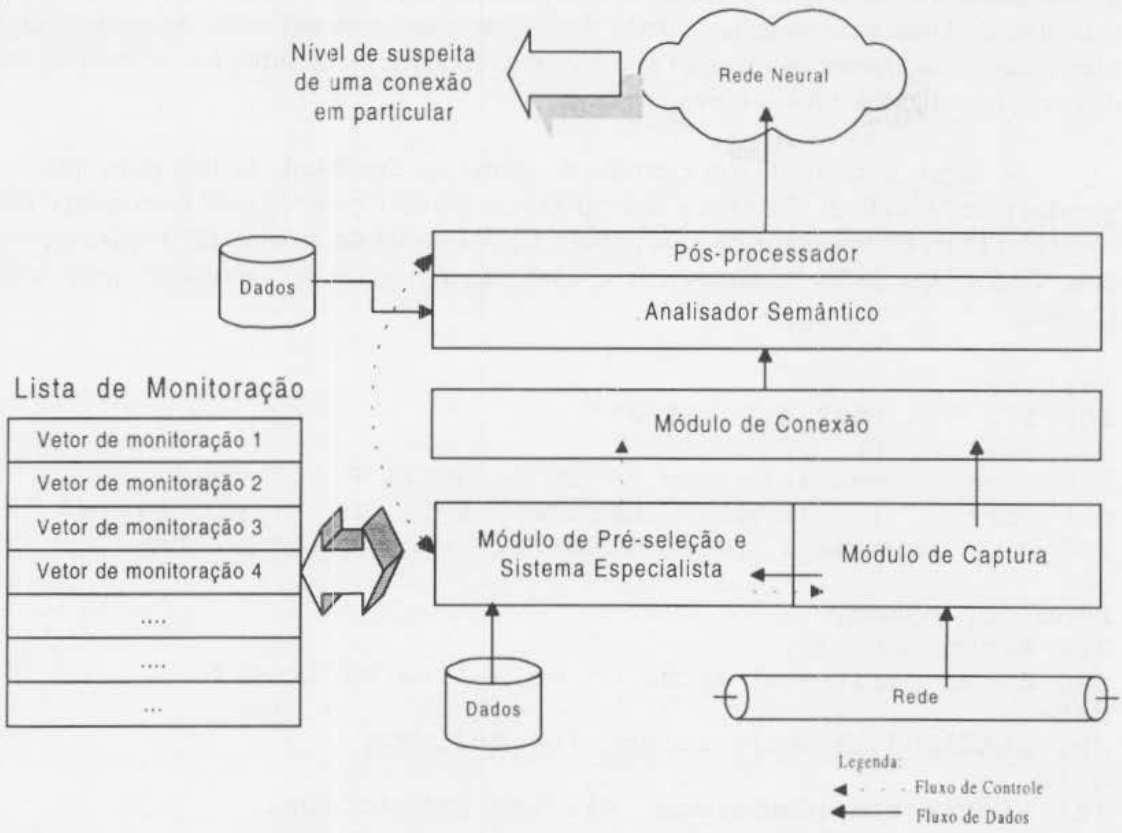

Figura 1 – Estrutura modular do Sistema de Detecção de Intrusão

O terceiro módulo é baseado no modelo hierárquico do Monitor de Segurança de Redes (MSR) [9-11]. Ele recebe os pacotes e organiza-os em uma relação de causa-efeito, identificando um fluxo de dados unidirecional. Isto é feito através da análise dos campos de origem e destino, portas, e número de sequência dos pacotes, solucionando então o problema da fragmentação. Este procedimento pode ser feito no nível de IP ou TCP. Este fluxo de dados representa a transcrição dos dados de uma particular conexão, que são então mandados para serem processados pelo Analisador Semântico.

O Analisador Semântico atua sobre o fluxo de dados da conexão, buscando por perfis de ataque que poderiam aparecer nos dados. Estes perfis são assinaturas de ataque e contém informações de como uma sessão suspeita se comporta. O Analisador Semântico irá procurar por strings suspeitas (*login root, access denied, etc*). Estas informações são enviadas para o último módulo de pós-processamento, que as unifica com as informações do sistema especialista (sensibilidade dos serviços e conexões) e à sua base de dados, para formar o vetor de estímulo para a rede neural. Estas informações são codificadas de forma binária para facilitar o treinamento e uso da rede neural, cada string suspeita tem um código binário diferente.

Na atual fase de implementação do protótipo do sistema de detecção, o vetor de estímulo contém apenas o código binário da porta utilizada na conexão (ou seja, o serviço: ftp, telnet, sendmail, etc) seguido pelos códigos binários das strings suspeitas encontradas na conexão. Futuramente estão previstas a adição de informações como: capacidade e nível de autenticação do serviço, nível de segurança das máquinas envolvidas, quantidade de dados transferida e horário da conexão.

A rede neural analisa o vetor de estímulo, e tenta atribuir um grau de suspeita, que representa o estado de suspeita de uma conexão em particular. Antes que a rede neural possa identificar ataques potenciais, ela deve ser treinada com um significativo e suficientemente grande número de vetores de estímulo, que representem o comportamento de conexões suspeitas e legítimas. Uma vez treinada, a rede deve usar suas características de generalização para identificar corretamente os usuários que mostrem características similares às contidas nas ações de intrusão utilizadas em seu treinamento.

