ANNIDA: Artificial Neural Network for Intrusion Detection Application
Aplicação da Hamming Net para Detecção por Assinatura

Lília de Sá Silva
Instituto Nacional de Pesquisas Espaciais (INPE)
lilia@dss.inpe.br

Adriana C. F. dos Santos
Instituto Nacional de Pesquisas Espaciais (INPE)
adriana.ferrari@lac.inpe.br

J. Demísio S. da Silva
Instituto Nacional de Pesquisas Espaciais (INPE)
demisio@lac.inpe.br

Antonio Montes
Centro de Pesquisas Renato Archer (CenPRA)
montes@lac.inpe.br

Resumo

Muitos desafios são enfrentados pelos profissionais da área de segurança em redes para proteger seus sistemas, impedindo ou minimizando o risco de serem violados através de ações maliciosas ou acesso não autorizado. Pesquisadores e analistas de rede têm trabalhado em conjunto para a obtenção de sistemas de detecção de intrusos mais eficientes, capazes de identificar e sinalizar ameaças cada vez mais inteligentes e audaciosas.

A proposta deste trabalho é apresentar o progresso do desenvolvimento de uma ferramenta de detecção de ataque baseada no uso da rede neural Hamming Net para processar assinaturas de ataque e identificar conteúdo malicioso em pacotes de rede TCP/IP. Neste artigo são apresentados o projeto da aplicação intitulada ANNIDA – Artificial Neural Network in Intrusion Detection Application, a arquitetura da rede neural empregada, bem como os resultados obtidos. Adicionalmente, são comentadas as possíveis alterações da ferramenta para os projetos futuros.

1. Introdução

Enquanto a maioria dos sistemas de detecção de intrusão (IDS) são construídos como sistemas especialistas baseados em regras para identificar ataques, uma quantidade limitada de pesquisa tem sido conduzida baseada na aplicação de redes neurais para tratar da deficiência inerente das abordagens baseadas em regras.

Um dos métodos mais tradicionais de detecção de intrusos à rede utiliza assinaturas de ataque. Neste método, uma base de dados é criada contendo seqüências de *strings*, denominadas assinaturas, que são consideradas indícios de um ataque.

Dentre as principais vantagens deste métodos pode-se citar: baixo número de falsos positivos; possível adoção de contra-medidas imediatas, mesmo para usuários com pouca experiência, permitindo corrigir rapidamente as falhas provenientes dos ataques; simplificação na quantidade de informação tratada; melhor desempenho em relação ao método baseado em anomalias, mesmo com grandes bases de assinaturas, principalmente pelo uso quase desprezível de operação de ponto flutuante.

Face às limitações e dificuldades do método baseado em regras e filtros, diferentes trabalhos apontam novos tipos de análise na busca de intrusões [12]. Estes estudos visam criar técnicas alternativas que melhorem a qualidade da análise de dados. Um dos melhores exemplos de técnica alternativa avançada é a aplicação de redes neurais encontrada em alguns trabalhos nesta linha de pesquisa [8] [9] [13][14][15].

As redes neurais podem ser entendidas como algoritmos que adquirem conhecimento sobre o relacionamento entre vetores de entrada e saída, generalizando essas relações para obter novos vetores de entrada e saída úteis ao fim proposto [2][3]. Essa técnica pode então ser usada em IDSs baseados em assinaturas para "aprender" traços de ataques e, então, procurá-los em um conjunto de dados. Além de acelerar o processo de busca de eventos ilegítimos ou tentativas de invasão, as redes neurais podem melhor representar o conhecimento sobre possíveis ataques.

Ao invés de processar instruções na seqüência, os modelos baseados em redes neurais simultaneamente exploram várias hipóteses usando vários elementos computacionais (neurônios) interconectados através de uma rede de pesos sinápticos [2][3]; o que acelera a análise do trafego.

A motivação inicial deste trabalho começou em 2004 com o estudo da possibilidade de aplicar a rede neural Hamming Net para classificar atividades ilegítimas no tráfego de rede [7], considerando que uma das vantagens desta rede neural é a capacidade de rápida classificação [4].

