Framework para Detecção e Filtragem de Alertas de Intrusão utilizando Redes Bayesianas

Everton Gamba Lermen, Gaspare Giuliano Elias Bruno

Ciência da Computação – Centro Universitário La Salle (UNILASALLE)
Av. Victor Barreto, 2288 – CEP 92.010-000 – Canoas – RS – Brasil

everton.lermen@gmail.com, gaspare@unilasalle.edu.br

Abstract. *The increase of computer networks and Internet use for spreading products and/or services, concern with security of the information and increase of attacks created a demand for tools that facilitates the identification of these attacks. The Intrusion Detection Systems - IDS appeared to supply this necessity. But the sprouting of new invasion techniques, researches that try to add intelligence to the IDS have been directed toward techniques that deal with decision problems, among them, Bayesian Networks - BN. This paper presents a proposal of framework that aggregate bayesian networks classification method to IDS's, making possible identification of new attacks and reduction of false alerts.*

Resumo. *O aumento das redes de computadores e utilização da Internet para divulgação de produtos e/ou serviços, a preocupação com a segurança das informações por elas trafegadas e o aumento dos ataques fez surgir uma demanda por ferramentas que facilitem a identificação destes ataques. Os Sistemas de Detecção de Intrusão - IDS surgiram para suprir esta necessidade. Mas, devido ao surgimento de novas técnicas de invasão, pesquisas que buscam adicionar inteligência aos IDS têm voltado seus olhares para técnicas que tratam problemas de decisão e incerteza, entre elas Redes Bayesianas - RB. Neste contexto, o artigo busca apresentar uma proposta de framework que agregue aos IDS's redes bayesianas como método de detecção, possibilitando a identificação de novos ataques e diminuição dos falsos alertas.*

1. Introdução

O crescente aumento das redes de computadores e popularização da Internet utilizada por empresas, instituições de ensino e pesquisa, usuários domésticos entre outros, seja para divulgar produtos e/ou serviços ou apenas conectar-se a este "mundo" digital, fez aumentar também a preocupação com a qualidade e segurança destas redes e das informações por elas trafegadas. Durante os últimos anos, o número, assim como a severidade dos ataques a redes de computadores, aumentou consideravelmente [Allen 2000]. No Brasil, conforme estatísticas de incidentes reportados ao CERT.br, somente nos primeiros três meses de 2008, foram reportados mais ataques que em todo o ano de 2002.

Métodos de ataques cada vez mais elaborados e, por outro lado, que exigem menos conhecimento do atacante, estão sendo utilizados para obter dados pessoais e informações sigilosas de empresas como: contas bancárias, senhas, cartões de crédito, entre outras. Frente a isto, técnicas para detectar ataques e/ou comportamento intrusivo,

tem sido foco de pesquisas. Estudos para melhorar a eficiência dos sistemas de detecção de intrusão vêm agregando, sob diferentes aspectos, redes bayesianas.

Neste contexto, este artigo tem como objetivo propor um *framework* que utilize redes bayesianas como método de classificação de alertas buscando: a) identificar ataques conhecidos e desconhecidos; b) gerar regras estáticas, para ferramenta *Snort*, dos ataques desconhecidos identificados; e c) diminuir o volume de falsos alertas, trazendo como principais diferenciais a proposta de utilização, variáveis selecionadas e algoritmos de construção e inferência.

Este artigo se divide em sete seções, contando esta introdução. Na seção 2 é abordado o conceito de sistemas de detecção de intrusão, bem como seus principais componentes – sensores e analisadores. A seção 3 apresenta o conceito de redes bayesianas e métodos de construção e inferência. Na seção 4 são discutidos dois trabalhos relacionados com a pesquisa. A seção 5, por sua vez, apresenta o modelo proposto, comparando com os trabalhos relacionados na seção anterior. Na seção 6 são expostos os resultados obtidos e, finalmente, na seção 7 são feitas as considerações finais.