A seguir é mostrado um exemplo de ataque ao SendMail. As três primeiras linhas são geradas pelo Módulo de Conexão e representam as portas e endereços IP de origem e destino. A conexão (TCP) foi originada do host fictício 1.2.3.4 (porta de usuário 1899) para a porta 25 do host 4.3.2.1. Os dados restantes são o conteúdo da conexão, capturados pelo Módulo de Captura.

```
::::::::::::::
TCP:1.2.3.4-1899_4.3.2.1-25
::::::::::::::
220 victim.someplace.com ESMTP Sendmail 8.7.5 ready.
mail from " | /bin/mail intruder@devil.com < /etc/passwd "
250 " | /bin/mail intruder@devil.com < /etc/passwd "... sender
ok.
rcpt to: nobody
250 Recipient ok.
354 Enter mail, end with "." on a line by itself
data .
250 QAA23003 Message accept for delivery
quit
221 victim.someplace.com  closing connection.
```

Este registro é uma assinatura de ataque e guarda o comportamento de uma conexão suspeita. Como há muitas informações desnecessárias, o Analisador Semântico irá filtrar os dados procurando pelos componentes principais de uma assinatura de ataque.

Com base nesta conexão, o Analisador Semântico irá produzir o vetor de estímulo como entrada para a rede neural mostrado na Tabela 1.

Código Binário	Strings Suspeitas
0 0 0 1 0 0	Port 25
0 0 0 1 0 0 1 0 0 1 1 0	mail from " |
1 1 1 0 1 0 0 0 0 1 1 1	| /bin/mail
1 0 0 0 0 1 0 1 1 1 1 0	< /etc/passwd
0 0 1 0 1 0 1 0 1 0 0 1	rcpt to: nobody
0 1 0 0 1 1 1 0 0 1 1 1	< /etc/passwd "... sender ok.
0 0 0 0 0 0 0 0 0 0 0 0	Slot vazio
0 0 0 0 0 0 0 0 0 0 0 0	Slot vazio
0 0 0 0 0 0 0 0 0 0 0 0	Slot vazio
0 0 0 0 0 0 0 0 0 0 0 0	Slot vazio
0 0 0 0 0 0 0 0 0 0 0 0	Slot vazio

Tabela 1 – Vetor de Estímulo produzido pelo Analisador Semântico

O vetor de estímulo é composto por duas partes, como mostrado na Figura 2:

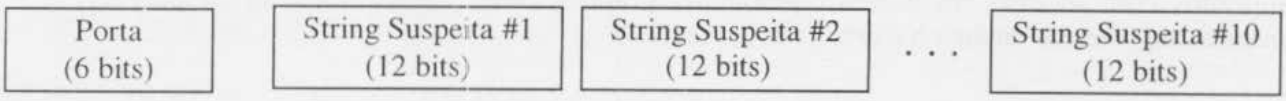

Figura 2 – Vetor de Estímulo

A primeira parte identifica a porta destino da conexão, isto é, o serviço utilizado. Esta parte tem 6 bits onde cada bit identifica uma porta e o código 000000 identifica as demais portas menos importantes.

A segunda parte corresponde às strings suspeitas. O vetor de estímulo pode conter até 10 strings suspeitas. Caso haja menos de 10 strings, os slots restantes são preenchidos com 0. Se forem encontradas mais de 10 strings na conexão, as outras serão ignoradas. Alternativamente, pode-se usar técnicas como janelas, correspondendo aos diferentes subconjuntos de strings encontradas como mostrado na Figura 3.

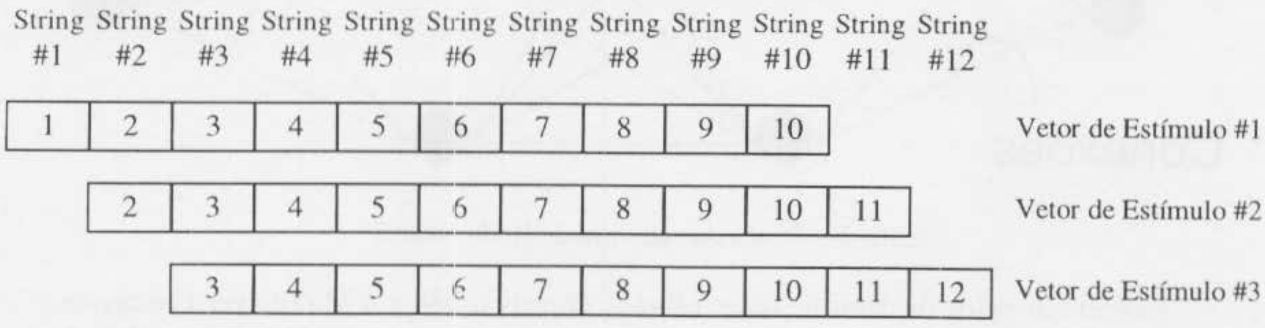

Figura 3 – Método de Janelas com Vetor de Estímulo

Redes Neurais

Um grande e complexo número de tarefas que um humano pode executar com aparente facilidade não são tão facilmente realizadas por computadores utilizando métodos e algoritmos tradicionais. É esperado que estas tarefas devam ser melhor executadas por computadores que tenham estruturas e processamento similares aos encontrados no cérebro humano, que é o caso de Redes Neurais Artificiais, RNAs [20]. RNA são sistemas distribuídos altamente paralelos compostos por simples unidades de processamento parecidas com neurônios, dispostas em uma ou mais camadas. Existe um grande número de conexões com pesos entre os neurônios. Estes pesos codificam o conhecimento de uma RNA e são usados para definir a influência de cada entrada recebida por um neurônio em sua saída. A saída de um neurônio é usualmente o resultado de uma função de ativação aplicada à soma ponderada de suas entradas. A Figura 4 mostra uma típica arquitetura de Rede Neural.