Durante o desenvolvimento desta pesquisa, houve a necessidade de se restringir o escopo inicial do projeto

para explorar dados de carga útil de pacotes de rede de modo *off-line* e utilizando um conjunto reduzido de padrões bem conhecidos, extraídos de arquivos de assinaturas do *Snort* [9].

Como resultado desta primeira etapa do trabalho, obteve-se um percentual de 70% de casamento do conjunto de entrada composto por conteúdo malicioso com as assinaturas introduzidas na rede neural em forma de vetores exemplares, e constatou-se a viabilidade de uso da Hamming Net para classificação de padrões de ataque. Verificou-se também que, quanto maior o número de exemplares, a rede alcançava a convergência mais rapidamente.

Diferente do trabalho anterior, que buscava somente uma assinatura (uma string com conteúdo malicioso ou único *content*) no conjunto de dados apresentado à rede neural, a nova contribuição descrita neste artigo envolve a detecção de ataques representados por mais de uma assinatura (múltiplas strings com conteúdo malicioso ou múltiplos *contents*). Resumindo, busca-se neste trabalho uma combinação de assinaturas, ampliando assim, a quantidade de ataques detectados.

Este artigo é dividido em sete seções, onde, em cada seção é tratado um assunto relacionado à aplicação desenvolvida. Iniciando pela seção 2, o conceito básico da rede neural Hamming Net e o seu funcionamento são descritos. Na seção 3 são apresentadas as características básicas das assinaturas do Snort. A arquitetura da rede neural usada neste trabalho é apresentada na seção 4. Na seção 5 a metodologia proposta para tratar da detecção de múltiplos *contents* é encontrada. Planos de teste são mostrados na seção 6. Concluindo, na seção 7 são discutidos os resultados e as possíveis metas a serem alcançadas nos próximos projetos.

2. Descrição da Hamming Net

Vários elementos computacionais não lineares, denominados neurônios artificiais, compõem uma rede neural. Estes elementos operam em paralelo e de modo massivamente distribuído e são interligados através de pesos sinápticos que se adaptam para obter melhor performance da rede [5]. A técnica de paralelismo massivo capacita a rede para trabalhar a elevadas taxas de processamento. Além disto, a grande quantidade de neurônios e conexões dão à rede a capacidade de tolerância a falhas, visto que a falha no processamento de alguns neurônios, em geral, não afeta a convergência da rede ao fim proposto [15].

Pesquisas recentes tratam da utilização de redes neurais classificadoras com a finalidade de buscar padrões anormais que indiquem possíveis intrusões dentro de um grande conjunto de dados [4][5][7][8]. A idéia básica é identificar, num conjunto de classes, a que melhor representa um determinado padrão desconhecido apresentado para análise (padrão de entrada).

Em continuação às pesquisas já desenvolvidas, empregou-se também para este trabalho a rede neural Hamming Net. Quando comparada com outras redes neurais observa-se que a Hamming Net não requer exaustivos treinamentos para aprender a reconhecer novos padrões de entrada e apresenta um crescimento de conexões de ordem linear relacionado ao numero de entradas, diferente da rede Hopfield, por exemplo, que apresenta um crescimento quadrático. Por isto a utilização da Hamming Net em nossa pesquisa.

Além disto, deve-se considerar que o resultado da classificação pela Hamming Net pode ou não representar um casamento perfeito (100% de similaridade), quando uma string do pacote de rede é comparada com a string de assinatura. O casamento perfeito indica a descoberta de um ataque conhecido, enquanto casamento de menor grau de similaridade, indica a possibilidade da descoberta de uma nova ameaça.

O aprimoramento deste trabalho consistiu na aplicação da Hamming Net em várias iterações com o objetivo de tratar da combinação de vários conteúdos maliciosos contidos na carga útil do pacote TCP/IP que são relacionados à mesma assinatura de ataque. No trabalho anterior, somente era possível a identificação de ataques representados por apenas um conteúdo malicioso.

Basicamente, o funcionamento da Hamming Net consiste em identificar a classe à qual uma dada entrada pertence, baseado no conjunto de padrões previamente introduzidos na rede, denominados "exemplares". A Hamming net trabalha cooperativamente com a Maxnet para a obtenção do valor mais semelhante desejado para a classificação. A figura 1 apresenta a estrutura desta rede.