2. Sistemas de Detecção de Intrusão

Os sistemas de detecção de intrusão são softwares ou hardwares que automatizam o processo de monitoramento e análise. Ao serem propostos, para complementar os níveis de segurança, permitem que as organizações protejam seus sistemas de ameaças que vêm junto com o aumento da conectividade em rede. Assim, um IDS pode ser definido como um sistema automático de detecção e alerta de qualquer natureza, onde uma intrusão tenha ocorrido ou esteja para acontecer [Axelsson 2006].

Um IDS pode ser dividido em dois componentes: sensores e analisadores. Os sensores podem ser classificados em duas categorias principais: baseados em computador e baseados em rede [Bace 2001][Chebrolu 2004][Scarfone 2007]. Sistemas de detecção de intrusão que utilizam sensores baseados em computador, oferecem maior precisão e confiabilidade em determinar quais processos e usuários estão envolvidos em um ataque; porém em contra partida, são mais suscetíveis a ataques por rodar no mesmo computador que está sendo monitorado. Já os sistemas de detecção de intrusão que fazem uso de sensores baseados em rede, são mais seguros, sendo muitas vezes, invisíveis para o atacante; mas também possuem uma visão limitada do ataque, sendo mais suscetíveis a falsos positivos e negativos.

Os métodos de análise, ou analisadores, mais utilizados são detecção por mau uso e detecção por anomalia [Axelsson 2006][Bace 2001][Chebrolu 2004][Scarfone 2007]. Sistemas de detecção de intrusão que fazem uso de analisadores que realizam detecção por mau uso, também são conhecidos como IDS's baseados em assinatura; identificando, normalmente, ataques já conhecidos com bastante eficiência. Por outro lado, dificilmente identificam variações de um ataque ou, um ataque do qual não possui assinatura.

3. Redes Bayesianas

Redes bayesianas - RB, também conhecidas como redes probabilísticas causais ou redes de confiança, têm sido cada vez mais utilizadas, em diferentes áreas do conhecimento,

para modelar domínios de problemas que contenham incerteza [Chebrolu 2004][Jemili 2007][Kruegel 2003].

Uma RB pode ser descrita como um grafo acíclico dirigido - GAD, onde os nodos representam variáveis randômicas e os arcos representam as diretas dependências probabilísticas entre eles. Mais precisamente, para um GAD, $G = (V, E)$, onde V representa um conjunto de nodos (ou vértices) e E um conjunto direto de conexões (ou arcos) entre pares de nodos. A distribuição probabilística comum, $P(X_v)$, sobre o conjunto de variáveis (normalmente discreta) X_v posicionada por V pode ser fatorada como [Kjaerulff 2008]:

$$P\ (X_v) = \prod_{v \in V} P(X_v | X_{pa(v)}) \qquad (1)$$

onde $X_{pa(v)}$ traz o conjunto de variáveis pai X_v para cada nodo $v \in V$. A fatoração da equação expressa um conjunto de suposições independentes, que são representadas pelo GAD, em termos de pares de nodos, que não estão diretamente conectados um ao outro por um arco direto. É a existência destas suposições de independência e de um conjunto pequeno de pais para cada nodo, que permite especificar a probabilidade condicional e executar, de forma eficiente, a inferência em uma rede bayesiana.

Uma RB $\mathrm{N} = (\mathrm{X}, \mathrm{G}, \mathrm{P})$ é composta por [Kjaerulff 2008]:

- um GAD $G = (V, E)$ com nodos $V = \{v_1, \ldots, v_n\}$ e arcos diretos E
- um conjunto de variáveis aleatórias, X, representada pelos nodos de G
- um conjunto de distribuições de probabilidade condicional, P, contendo uma distribuição, $P\left(X_v \middle| X_{pa(v)}\right)$, para cada variável aleatória $X_v \in X$

Uma RB pode ser construída de forma manual, automática (através de uma base de dados), ou da combinação manual e base de dados. Nesse último método de construção, parte do conhecimento sobre a estrutura é definida manualmente, misturando os parâmetros com informações estatísticas extraídas da base de dados.