Uma das principais características dos modelos de RNA é sua habilidade de aprender por exemplos, o que significa que elas não necessitam ser explicitamente programadas para realizar uma dada tarefa. A rede deve se treinada com um conjunto de exemplos representativos da tarefa que ela deve reconhecer. Após treinada, a rede deve ser capaz de generalizar o que foi aprendido para situações similares. Esta habilidade permitirá o correto reconhecimento de padrões similares àqueles encontrados no conjunto de treinamento. Sistemas RNA tem sido aplicados com sucesso em diversos problemas práticos como: reconhecimento de padrões, sistemas de controle dinâmico e predições [21-24].

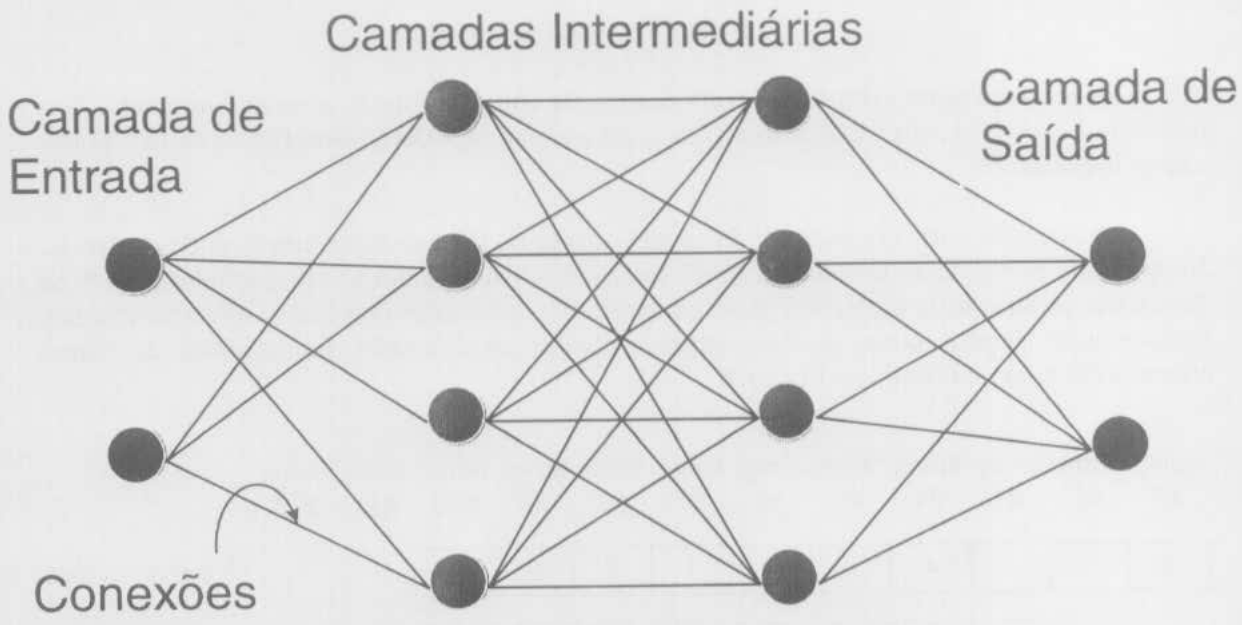

Figura 4 – Típica Arquitetura de Rede Neural

Existem diversos modelos de redes neurais. O mais usado é o MultiLayer Perceptron, MLP [25]. Os experimentos apresentados neste artigo usam redes neurais MLP. Estas redes são geralmente treinadas pelo algoritmo backpropagation. Este algoritmo usa uma generalização da regra delta rule para treinar uma rede com várias camadas começando na camada de saída e voltando camada por camada. Geralmente, este treinamento requer diversos passos. Como o algoritmo backpropagation requer um longo tempo para treinar a rede, algumas modificações

foram utilizadas que aceleram o processo de treinamento. As variações do backpropagation que foram usadas são Quickprop [26] e Rprop [27].

Experimentos e Resultados

As redes neurais usadas nestes experimentos tem 126 neurônios na camada de entrada correspondentes aos 126 bits do vetor de estímulo. Diversas configurações diferentes foram testadas para identificar a que apresenta a melhor performance para este problema.

Os testes foram realizados com o simulador SNNS (*Sttutgart Neural Network Simulator*) [28]. Foram feitos testes com 8 topologias diferentes, todas com 126 neurônios na camada de entrada e um na camada de saída. As topologias usadas foram: 126-1 (126 neurônios de entrada e 1 de saída, sem camadas intermediárias), 126-5-1, 126-10-1, 126-20-1, 126-40-1, 126-60-1 (com uma camada intermediária, onde o segundo número indica o número de neurônios nesta camada), 126-20-1-1 e 126-20-5-1 (com duas camadas intermediárias). Foi usada uma segunda camada intermediária na tentativa de se aumentar a performance. Nos testes, foi verificado que a rede com a melhor performance foi a 126-20-1, então foi decidido inserir a segunda camada intermediária nesta topologia, com 1 e 5 neurônios. Todas as topologias utilizadas são completamente conectadas, isto é, todos os neurônios em uma camada estão conectadas a todos os neurônios da camada seguinte.