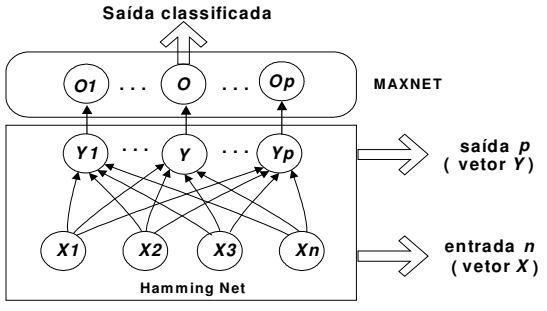

FIGURA 1. Estrutura da rede *Hamming Net*

Seja a rede da figura 1 uma Hamming Net preparada para identificar um padrão de entrada desconhecido composto por *n* bits em representação bipolar (vetor *X*). Nesta rede são previamente introduzidas *p* diferentes classes ou exemplares na Hamming Net (vetor *Y*), o que requer que a Hamming Net e a Maxnet tenham p neurônios de saída. A saída da Hamming Net após o seu processamento contém o maior valor que corresponde à classe que é mais semelhante ao padrão de entrada, com a menor distância de Hamming [6]. A MAXNET lê as saídas da Hamming Net e as processa até ressaltar este valor e ao mesmo tempo reduzir a zero os demais. O valor resultante da MAXNET corresponde à classe que mais se assemelha à entrada.

3. Assinaturas Snort

Snort é um sistema de detecção de intrusos baseado em assinaturas que usa uma combinação de regras e pré-processadores para analisar o tráfego de uma rede.

O processo de detecção do *Snort* [9] consiste em examinar o *payload* do pacotes recolhidos da rede, procurando por palavras-chave (*contents*) e pode ser usado para detectar uma grande variedade de ataques e *probes* de redes, tais como: *buffer overflow*, *stealth port scans*, CGI ataques, SMB *probes*, *fingerprinting*, entre outros.

A base do *Snort* é composta por um grande conjunto de regras - cerca de 3000 regras. Estas encontram-se disponibilizadas na Internet [10] e são constantemente atualizadas pelo grupo de desenvolvimento do *Snort*.

As regras do *Snort* são *stateless*, ou seja, cada regra inspeciona um e somente um pacote [1]. As regras por si mesmas não têm como conhecer eventos ocorridos em um pacote anterior ou no seguinte.

Uma assinatura ou regra do *Snort* é dividida em duas partes: "Cabeçalho" e "Opções", como mostra a figura 2.

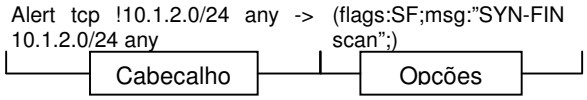

FIGURA 2. Formato da regra *Snort*

A parte "Opções" da regra *Snort* é separada da parte "Cabeçalho" por meio de parênteses.

No "Cabeçalho" são definidos os *hosts* e portas aos quais as "Opções" da regra se destinam. Na parte "Opções", são descritos os parâmetros a serem buscados na análise do conteúdo do pacote. Sendo verdadeiras as condições especificadas tanto no "Cabeçalho" quanto na parte "Opções", um alerta é emitido ou outro tipo de ação é disparado, sinalizando a ocorrência de um ataque.

Cada opção na regra é composta por uma palavra-chave, como *flags*, *msg*, *content*, entre outras. Por exemplo, uma regra para alertar sobre uma tentativa de *Buffer Overflow* no *Snort* tem o aspecto apresentado na figura 3.

```
Alert tcp any any -> 192.168.1.0/24 143 (content: "|E8C0 FFFF FF|/bin/sh";msg: "New IMAP Buffer Overflow detected!";)
```

FIGURE 3. *Snort* Rule Example

A mensagem de alerta "*ew IMAP Buffer Overflow detected!*" será apresentada quando for detectado um padrão anormal (indicado pela string "|E8C0 FFFF FF|/bin/sh") no tráfego TCP proveniente de uma porta qualquer de uma rede qualquer com destino à porta 143 da rede 192.168.1.x.

Após o casamento da primeira regra com o *payload* do pacote, o Snort dispara a ação correspondente a esta regra e não examina as demais.