A construção manual de uma RB pode ser uma tarefa difícil, exigindo grande habilidade e criatividade, como também um bom conhecimento do domínio do problema. Uma vez construída, seja de forma manual ou automática, os parâmetros da RB podem ser continuamente atualizados com novas informações. Assim, o modelo inicial fornecido é gradualmente aprimorado.

Independente da utilização de dados, completos ou incompletos, existem dois paradigmas principais para construção de uma RB: o de independência condicional e o de busca e pontuação. O algoritmo PC, batizado com as iniciais de seus desenvolvedores, Peter Spirtes e Clark Glymour em 1991, está inserido no paradigma de independência condicional. Seu trabalho é procurar um RB que represente o relacionamento independente entre as variáveis em uma base de dados [Kjaerulff 2008].

O algoritmo *Expectation Maximization* - *EM* é um método de estimação a partir de dados incompletos. Basicamente, se alguma variável foi algumas vezes observada e outras não, este algoritmo utiliza os casos para os quais as variáveis foram observadas, para aprender a predizer seus valores quando não. Este método está inserido dentro do paradigma de busca e pontuação [Kjaerulff 2008][Luna 2004].

A realização de inferência em uma RB é, geralmente, um problema *NP-difícil*; inclusive as inferências aproximadas consistem em um problema *NP-difícil*. Felizmente, eficientes algoritmos foram desenvolvidos, tornando possível realizar inferência em RB em frações de segundos. Contudo, esta eficiência é diretamente dependente da estrutura do GAD [Kjaerulff 2008].

Normalmente, uma RB representa indicação causal do tipo $X \rightarrow Y$, onde X é uma causa de Y e, também, onde Y normalmente assume o papel de um efeito perceptível de X. Esse efeito, tipicamente, não pode ser observado, sendo necessário derivar à distribuição de probabilidade posterior $P(X|Y = y)$, dada a observação $Y = y$, utilizando a distribuição prévia $P(X)$ e a distribuição de probabilidade condicional $P(Y|X)$ especificada no modelo. Thomas Bayes criou o famoso teorema de Bayes para realizar este cálculo:

$$P(X|Y = y) = \frac{P(Y = y|X)P(X)}{P(Y = y)}, \qquad (2)$$

onde $P(Y = y) = \sum_x P(Y = y|X = x)P(X = x)$. Este teorema teve um papel fundamental na inferência estatística, porque a probabilidade de uma causa pode ser pressuposta, quando seu efeito for observado.

Existem, basicamente, dois métodos de inferência, os denominados exatos e os aproximados. Árvore de junção faz parte dos métodos de algoritmos denominados exatos, onde o cálculo das probabilidades é realizado *a posteriori*, através de somatório e combinações de valores. O objetivo é construir uma estrutura de dados que pode ser utilizada para calcular qualquer consulta através da passagem de mensagens na árvore [Jemili 2007] [Kjaerulff 2008] [Luna 2004].

O algoritmo de Monte Carlo faz parte do grupo de métodos aproximados, utilizando técnicas de simulação para obter valores aproximados das probabilidades. Este algoritmo gera um conjunto de amostras, escolhidas aleatoriamente, realizando inferência sobre elas. A precisão dos resultados está diretamente relacionando com o tamanho da amostra e, diferentemente dos métodos exatos, a estrutura da rede não é relevante para a realização do cálculo de inferência [Heckerman 1995].

4. Estado da Arte

Com o objetivo de entender e analisar os diversos problemas, sobre a utilização de redes bayesianas em um sistema de detecção de intrusão, foram analisadas diversas iniciativas que abordam o assunto. Os trabalhos aqui relacionados são [Jemili 2007] e [Abouzakhar 2003].

O artigo [Jemili 2007] apresentado propõe a utilização de redes bayesianas como método de identificação de ataques em sistemas de identificação de intrusão. O *framework* utiliza o algoritmo *K2*, limitando em quatro, o número de parentes de cada nodo e utilizando dois métodos de classificação: a) Normal e Ataque; b) *DOS*, *Probing*, *R2L*, *U2R* e Outros; para construção / aprendizagem, sobre a base do *DARPA'99*.