Foram usados 3 algoritmos de treinamento, cada um com parâmetros diferentes mostrados na Tabela 2.

Algoritmo de Treinamento	Parâmetros
BackPropagation	n:1 delta:0.5 n:1 delta:0.1 n:0.1 delta:0.1 n:0.2 delta:0.5
RPROP	$delta_0$:0.1 $delta_{max}$:50 a:2 $delta_0$:0.1 $delta_{max}$:50 a:0.1 $delta_0$:0.1 $delta_{max}$:50 a:0.001
QuickProp	n:0.2 u:2.25 v:0.0001 delta:0.1 n:0.2 u:1.75 v:0.0001 delta:0.1

Tabela 2 – Algoritmos de Treinamento e Parâmetros

Cada uma das 8 topologias foram testadas utilizando-se todos os algoritmos com todos os respectivos possíveis parâmetros. Para cada possível configuração (topologia, algoritmo de treinamento e parâmetros) foi verificado que valores diferentes nos pesos iniciais dos neurônios causaram grandes diferenças na saída da rede. Para conseguir um resultado confiável, foram feitos 20 treinamentos diferentes para cada configuração possível e foi usado o valor médio como um valor de comparação adequado. Foram feitos 1440 treinamentos diferentes ao todo e os resultados agrupados em 3 gráficos (Figuras 5, 6 e 7) correspondentes aos três algoritmos: Backpropagation, Rprop e QuickProp. O eixo X representa as topologias usadas, o eixo Y, o erro médio e as barras em cada topologia representam os parâmetros usados indicados nas caixas à direita do gráfico.

Foram usados três conjuntos de dados para realizar os testes: treinamento, validação e teste, com 120, 56 e 56 padrões respectivamente; e todos os conjuntos contendo 50% de padrões

de comportamento intrusivo e 50% de não intrusivo, para garantir um resultado não tendencioso da rede neural. Foram adicionados ruídos em alguns padrões de todos os conjuntos.

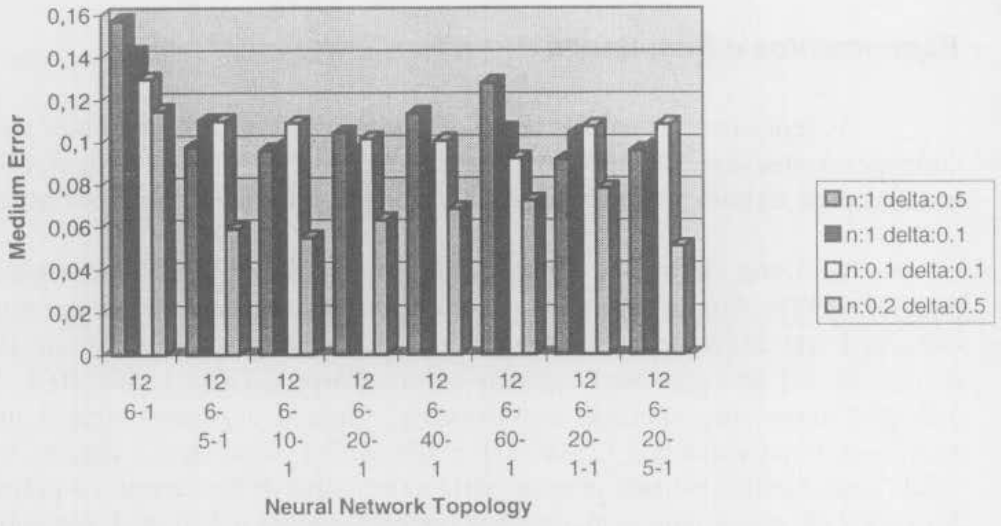

Figura 5 – Resultados do Algoritmo Backpropagation

Para o treinamento, foram usados 500 ciclos de treinamento e uma validação a cada 10 ciclos. Após o treinamento, foi feito o teste com o terceiro padrão (teste).

