Utilização de Sistemas Imunológicos Artificiais

para Sistemas de Detecção de Intrusão

Rodrigo Damasceno Marangon

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Prof. Eduardo Pagani Julio

Juiz de Fora, MG

Dezembro de 2009

Rodrigo Damasceno Marangon

Utilização de Sistemas Imunológicos Artificiais para Sistemas de Detecção de Intrusão

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Eduardo Pagani Julio

JUIZ DE FORA - MG
DEZEMBRO, 2009

Utilização de Sistemas Imunológicos Artificiais para Sistemas de Detecção de Intrusão

Rodrigo Damasceno Marangon

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 16 de dezembro de 2009.

Comissão Examinadora:

_____

Eduardo Pagani Julio orientador.
MSc em Computação / UFF

_____

Marcelo Lobosco
DSc em Engenharia de Sistemas e Computação COPPE/UFRJ

_____

Ana Paula Couto da Silva
DSc em Engenharia de Sistemas e Computação COPPE/UFRJ

JUIZ DE FORA
DEZEMBRO, 2009

Resumo

Este trabalho apresenta Sistema de Detecção de Intrusão integrado com Sistemas Imunológicos Artificiais, e suas principais características. São apresentados os principais conceitos de um detector de intrusão e as variações mais utilizadas deste software. O assunto abordado mostra o funcionamento básico do Sistema Imunológico por trás do detector, e são detalhados conceitos, história e políticas de segurança. Por fim o trabalho apresenta alguns testes de detecção de intrusão com utilização de um Sistema Imunológico Artificial.

Abstract

This paper presents an Intrusion Detection System integrated with Artificial Immune Systems, and its main features. Are presented the main concepts of an intrusion detector and the mostly used variations used of this software. The issue addressed shows the basic functioning of the Immune System behind the detector, and are detailed concepts, history and security policies. Finally, the paper presents some tests of intrusion detection with utilization of an Artificial Immune System.

Agradecimentos

Agradeço a minha família pelo incentivo e por apostarem em mim. Ao meu avô José (in memorian) e aos meus amigos Eduardo e Jane pelo imprescindível apoio em momentos difíceis. A todos os professores que de alguma forma me prepararam para realização deste trabalho.

Meus sinceros agradecimentos ao meu orientador Eduardo Pagani Julio pelo incentivo, apoio e orientação deste trabalho.

Sumário

Lista de Figuras

Lista de Reduções

3DES	*Triple Data Encryption Standard*
ADAM	*Audit data analysis and mining*
AES	*Advanced Encryption Standard*
ARPAnet	*Advanced Research Projects Agency Network*
CERT	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança
DARPA	*Defense Advanced Research Projects Agency*
DIDS	*Distributed Intrusion Detection System*
FTP	*File Transfer Protocol*
DDoS	*Distributed Denial of Service*
DES	*Data Encryption Standard*
DoS	*Denial of Service*
GPG	*GNU Privacy Guard*
HBD	*Host-based detection*
HIDS	*Host-based Intrusion Detection System*
HTTP	*HyperText Transfer Protocol*
ICN	*Los Alamos National Laboratory's Integrated Computing Network*
IDS	*Intrusion Detection System*
ISO	*International Organization for Standardization*
IP	*Internet Protocol*
IPTO	*Information Processing Techniques Office*
MIDAS	*Multics Intrusion Detection and Alerting System*
MIT	Massachusetts *Institute of Technology*
NAT	*Network Address Translation*
NIDS	*Network-based Intrusion Detection System*
NBD	*Network-based detection*
PAMP	*Pathogen Associated Molecular Pattern*
PGP	*Pretty Good Privacy*
PHP	*Hypertext Preprocessor*
SNMP	*Simple Network Management Protocol*
SSL	*Secure Socket Layer*

SSH	*Secure Shell*
SRI	Stanford Research Institute
UCLA	*University of California*

Capítulo 1 – Introdução

A *Internet* foi resultado de pensadores visionários no começo dos anos 60 que viram um grande potencial em computadores trocando informação para pesquisas e desenvolvimento nos campos científico e militar [HOWE, 2009].

J.C.R. Licklider propôs uma rede global de computadores em 1962 numa série de memorandos discutindo sobre o conceito de "Rede Galática" [ONENESS, 2009], e saiu para a DARPA (*Defense Advanced Research Projects Agency* – Agência de Pesquisas em Projetos Avançados da Defesa) no fim de 1962 para ser o chefe do IPTO (*Information Processing Techniques Office* – Escritório de Técnicas de Processamento da Informação). Em abril de 1963, mandou um memorando para seus colegas informando que superou os primeiros desafios para estabelecer uma rede de comutação de tempo com os softwares da era [LICKLIDER, 1963].