Foram consideradas as seguintes variáveis, nesta respectiva ordem: *protocol_type*, *service*, *land*, *wrong_fragment*, *num_failed_logins*, *logged_in*,

root_shell, *is_guest_login*, *attack_type*. Optou-se também por utilizar o algoritmo árvore de junção para realizar inferências sobre a rede.

Um ponto importante, mas que deve ser considerado com cuidado, é o fato de ocorrer uma realimentação da RB. Este procedimento deve ser adotado, quando se tem uma base grande de conhecimento; pois, caso contrário, pode ocorrer erro de classificação, prejudicando toda a rede.

O artigo [Abouzakhar 2003], por sua vez, também apresenta uma abordagem para identificação de ataques em sistemas de detecção de intrusão, utilizando redes bayesianas. Contudo, os algoritmos de construção / aprendizagem e inferência não são informados. Apenas é informado que, foi utilizada árvore de decisão, para transformar as variáveis contínuas em discretas. Como no artigo [Jemili 2007], este também utiliza a base do *DARPA* para construção / aprendizagem da rede. Porém, as variáveis utilizadas não são iguais – *protocol_type*, *service*, *count*, *srv_count*, *attack_type*, *rerror_rate*, *srv_rerror_rate*, *duration*, *serror_rate*, *srv_serror_rate* – montando, dessa forma, a rede bayesiana de forma distinta.

Diferentemente do artigo anterior, é utilizando um método conhecido como *Lift Chart*[1] para validar a aprendizagem da RB.

5. Modelo Proposto

O modelo proposto visa agregar pró-atividade ao IDS, tornado possível identificar novos ataques. Para adquirir esta *inteligência*, é necessário identificar o perfil de acesso normal e de ataque do ou dos computadores monitorados. Este perfil, também chamado de base de conhecimento, é utilizado para treinar a rede bayesiana. É com base nestas informações que os acessos passam a ser classificados em normal ou ataque.

A eficiência deste *framework* proposto, ou de qualquer IDS que se utilize de redes bayesianas, está diretamente relacionada à fase de treinamento e, conseqüentemente, a base de conhecimento. Quanto maior e mais confiável for a base de conhecimento, mais eficaz será a identificação de acessos normais e de ataques; diminuindo, assim, o número de falsos positivos e falsos negativos. Os diferenciais apresentados neste modelo, com relação aos modelos estudados são:

a) Estrutura da rede: optou-se por construir a estrutura da rede, de forma manual, montando assim, uma *naive-RB*. Já para construir o conhecimento, ou seja, aprender as probabilidades, o algoritmo EM foi escolhido, por possibilitar esta construção de conhecimento, a partir de dados incompletos, e por estar disponível na ferramenta Netica[2].
b) Variáveis contempladas
    - ip_origem: identifica o endereço IP do computador que originou o acesso.
    - ip_destino: identifica o endereço IP do computador ao qual o acesso foi direcionado.

[1] O lift chart fornece uma imagem geral do desempenho da previsão de um modelo de mineração de dados determinado, durante um conjunto de dados especificado, comparando o desempenho de previsão do modelo de mineração com um modelo ideal e um modelo aleatório [Apollo 2005].

[2] Netica é um programa utilizado para se trabalhar com redes bayesianas e diagramas de influência [Norsys 2008].

- porta_destino: identifica o serviço acessado no computador destino.
- mem: indica o nível de utilização de memória do computador destino.
- cpu: indica o nível de utilização de processamento do computador destino.
- ataque: pode conter apenas dois valores, sim ou não.

c) Proposta de utilização: por ser uma técnica custosa computacionalmente, este *framework* é proposto para ser utilizado, não em um ambiente de produção, onde o tempo de resposta é crítico, mas sim em frente a uma *honeynet*, ou seja, um ambiente preparado para ser invadido, não comprometendo assim, a performance de acesso da rede de produção. Também se optou por gerar regras estáticas dos novos ataques identificados, para IDS's baseados em regras.