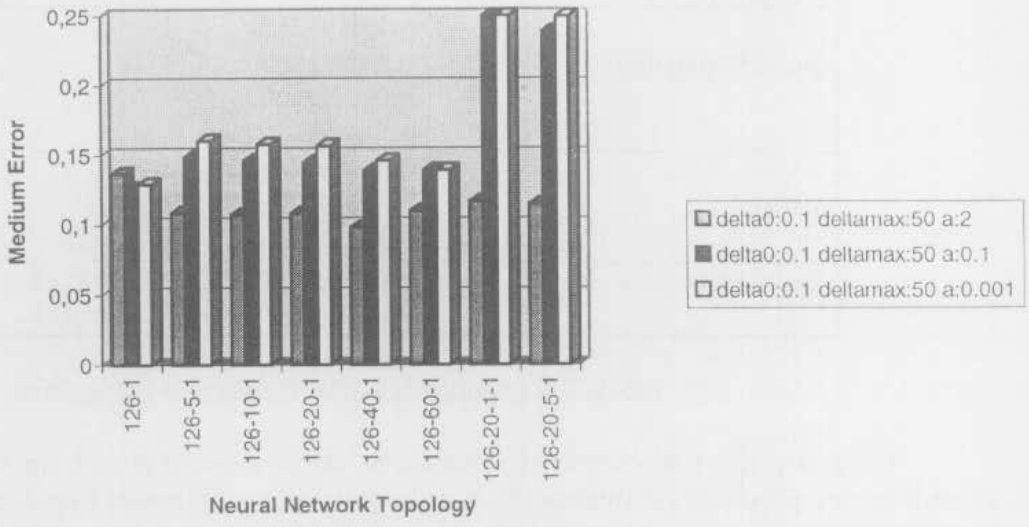

Figura 6 – Resultados do Algoritmo Rprop

Os padrões de ataque e não-ataque foram capturados em um ambiente de rede controlado por simulações de ataque e comportamentos normais. As simulações de ataque foram feitas manualmente usando-se técnicas de intrusão conhecidas e por sistemas como SATAN [29] e ISS [30]. Com estes métodos, foi conseguido um bom conjunto de assinaturas para treinar as redes neurais.

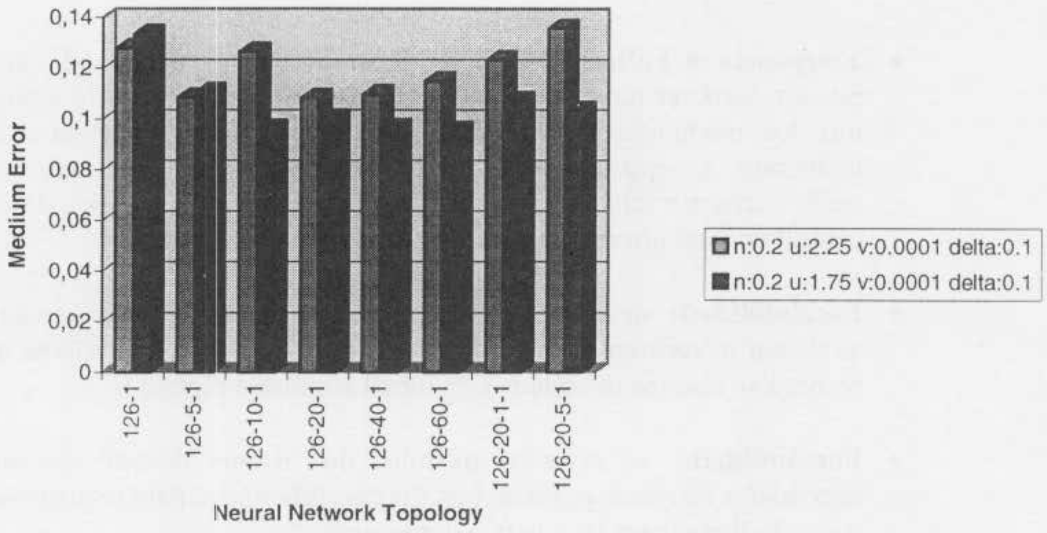

Figura 7 – Resultados do Algoritmos QuickProp

A análise dos resultados mostra que a rede neural com a melhor performance foi a 126-20-5-1, com duas camadas intermediárias, algoritmo de treinamento e parâmetros n = 0.2 e delta = 0.5, apresentando uma taxa média de erro de 5.067%. É interessante notar também que este algoritmo de treinamento com estes parâmetros teve os melhores resultados, e nas topologias com apenas uma camada intermediária, os melhores resultados situaram-se entre as topologias 126-5-1 e 126-20-1, tornando-se pior com mais ou menos neurônios na camada intermediária.

Sistema de Segurança Distribuído

Aqui será apresentado o modelo de um sistema de segurança distribuído. A intenção de se implementar um sistema distribuído é aproveitar toda a gama de recursos computacionais já existentes para a execução distribuída das partes do sistema, além de apresentar diversas outras vantagens e capacidades. Este ambiente será a base para a integração do firewall com o sistema de detecção, que serão duas aplicações subordinadas ao sistema.

O gerenciamento dos objetos do sistema de forma distribuída será realizado via CORBA [33], que provê uma plataforma simples de desenvolvimento para aplicações distribuídas.

O sistema deverá ter os seguintes requisitos básicos:

- **Modular**: deverá ser formado por módulos pequenos que realizem tarefas específicas. Essa estrutura deve permitir que mudanças, inserções e remoções de módulos sejam simples, sem a necessidade de grandes alterações no restante do sistema;

- **Processamento Distribuído**: os módulos podem estar em uma ou várias máquinas de forma transparente. Este recurso visa aproveitar a capacidade computacional ociosa da rede;

- **Gerenciamento Remoto**: o administrador deverá ser capaz de controlar o sistema remotamente via um Web Browser (Netscape Navigator ou Internet Explorer) de forma segura. Com isto, em qualquer parte da rede, ou até mesmo fora dela, o

administrador poderá se comunicar com o sistema sem a necessidade de estar fisicamente presente em um determinado ponto da rede;

- **Tolerância a Falhas**: deverá ter recursos que permitam que diversas máquinas possam fornecer um mesmo serviço de forma que caso ocorra algum problema com um dos módulos, outro módulo idêntico em outra máquina assuma seu lugar, mantendo a operacionalidade do sistema como um todo. Este recurso é particularmente interessante pois dá ao sistema a característica de não ter um ponto central no qual um ataque comprometeria todo o sistema;