Leonard Kleinrock desenvolveu a teoria da comutação de pacotes, que é a base das conexões de *Internet* atuais [KLEINROCK, 1996]. Quando Lawrence Roberts conectou em 1965 um computador do MIT com um da Califórnia usando conexão discada mostrou que era possível ter uma rede de amplo alcance, e a teoria de Kleinrock fora confirmada.

Então Roberts trabalhou com Licklider em 1966 e desenvolveu um plano para o projeto da ARPAnet(*Advanced Research Projects Agency Network* – Rede da Agência de Pesquisa em Projetos Avançados) [ROBERTS, 2009].

Em 29 de outubro de 1969 foi transmitido o primeiro pacote em redes comutadas entre os nós do laboratório de Kleinrock na UCLA (*University of California* – Universidade da Califórnia) e o laboratório de Douglas Engelbart no SRI (*Stanford Research Institute* – Instituto de Pesquisa de Stanford) [SUTTON, 2004].

No fim de 1969 a ARPAnet era constituída por quatro nós, e em 1972 a ARPAnet foi apresentada publicamente por Robert Kahn, na Conferência Internacional sobre Comunicação por Computadores [INTERNATIONAL, 1972].

No fim dos anos 70, aproximadamente 200 máquinas estavam conectadas a ARPAnet, e ao fim da década de 80, o número de computadores conectados chegou a cem mil [ROBERTS, 2009].

Na década de 90 a *Internet* se iniciou como evolução da ARPAnet, agora com objetivo comercial. Ainda em 1990, 300 mil computadores estavam conectado à rede, e é criado o primeiro serviço comercial de acesso por linha discada. Em 1994 já existiam 3,2

milhões de computadores conectados e 3 mil web-sites. Doze meses depois, o número de computadores conectados dobrou e o número de web-sites aumentou para 25 mil. No final do ano seguinte o número de computadores sobrou novamente, e os *web-sites* se multiplicaram por 10 [GRIFFITHS, 2002].

Atualmente a *Internet* conta com um grande número de sistemas finais conectados, e se tornou essencial como meio para diferentes atividades, como serviços, transações comerciais e operações administrativas.

A *Internet* cresce cada vez mais, e com isto crescem também os seus problemas, principalmente no que diz respeito à segurança. Fraudes e roubos de informações são as atividades criminosas mais comuns que colocam em risco o uso da *Internet* para seus legítimos fins que foram descritos anteriormente.

A motivação deste trabalho é apresentar um Sistema de Detecção de Intrusão (IDS – *Intrusion Detection System*) que utiliza algoritmos imunológicos. Além da prevenção de intrusão que é objetivo do IDS e do crescimento do interesse por sistemas computacionais bio inspirados, a implementação do algoritmo imunológico no Sistema de Detecção de Intrusão é um avanço no processo para detecção e combate a invasões, e com consequente melhoria na análise de tráfego. Estes itens sempre foram importantes na pesquisa para se ter um detector de intrusão mais eficiente.

No Capítulo 2 são mostrados conceitos de segurança em redes e algumas das propriedades essenciais para uma comunicação segura. O Capítulo 3 define inicialmente tipos comuns de ataque na Internet e logo após medidas que podem ser usadas para proteger uma rede destas ameaças. O Capítulo 4 especifica conceitos de Sistemas de Detecção de Intrusão e as diferentes implementações que existem atualmente. O Capítulo 5 introduz conceitos fundamentais sobre Sistemas Imunológicos para que seja discutida a aplicação de um Sistema Imunológico Artificial em um Sistema de Detecção de Intrusão. São apresentados detalhes de um sistema deste tipo e testes desenvolvidos junto com este trabalho para testar a eficiência deste sistema sobre diversos aspectos. O Capítulo 6 finaliza a discussão e aponta vantagens, desvantagens e o que é possível ser feito num futuro próximo para que a integração entre um Sistema de Detecção de Intrusão com um Sistema Imunológico Artificial seja aprimorada.

Capítulo 2 – Segurança em redes

O objetivo da segurança em redes é dar às pessoas a liberdade de aproveitar a rede de computadores sem ter medo de comprometer seus direitos e interesses, enquanto permite a informação que é requerida acessível para os usuários que tem o legítimo direito de acesso [WANG, 2008].