3.1 Opção "content"

"*Content"* é uma das opções mais poderosas da regra *Snort* por prover a informação do conteúdo malicioso (*string* "*content*") a ser buscado no *payload* do pacote.

A pesquisa da *string* "*content*" no *payload* é considerada computacionalmente cara. Se o processo de busca não for construído de modo inteligente, o processamento da busca pode tornar-se muito lento. Embora os desenvolvedores do *Snort* tenham maximizado a eficiência nos códigos de busca por *strings* "*content*", esta ainda é uma operação cara.

A figura 4 apresenta o formato de um pacote TCP/IP, cujo campo "*payload"* contém a string "*content*" que é investigada nesta aplicação.

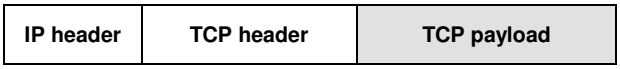

FIGURA 4. Formato do Pacote TCP/IP

A figura 5 ilustra um exemplo de regra Snort com duas *strings* "*content"* - "/bin/sh" e "|00 FA 00 FF|". Estas *strings* podem aparecer em qualquer ordem no *payload*, segundo [1]. Isto permite que múltiplas *strings* "*content*" sejam especificadas e, se qualquer uma delas casarem no processo de busca, a regra é disparada.

Sample rule:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 53 \
(msg: "EXPLOIT BIND tsig Overflow Attempt"; \
content: "|00 FA 00 FF|"; content: "/bin/sh";);
```

Sample output:

```
02/22-15:33:19.472301 ATTACKER:1024 -> VICTIM:53
UDP TTL:64 TOS:0x0 ID:6755 IpLen:20 DgmLen:538
Len: 518