- **Escalabilidade**: deve ser possível a inserção de módulos dinamicamente de forma a aumentar os recursos disponíveis no sistema, ou seja, a inclusão de uma aplicação ou recurso ao sistema deve se dar de forma simples e rápida;

- **Portabilidade**: os diversos módulos do sistema devem ser capazes de serem executados na maior gama de plataformas diferentes. Este requisito será suprido com o uso da linguagem Java para a implementação;

- **Comunicações Seguras**: as comunicações entre módulos devem se dar de forma segura através de criptografia e sistemas de autenticação e validação como o protocolo SSL V3.0 [32], desenvolvido pela Netscape. Deverá ainda prover mecanismos para autenticação dos módulos de forma que um possível atacante não consiga acesso ao sistema nem às informações provenientes das comunicações;

- **Balanceamento de Carga**: deve ser possível ao sistema identificar um módulo que esteja sendo executado em uma máquina sobrecarregada e transferir suas funções para outro módulo igual que esteja em outra máquina menos sobrecarregada, para aumentar a performance do sistema e não prejudicar em demasia uma determinada máquina.

Todas estas características apresentadas visam tornar o sistema o mais flexível e robusto possível. O sistema proposto se baseará no NetTracker [16], um ambiente de gerenciamento de redes baseado em SNMP e herdará suas características mais importantes e sua estrutura básica. Devido à sua grande modularidade e flexibilidade, poucas mudanças serão necessárias para se construir um ambiente de segurança a partir do NetTracker. Todos os requisitos do sistema de segurança que não estiverem presentes no NetTracker serão acrescentados. Outras características fundamentais como Processamento Distribuído, Tolerância a Falhas, Comunicações Seguras e Balanceamento de Carga serão fornecidos pelo Visibroker [31], a implementação de CORBA escolhida para a implementação o sistema.

Integração Firewall-Sistema de Detecção de Intrusão

Como o desenvolvimento de um firewall foge ao objetivo deste trabalho, será desenvolvido um módulo de comunicação e controle do firewall com o sistema de segurança. Este módulo será um objeto de aplicação do sistema de segurança. Ele será responsável pela configuração do firewall com sistema de segurança e por responder as requisições de dados e estatísticas. Estas requisições são para análise do administrador ou para alimentar o sistema de detecção.

A integração do sistema de detecção de intrusão com o de segurança será feita através de um objeto do sistema de detecção que seja derivado de um objeto de aplicação do sistema de segurança. Este objeto será responsável pela comunicação entre os dois sistemas