Visando ilustrar a RB criada, a Figura 1 abaixo traz a direita, a rede bayesiana proposta no artigo [Jemili 2007], ao centro a rede aqui proposta e a esquerda, a rede proposta no artigo [Abouzakhar 2003].

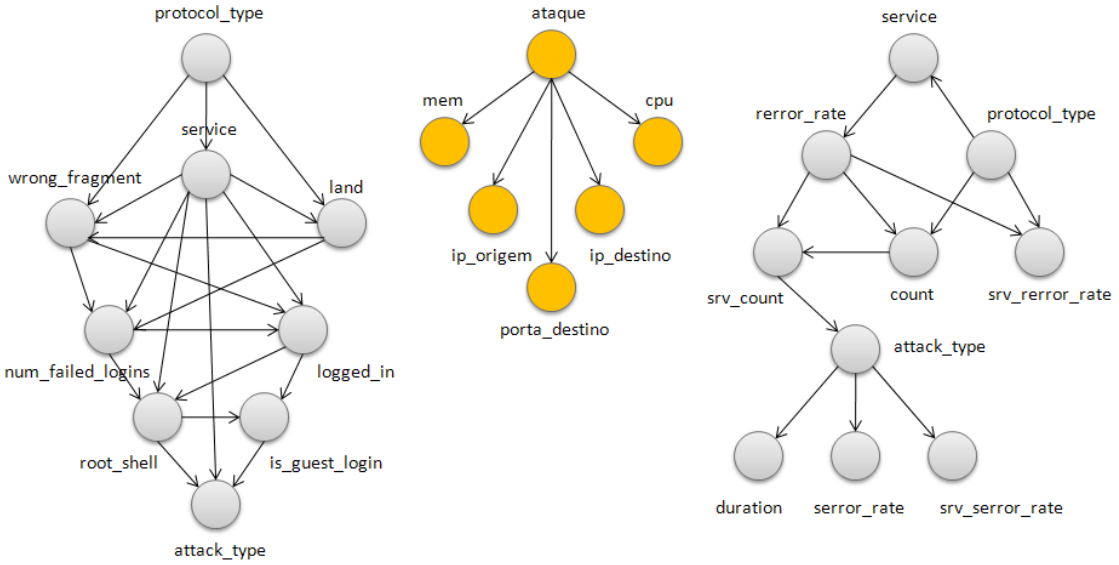

Figura 1. Comparação entre modelos de redes bayesianas.

Para validar o *framework* proposto foi necessário montar um ambiente, onde ataques e tráfegos normais de rede pudessem ser simulados de forma controlada; garantindo assim, confiabilidade para a fase de validação. Para isto, foram utilizados três computadores, denominados respectivamente: invasor, *honeywall* e vítima. Como o objetivo não era confrontar o *framework* em volume de ataque, mas sim sua eficiência, optou-se por realizar três tipos de ataques:

- Ataques de inserção de código em servidores WEB (IIS/ASP);
- Ataques de força bruta a servidores SSH;
- Ataques de inserção de código a servidores SQL Server.

Foram realizados 201 ataques e 100 acessos normais, para posterior construção da RB, montagem do cenário de treinamento e cenários de teste. Para cada rodada foram gerados dois arquivos, um com os *logs* coletados a partir da interface de rede, e outro com as informações de processamento e memória. Este método foi adotado para identificar o perfil normal e o perfil em ataque do computador *vítima*. As informações

foram coletadas com o auxílio da ferramenta TCPDUMP e NET-SNMP, sendo armazenadas em um banco de dados MySQL.

Durante a construção da RB, verificou-se a necessidade de realizar um tratamento da base de conhecimento. Um ponto considerado importante foi a conversão dos valores numéricos das variáveis memória (mem) e processamento (cpu) em baixa, média e alta. A conversão foi realizada analisando a faixa de valores destas variáveis, no perfil normal e no perfil de ataque. A Tabela 1 apresenta duas conversões, trazendo a faixa de valores e sua respectiva descrição. Com as faixas de valores utilizadas na primeira conversão, foi construída a RB. Durante o treinamento, devido esta distribuição, identificou-se que um ataque que gerasse uma utilização alta de memória (na faixa entre 66 a 100), resultava num falso negativo; ou seja, um ataque era considerado um tráfego normal de rede. Por isso, foi necessária uma nova conversão, redistribuindo a faixa de valores da variável memória.