<lines omitted to condense output>

00 3F 90 E8 72 FF FF FF 2F 62 69 6E 2F 73 68 00 .?..r.../bin/sh.
0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D ................
1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D .. !"#$%&'()*+,-
2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C EB ./0123456789:;<.
07 C0 00 00 00 00 00 3F 00 01 02 03 04 05 06 07 .......?........
08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 ................
18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 ........ !"#$%&'
28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 ()*+,-./01234567
38 39 3A 3B 3C EB 07 C0 00 00 00 00 00 3F 00 01 89:;<........?..
02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 ................
D8 FA FF BF D8 F7 FF BF D0 7C 0D 0B 04 F7 10 40 .........|.....@
22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 "#$%&'()*+,-./01
32 33 34 35 36 37 38 39 3A 3B 3C EB 07 C0 00 00 23456789:;<.....
00 00 00 3F 00 01 02 03 04 05 06 07 08 09 0A 0B ...?............
0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B ................
1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B .... !"#$%&'()*+
2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B ,-./0123456789:;
3C EB 07 C0 00 00 00 00 00 00 00 FA 00 FF <.............
```

FIGURA 5. Regra *Snort* com múltiplas

A saída apresentada no lado esquerdo da figura 5 exibe os caracteres hexadecimais contidos no *payload*, seguido da interpretação ASCII no lado direito da figura. A regra criada analisa o tráfego UDP destinado à porta 53 de um *host* que é externo à rede protegida.

Especificamente, a existência de duas *strings* é buscada: a primeira, expressa em hexadecimal - '00 FAN 00 FF', e a segunda em ASCII - '/bin/sh'. Ambas as *strings* podem aparecer no *payload* em qualquer ordem.

4. Arquitetura da Rede Neural

O objetivo da aplicação ANNIDA é classificar uma ou mais strings ilegítimas identificando padrões de ataque, com base no conjunto de assinaturas do Snort.

Para esta aplicação, foi construída uma arquitetura de Hamming Net composta por 14 neurônios de entrada (vetor X), N exemplares ou assinaturas (vetor d) e N neurônios de saída (vetor Y), como apresentado na figura 7. Os pesos da Hamming net são fixos e dados por w=d/2. O aprendizado na Hamming Net depende da inserção direta de conhecimento (assinaturas do Snort) nos pesos da rede. Os pesos e bias são fixos para todos os neurônios.

A Hamming Net gera o conjunto de saída intermediário (vetor Y) com dados dispostos em N neurônios. Estes dados são apresentados à MAXNET, como apresentado na figura 6, a qual retorna a classe (assinatura) que mais se assemelha ao padrão de entrada apresentado à *Hamming Net.*

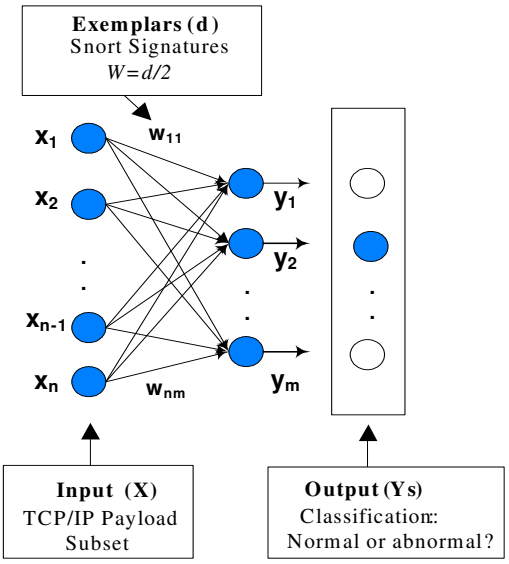

FIGURA 6. Diagrama da Rede Neural

5. Tratamento de Múltiplos "*Contents*"

Na aplicação descrita em [7] foi implementada a busca de uma *string* "*content*" única no pacote de rede.

Neste trabalho, o processo de busca de conteúdo malicioso foi aperfeiçoado, considerando que vários ataques somente podem ser detectados a partir da combinação de diferentes "*contents*" que compõem a mesma assinatura. Para isto, foi utilizado um arquivo de "*contents*" associados, onde a combinação de "*contents"* apresentada de cada linha do arquivo representa uma assinatura. Cada coluna deste arquivo armazena valor de um "*content*" em cada linha, como ilustrado na figura 7.

O método adotado consiste em apresentar à *Hamming Net* a primeira entrada para análise (conjunto $X = [X_1 \ldots X_n]$), por exemplo, a string {"xxxSITEyibolauiC|3A5C|ymkl"} e elementos da primeira coluna (conjunto s1={LIST, SITE, SITE, SITE, CWD}) do arquivo de assinaturas como padrões exemplares. O conjunto s1 é usado para cálculo dos pesos do primeiro nível da rede, gerando o conjunto w1. Após a entrada ser classificada, a primeira *string* maliciosa encontrada é sinalizada (neste caso,"SITE").