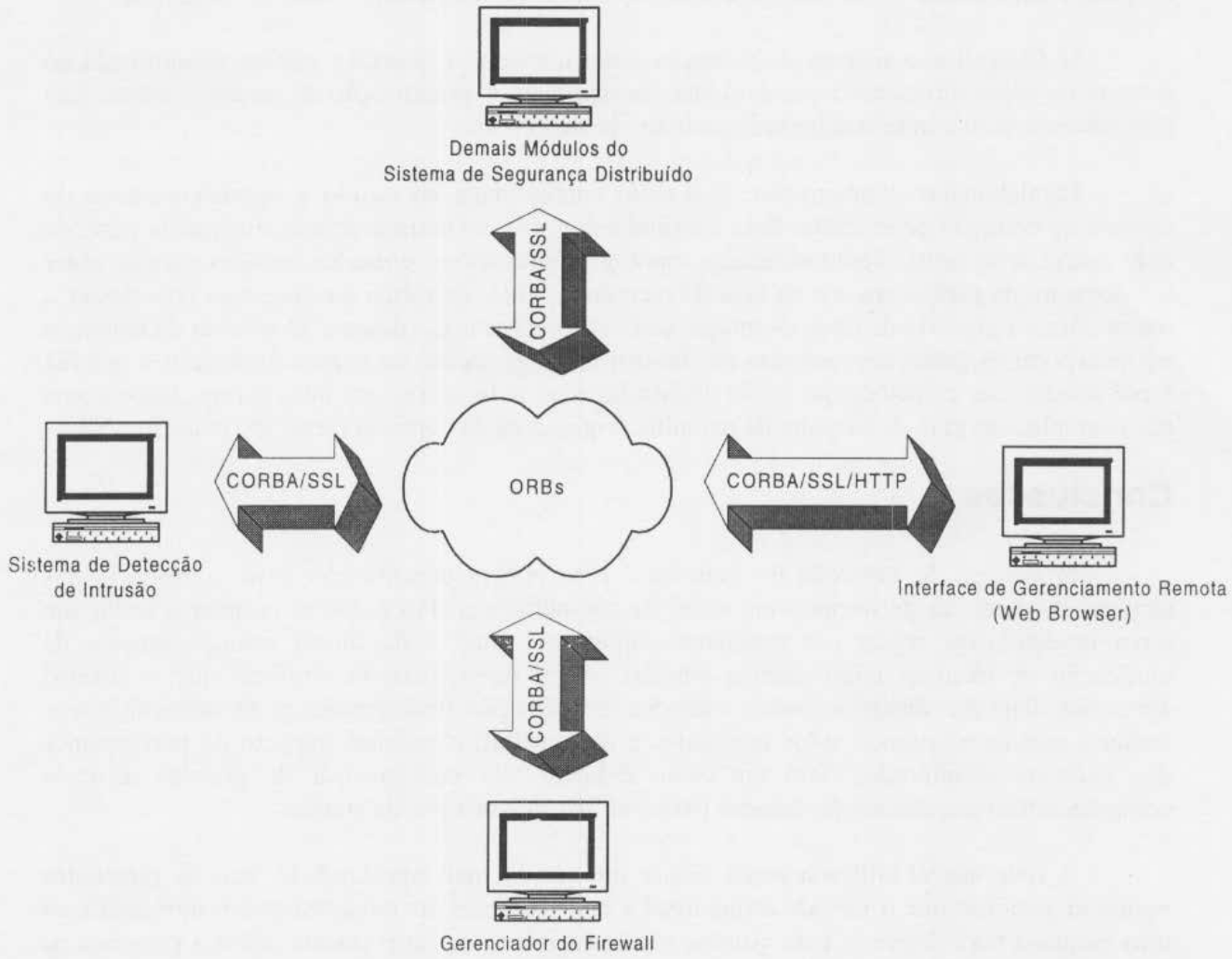

Figura 8 - Arquitetura de Comunicação dos Módulos

Desta forma, os dois sistemas estarão integrados no sistema de segurança como mostra a Figura 8. Todos os dados que deverão ser trocados deverão necessariamente passar pelo sistema de segurança para que a execução de cada sistema possa ser feita transparentemente para o outro. Será estudada a possibilidade de que, em casos extremos, como por exemplo quando o sistema de detecção perde a comunicação completa com o sistema de segurança, o sistema de detecção possa enviar mensagens broadcast pela rede para avisar o firewall de que algo está errado e este por sua vez, tomar uma atitude defensiva automaticamente.

Basicamente, a função do sistema de detecção de intrusão será procurar por comportamentos intrusivos nas conexões permitidas pelo firewall, fornecendo um grau de suspeita para as conexões de forma que o sistema de segurança possa tomar as devidas providências. O sistema de segurança pode apresentar ao administrador o nível de suspeita nas conexões das máquinas graficamente; ou ainda, pode automaticamente configurar o firewall para que bloqueie a conexão suspeita. Isto visa diminuir o tempo de resposta à uma ação intrusiva.

O sistema de detecção faz uma pré-seleção das conexões antes de enviá-las para serem analisadas. Esta pré-seleção se baseia em fatos anteriores que tenham ocorrido envolvendo as máquinas, os domínios ou os serviços da conexão. O firewall pode fornecer informações importantes neste sentido para melhorar a eficiência desta pré-seleção. Por exemplo, se uma determinada máquina tem muitas de suas conexões bloqueadas pelo firewall, ela já terá um alto

grau de suspeita e uma conexão válida proveniente desta máquina deverá ser considerada mais suspeita do que conexões de outras máquinas que nunca tiveram suas conexões bloqueadas.

O firewall e o sistema de detecção terão uma única interface gráfica subordinada ao sistema de segurança, sendo possível sua configuração e visualização de dados e estatísticas remotamente pelo administrador utilizando um browser web.

Paralelamente à integração, será dado continuidade ao estudo e aperfeiçoamento do sistema de detecção de intrusão. Essa continuidade será concentrada principalmente na parte de rede neural do sistema. Serão estudadas topologias alternativas e usadas técnicas para se obter um aumento da performance e na taxa de reconhecimento, de forma que se possa reconhecer o maior número possível de tipos de ataque com as menores taxas de erro. O sistema de detecção irá incorporar os dados provenientes do firewall em duas partes: no sistema especialista que faz a pré-seleção das conexões que serão analisadas e na rede neural que incorporará dados como por exemplo, um grau de suspeita da máquina originadora da conexão fornecido pelo firewall.

Conclusões

O sistema de detecção de intrusão é uma efetiva contribuição para a melhoria das técnicas de detecção de intrusos em redes de computadores. Foi possível projetar e testar um novo modelo para atacar um problema importante, atual e de difícil solução através da unificação de técnicas relativamente simples. Além disso, pode-se verificar que o sistema apresenta diversas vantagens sobre métodos de detecção tradicionais: pode detectar novos ataques, mesmo não tendo sidos mostrados à rede neural; não causa impacto na performance dos sistemas monitorados; tem um custo reduzido por não precisar de grandes recursos computacionais e apresenta facilidades para codificar novos tipos de ataque.

A rede neural utilizada ainda requer um estudo mais aprofundado, mas os resultados permitem concluir que o método é funcional e pode detectar um comportamento intrusivo com uma pequena taxa de erros. Este sistema não tenta inferir definitivamente sobre a presença de intrusão, ao invés, ele provê um grau de intrusão que irá indicar probabilidade de atividades intrusivas.

A integração deste sistema com um firewall em um único ambiente irá permitir que sistemas computacionais possam ter um maior grau de segurança do que o encontrado atualmente em soluções isoladas. Trabalhando em outro escopo, o sistema de detecção analisa diretamente os dados da conexão, enquanto que o firewall não tem acesso a estes dados. É esta característica que faz da união de um sistema de detecção de intrusão e de um firewall uma solução altamente eficiente. Além do fato de que esta integração permitirá respostas extremamente rápidas em casos de tentativas de ataque e de intrusão.

Agradecimentos

Os autores agradecem o apoio da FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo e da CAPES - Coordenadoria de Aperfeiçoamento de Pessoal de Ensino Superior, no financiamento deste projeto.