Tabela 1 – Faixa de valores definida para cada variável de memória e processamento.

Variável	Faixa de valores (1ª conversão)	Faixa de valores (2ª conversão)	Descrição
MEM	1 a 35	1 a 35	Baixa
	36 a 65	36 a 50	Média
	66 a 100	51 a 100	Alta
CPU	1 a 25	1 a 25	Baixa
	26 a 50	26 a 50	Média
	51 a 100	51 a 100	Alta

Com a rede construída e já treinada com as informações da base de dados, iniciou-se o processo de inferência, para verificar se o conhecimento adquirido estava correto. Para a realização das inferências, foi utilizado o algoritmo árvore de junção, disponível na ferramenta Netica.

Para realizar a validação do *framework* foi criado um cenário de treinamento, além de quatro cenários de teste. Durante os testes foram realizadas inferências na rede bayesiana, ocasionando alterações na variável ataque, resultando com que um tráfego seja considerado normal ou ataque. De posse das informações, o comportamento do *framework* foi analisado quanto:

a) ao número de falsos positivos e falsos negativos, em relação ao tráfego conhecido;
b) ao número de falsos positivos e falsos negativos, em relação ao tráfego desconhecido.

6. Resultados

Os resultados aqui descritos foram obtidos através da realização de quatro cenários de teste. No primeiro cenário, a RB foi confrontada com um ataque conhecido, ou seja, mapeado durante a fase de treinamento – um ataque, partindo de um computador conhecido por realizar ataques. O objetivo deste cenário era verificar a ocorrência de falso negativo. Neste cenário, as inferências realizadas resultaram na indicação de 100% que o tráfego analisado é realmente um ataque, ou seja, 0% de falso negativo.

O segundo cenário confrontou a RB com um acesso normal, também mapeado durante a fase de treinamento – o acesso partiu de um computador conhecido por realizar acessos desta natureza. O objetivo deste cenário era verificar a ocorrência de falso positivo. Novamente, as inferências realizadas resultaram na indicação de 100% que o tráfego analisado é realmente um acesso normal, resultando em 0% de falso positivo.

No terceiro cenário, um computador, que durante a fase de treinamento, só havia realizado acesso normal, realiza um ataque. Embasado, fortemente, nas variáveis de memória e processamento, a rede verificou um percentual de 100% de possibilidade deste acesso, ser um ataque, ou seja, 0% de falso negativo.

Neste último cenário a RB é confrontada com um novo ataque, ou seja, as variáveis *ip_origem* e *porta_destino*, não foram identificas durante o treinamento. Mesmo não conhecendo o computador de origem e a porta de destino, a RB identificou com 96,9% o acesso como sendo um ataque, ou seja, 3,15% de ser um falso positivo. Apesar do elevado percentual, esta informação deve ser analisa com cuidado, pois pode ocorrer que, mesmo para um acesso normal a este serviço, a utilização de processamento e memória sejam elevados; ocasionando uma mudança no perfil de acesso normal e de ataque. Logo, se este fosse um acesso normal, ocorreria um falso positivo.

Caba ressaltar novamente, que quanto maior e mais confiável for a base de conhecimento, melhor e mais eficiente será a identificação de acesso normal e ataque.

7. Considerações Finais

Ao longo deste artigo foi apresentado um modelo de *framework* para sistema de detecção de intrusão utilizando redes bayesianas. Os resultados aqui apresentados são de grande relevância, porque possibilitam que outras propostas sejam norteadas por este estudo.