Elementos (strings) associados ao "content" malicioso encontrado são pesquisados na segunda coluna (conjunto s2) do arquivo de assinaturas. Todas as associações de "content" encontradas no conjunto s2 são agrupadas e fazem parte do subconjunto s2 – sub(s2)={CHMOD, EXEC, C|3A5C|}, como apresentado na figura 7. Este subconjunto é processado para o cálculo dos pesos (conjunto w2) que serão utilizados na próxima iteração da rede neural.

O subconjunto s2 associado ao "content" de s1 que foi classificado no estágio anterior é apresentado à Hamming Net, a qual produzirá uma nova classificação e sinalizará o próximo conteúdo malicioso encontrado.

Este procedimento se repete para todas as próximas associações encontradas até que não existam mais associações naquela linha do arquivo investigada. Então, a classificação final é obtida como a combinação de todos os conteúdos associados que foram sinalizados nos estágios anteriores. A classificação final do exemplo apresentado na figura 7 retorna a string de ataque "SITE Cl3A5Cl".

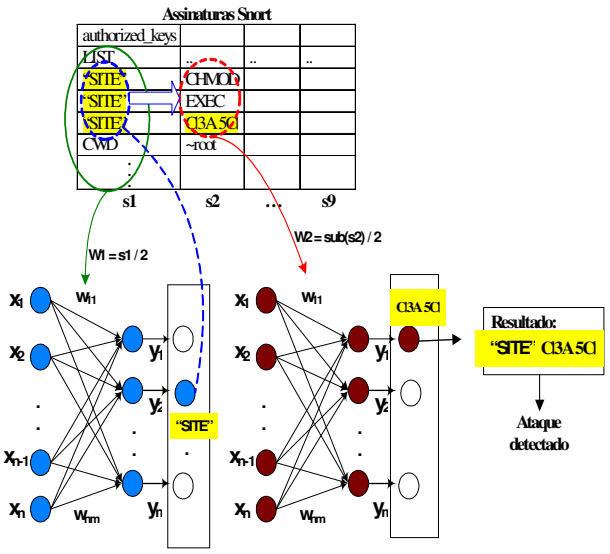

FIGURA 7. Diagrama do Método Proposto

6. Testes e Resultados

Na fase de teste da aplicação, foram utilizados conjuntos de dados de entrada, simulados a partir de assinaturas do *Snort*, contendo múltiplos "*contents*" de ataque conhecidos com ruídos. Linhas de dados de entrada com informações de ataques foram analisadas com uma máscara de tamanho variável, do tamanho das assinaturas inseridas na rede neural, em formato bipolar de 14 bits. O conjunto de dados exemplares utilizados para a classificação pela Hamming Net foi preparado a partir de arquivos do *Snort*, cujos dados foram representados em formato 14 bits.

Para os testes da aplicação, foram utilizados principalmente os seguintes parâmetros variáveis:

- NUMEXEMP – quantidade de assinaturas (linhas) no arquivo de exemplares
- MAXLEN – tamanho da maior assinatura no arquivo de assinaturas exemplares
- TOTLEN – total de tamanhos de máscaras
- NÍVEL – quantidade de strings "*contents*", cuja associação representa um ataque específico, contidas no conjunto de dados de carga útil apresentado à rede neural para classificação.

As assinaturas do Snort são dispostas em arquivos (.rules) e agrupadas por serviço de rede ou classificadas segundo o tipo de ataque. Dentre os experimentos realizados, constam neste artigo os resultados obtidos das classes de assinaturas: ddos, dns, exploit, finger, ftp, netbios, icmp e oracle, como apresentado na tabela 1.

TABELA 1. Resultados de Testes

Classe	Entrada	Dados de Exemplares	Classif.
dns	nivel 2	NUMEXEMP=19 MAXLEN=22 TOTLEN=11	100%
ftp	nivel 3	NUMEXEMP=69 MAXLEN=27 TOTLEN=16	100%
netbios	nivel 9	NUMEXEMP=430 MAXLEN=161 TOTLEN=79	80%
icmp	nivel 1	NUMEXEMP=11 MAXLEN=64 TOTLEN=8	100%
oracle	nivel 3	NUMEXEMP=227 MAXLEN=70 TOTLEN=40	70%
oracle	nivel 2	NUMEXEMP=227 MAXLEN=70 TOTLEN=40	100%
oracle	nivel 1	NUMEXEMP=227 MAXLEN=70 TOTLEN=40	80%
ddos	nivel 1	NUMEXEMP=227 MAXLEN=70 TOTLEN=40	80%
exploit	nivel 2	NUMEXEMP=77 MAXLEN=94 TOTLEN=34	100%
finger	nivel 1	NUMEXEMP=14 MAXLEN=14 TOTLEN=8	100%

7. Conclusões

A pesquisa anterior [7] confirmou a viabilidade de uso da rede neural Hamming Net para a rápida

detecção de conteúdo malicioso em conjuntos de dados e encorajou o desenvolvimento da aplicação ANNIDA.

ANNIDA foi projetada para detectar em modo off-line informação maliciosa em dados de pacote TCP/IP, simulados a partir de assinaturas Snort acrescidas de ruído e com o uso de múltiplos "*contents*" combinados que representam diferentes ataques.

O principal desafio deste projeto foi estabelecer uma metodologia para tratamento dos dados de entrada para a rede neural, tal que várias *strings* associadas fossem manipuladas e tivessem sua dependência preservada quando pesquisando um dado ilegítimo.

Os resultados obtidos foram bastante satisfatórios, principalmente na observação de assinaturas compostas por até 3 níveis de strings associadas que resultou em média 100% de classificação correta. Para a busca de nível maior, ou seja, por exemplo, a pesquisa da combinação máxima de 9 strings em regras Snort, o resultado médio obtido foi de 70%. Também resultados nesta faixa foram encontrados quando conjuntos maiores de exemplares foram introduzidos na rede neural, por exemplo, no uso de 227 exemplares e 3 strings associadas para pesquisa de ataques Oracle.