Referências

[1] R. Bace. *A New Look at Perpetrators of Computer Crime*. In Proc. 16th Department of Energy Computer Security Conference, 1994.

[2] P. Neumann & D. Parker. *A Summary of Computer Misuse Techniques*. In Proc. 12th National Computer Security Conference, pp. 396-407, 1989.
[3] Department of Defense, Trusted Computer Trusted System Evaluation Criteria, National Computer Security Center, DOD 5200.28-STD, Dec. 1985.
[4] H.S.Javitz & A.Valdez. *The SRI IDES Statistical Anomaly Detector*. In Proc. 1991 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May, 1991.
[5] J.R.Winkler & W.J.Page. *Intrusion and Anomaly Detection in Trusted Systems*. In Proc. Fifth Annual Computer Security Applications Conference, Tucson, AZ, pp.115-124. Dec,1990.
[6] D.E. Denning. *An Intrusion-Detection Model*. IEEE Trans. on Software Eng. Vol. SE-13, pp. 222-232, Feb.1987.
[7] T.F. Lunt et al. *IDES: A Progress Report*. In Proc. Sixth Annual Computer Security Applications Conference. Tuscon, AZ, Dec. 1990.
[8] T.F. Lunt et al. *A Real Time Intrusion Detection Expert System (IDES)*. Interim Progress Report, Project 6784, SRI International, May 1990.
[9] L.T. Heberlein et al. *A Network Security Monitor*. In Proc.1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, pp 296-304. May, 1990.
[10] L.T. Heberlein et al.. *Towards Detecting Intrusions in a Networked Environment*. In Proc. 14th DOE Conference on Computer Security. Concord, CA, pp 17-47. May 1991.
[11] L.T. Heberlein, K.N. Levitt and B. Mukherjee. *A Method to Detect Intrusive Activity in a Networked Environment*. In Proc. 14th National Computer Security Conference. Washington, DC, pp 362-371.Oct. 1991.
[12] A. Cansian, E. Moreira, A. Carvalho, J. Bonifácio Jr.: Network Intrusion Detection Using Neural Networks. In Proc. of International Conference on Computational Inteligence and Multimedia Applications, ICCIMA'97, Gold Coast, Australia, pp 276-280, Feb. 1997.
[13] A. Cansian, E. Moreira, R. Mouro, F. Morishita and A. Carvalho: An Adaptative System for Detecting Intrusion in Networks. In Proceedings of the III International Congress on Information Engineering, Buenos Aires, Argentina, pp 96-105, April 1997.
[14] A. Cansian, E. Moreira, J. Bonifácio and A. Carvalho: Modelo Adaptativo para Detecção de Comportamento Suspeito em Redes de Computadores. In the Proc. of the XV Brazilian Symposium on Computer Networks, SBRC'97, pp 51-60, São Carlos, SP, May 1997.
[15] J. M. Bonifácio Jr, A. Cansian, A. C. P. L.F de Carvalho, E. S. Moreira : Neural Networks Applied in Intrusion Detection Systems. To be published in IJCNN '98 International Joint Conference on Neural Networks, Anchorage, Alaska, May, 1998.
[16] R. B. Mouro, F. T. Morishita e E. S. Moreira: NetTracker: Uma Arquitetura Operacional Extensível para Ferramentas de Gerenciamento de Redes. In the Proc. of the XV Brazilian Symposium on Computer Networks, SBRC'97, pp 164-174 , São Carlos, SP, May 1997.
[17] M. J. Ranum, "An Internet Firewall", proceedings of World Conference on Systems Management and Security, 1992. ftp://decuac.dec.com/pub/docs/firewallfirewall.ps.
[18] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C. Ho, K. N. Levitt, B. Mukherjee.: "Intrusion Detection Systems (IDS): A Survey of Existing Systems and a Proposed Distributed IDS Architecture", University of California, http://www.jump.net/~snapp/papers.html.
[19] http://www.cert.org
[20] Rumelhart, D.E.; McClelland, J.L. and The PDP Research Group. *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press 1986.
[21] Carvalho, A.C.; Fairhurst, M.C. e Bisset, D.L. An integrated Boolean neural network for pattern classification, Pattern Recognition Letters, Vol. 15, pp. 807-813, August 1994.
[22] Wong A. J. Recognition of General Patterns Using Neural Networks, Biological Cybernetics, Springer-Verlag, Vol. 58, pp. 361-372, 1998.
[23] Haykin, S.; *Neural Networks: A Comprehensive Foundation*, Macmillan Publishing Co:/ IEEE Press, 1994.
[24] Lupo, C.J. Defense Applications of Neural Networks, IEEE Communications Magazine, Vol. 27, No. 11, pp. 82-88, November 1989.

[25]Rumelhart, D.E., Mcclelland, J.L.; *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, MIT Press 1986.
[26] Fahlman, S.E., An Empirical Study of Learning Speed, Technical Report, Carnegie Mellon University, September 1988.
[27] SCHIFFMANN, W., JOOST M., WERNER R. Optimization of Backpropagation Algorithm for Training Multilayer Perceptrons, Universidade de Koblenz,1993.
[28] http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html
[29] http://flying.fish.com/satan/
[30] http://iss.net/
[31] http://www.visigenic.com/prod/vbrok/vb30DS.html
[32] http://home.netscape.com/newsref/std/SSL.html
[33] http://www.omg.org