Durante a realização dos cenários de testes, tornou-se possível identificar que a utilização de redes bayesianas obteve sucesso na identificação de ataques e tráfego normal de rede – seja ele conhecido ou não. A utilização das variáveis de memória e processamento contribuiu diretamente para estes resultados. Frente a esses resultados, a utilização de redes bayesianas em sistemas de identificação de intrusão mostrou-se viável, uma vez que foi baixo o percentual de falsos positivos e falsos negativos.

As potencialidades deste trabalho estão: a) na utilização de uma *naive-Rede* bayesiana, que possibilita maior independência das variáveis; b) na utilização das variáveis de processamento e memória; e c) na proposta de utilizar este *framework* em uma *honeynet*, não limitando sua utilização em conseqüência do tempo de resposta – ao transformar o conhecimento aprendido pela rede bayesiana gerando regras estáticas, para serem utilizadas em sistemas de identificação de intrusão desta natureza, em uma rede de produção, onde o tempo de resposta é importante.

Este trabalho, entretanto, possui algumas limitações: a) restringiu sua base de conhecimento a apenas três ataques, que tiveram características distintas no perfil de acesso normal e no perfil de ataque; b) uma ferramenta escolhida – Netica – necessita de licença para operar com total funcionalidade; c) as regras estáticas geradas, a partir da base de conhecimento da rede bayesiana, são compatíveis apenas com a ferramenta

Snort; d) não leva em consideração o tempo de resposta para identificação de um acesso; e e) não foi prevista, nesta fase, realimentação da RB com novos conhecimentos.

Tendo em vista as contribuições e as limitações identificadas, seria interessante aprimorar o desenvolvimento do *framework*, agregando interfaces gráficas de controle e gerenciamento. Como alternativa, poderia ser utilizada a ferramenta Weka e a linguagem de programação Java. Também é válida a busca por uma alternativa aos scripts utilizados para obter as informações de processamento e memória do computador vítima.

A RB construída mostrou eficiência, contudo, a inclusão de novas variáveis pode vir a contribuir para o aprimoramento da rede. Algumas variáveis identificadas, mas não utilizadas neste trabalho são: número de conexões realizadas em certa fatia de tempo e detalhamento do pacote.

Referências

Allen, J., Christie, A., Fithen, W., Mchugh, J., Pickel, J. e Stoner, E. (2000) "State of the Practice of Intrusion Detection Technologies". Technical Report – CMU/SEI-99-TR-028. Carnegie Mellon Software Engineering Institute, Pittsburgh.

Axelsson, S. e Sands, D. (2006) "Understanding Intrusion Detection through Visualization". Springer.

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

Chebrolu, S., Abraham, A. e Thomas, J. P. (2004) "Feature deduction and ensemble design of intrusion detection systems". Computers & Security.

Scarfone, K. e Mell P. (2007) "Guide to Intrusion Detection and Prevention Systems (IDPS)". National Institute of Standards and Technology.

Jemili, F., Zaghdoud, M. e Ahmed, M. B. (2007) "A Framework for an Adaptative Intrusion Detection System using Bayesian Network". IEEE.

Kruegel, C., Mutz, D., Robertson, W. e Valeur, F. (2003) "Bayesian Event Classification for Intrusion Detection". IEEE.

Kjaerulff, U. B. e Madsen, A. L. (2008) "Bayesian Networks and Influence Diagrams, A Guide to Construction and Analysis". Springer.

Luna, J. E. O. (2004) "Algoritmos EM para Aprendizagem de Redes Bayesianas a partir de Dados Incompletos". Universidade Federal de Mato Grosso do Sul.

Heckerman, D. (1995) "A Tutorial on Learning With Bayesian Networks". Technical Report – MSR-TR-95-06. Microsoft Research, Advanced Technology Division.

Abouzakhar, N. S., Gani A., Manson, G. e Abuitbel, K. D. (2003) "Bayesian Learning Networks Approach to Cybercrime Detection". PGNet.

Apollo Data Technologies. (2005) "Inventory Predictive Modeling via Microsoft SQL Server 2005 Analysis Services". SQL Server Technical Article, Microsoft Corporation.

Norsys Software Corporation. (2008) "Netica Application", http://www.norsys.com/