Devido às características da rede neural utilizada e a flexibilidade de determinação do grau de similaridade entre a string de ataque conhecida (exemplar) e a string analisada (entrada), é possível utilizar a aplicação para detectar, além de ataques conhecidos, informações de tentativas de ataques ou de novos ataques.

O objetivo em vista para futuros trabalhos envolve o teste com o conjunto todo de regras do Snort (atualmente com 3944 assinaturas) para avaliar o desempenho da aplicação e a precisão dos resultados para um grande conjunto de dados.

Um novo desafio é a compreensão de outros parâmetros (além da opção "content") envolvidos na semântica das regras Snort. Estas informações também são importantes e compreendem opções tais como "depth" e "offset" utilizadas em conjunto com a opção "*content"* para acelerar a busca de *strings* no pacote de rede.

Também deseja-se apresentar como resultado final, além da associação de *strings* encontradas no conjunto de entrada, representativas de um ataque, o tipo de ataque ocorrido.

Finalmente, deseja-se testar a aplicação para dados reais de pacotes que trafegam pela rede de computadores em uma janela de tempo pré-determinada.

8. Referências

[1] S. NORTHCUTT, *Network Intrusion Detection*. 3.ed. New York: New Riders Publishing, 2002. ISBN 0-73571-265-4.

[2] L. FAUSSET, *Fundamentals of Neural Networks: architectures, algorithms, and applications*. New York: Prentice Hall, 1994. ISBN 0-13-334186-0.

[3] S. HAYKIN, *Redes Neurais – Princípios e Prática*. 2.ed. Porto Alegre: Bookman, 2001. ISBN: 85-7307-718-2.

[4] C. HUNG, and S. LIN, Adaptative Hamming Net: A Fast-Learning ART1 Model without Searching. *Neural Networks*, v.8, n. 4, p. 605-618, 1995.

[5] R.P. LIPPMANN, An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, v. 4, p.4-22, abr. 1987.

[6] T. Downs, COMP3700 Machine Learning – 3E381 Neural Computing Lecture 8 Unsupervised Learning. Austrália, July, 2002. Available in: < *http://www.itee.uq.edu.au/~comp3700/lectures/lecture8.pdf* >. Accessed in: 24 nov. 2003.

[7] L.S. Silva, A. C. F. Santos, J. D.S. Silva, A. Montes, A Neural Network Application for Attack Detection in Computer Networks – International Joint Conference in Neural Networks (IJCNN), Budapeste, Hungria, 2004.

[8] L.S. Silva, A. C. F. Santos, J. D.S. Silva, A. Montes,Uma solução híbrida para detecção de anomalias em redes - IV WORCAP – Workshop dos Cursos de Computação Aplicada, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, São Paulo, 2004.

[9] B. Caswell, J. Beale, J. C. Foster, J. Posluns Snort 2 Sistema de Detecção de Intruso Open Source –, Editora Alta Books, Rio de Janeiro, 2003.

[10] Snort Rules Download, http://www.snort.org/rules - acessado em 09/03/2005

[11] R. M. Silva, M. A. G. M. Maia, Redes Neurais Artificiais Aplicadas À Detecção De Intrusos Em Redes Tcp/Ip, SSI – Simpósio de Segurança da Informação – ITA – São José dos Campos, SP, 2004.

[12] H., Lingxuan and H. Zhijun, Neural network-based intrusion detection systems, Source: Proceedings of the Sixth International Conference for You Computer Scientist: in Computer Science and Technology in New Century, 2001, p 296-298.

[13] W. Quanmin, and L. Weimin A model for intrusion detection based on fuzzy match and neural network**.** Source: Proceedings of the International Symposium on Test and Measurement, v 1, 2001, p 411-414.

[14] Comparison of BPL and RBF network in intrusion detection system Zhang CL, Jiang J, Kamel M Rough Sets, Fuzzy Sets, Data Mining, And Granular Computing Lecture Notes In Artificial Intelligence 2639: 466-470 2003.

[15] S. RUSSEL, and P. NORVIG Artificial Intelligence: A Modern Approach. New York: Prentice Hall, 1995. ISBN: 0- 13-103805-2.