No entanto segurança, pela sua natureza é inconveniente, pois quão mais robusto é o mecanismo de segurança, mais inconveniente o processo se torna. E esta robustez em certo grau pode fazer com que objetivo de segurança em redes seja comprometido [VACCA, 2009].

Apesar do problema descrito anteriormente, as ameaças a segurança dos usuários tem se tornado cada vez maior. E com o crescente aumento do uso da *Internet*, a procura por um ambiente seguro aumenta. Atualmente a *Internet* conecta mais de um bilhão e meio de pessoas [STATS, 2009].

O Centro de Estudos, Pesquisa e Tratamento de Incidentes de Segurança no Brasil (CERT.br) confirma que cada vez mais é preciso investir em segurança em redes de computadores. Até Junho de 2009 foram reportados quase 300 mil incidentes, e este número tem aumentado de ano a ano como mostra a Figura 2.1 [CERT.BR, 2009].

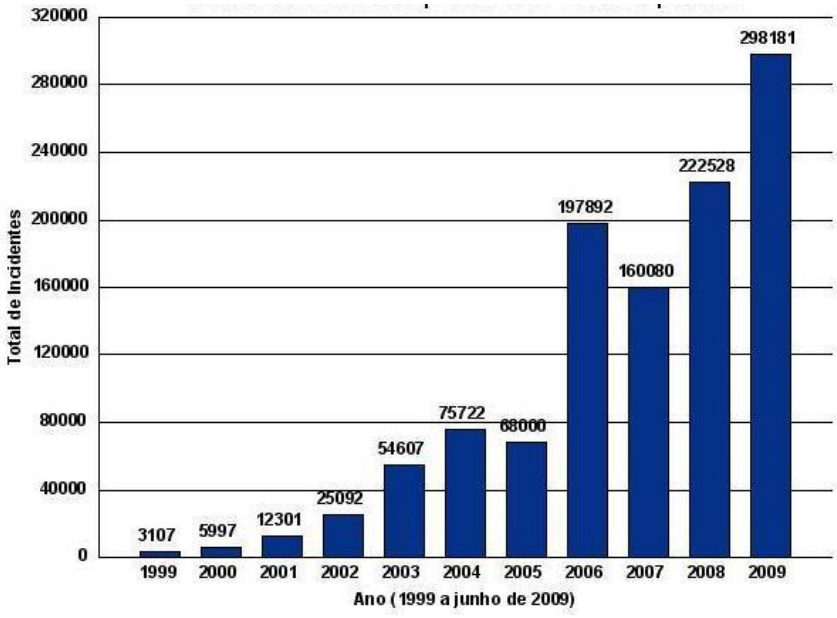

Figura 2.1 – Total de incidentes reportados ao CERT.br por ano (CERT.br, 2009)

Um dos motivos principais para o aumento de número de incidentes é de que os dados transmitidos pela *Internet* passam por equipamentos controlados por outras pessoas.

Deste modo qualquer usuário pode se tornar um atacante, alvo ou os dois ao mesmo tempo, pois mesmo que alguém não queira atacar outro na rede ele o faz, pois ele se tornou uma ferramenta de ataque.

Outro ponto importante é que com o passar do tempo os ataques são executados com mais facilidades devido ao uso de ferramentas disponíveis na *Internet*. Não é mais necessário um alto conhecimento sobre ataques para se praticar um.

Neste Capítulo serão apresentados conceitos que garantem segurança na comunicação numa rede de computadores.

2.1 – Conceitos de segurança em redes

É função vital a segurança em redes de computadores para proteger os dados que estão tanto armazenados no computador quanto os que são transmitidos pela rede. Para combater os ataques, são utilizadas políticas de segurança (um conjunto de critérios para se prover serviços de segurança [ISO 7498-2, 1989]). Segundo [KUROSE E ROSS, 2006], uma comunicação é considerada como sendo segura se possuir as seguintes propriedades:

- confidencialidade;
- integridade;
- autenticação;
- disponibilidade;
- controle de acesso.

Estes conceitos serão abordados a seguir.

2.1.1 – Confidencialidade

Confidencialidade se refere a limitar o acesso de dados privados a somente um grupo de entidades. Para se colocar em prática este aspecto, geralmente são usados métodos que identificam unicamente usuários de um dado e, como suporte, métodos de controle para identificar o nível de acesso aos recursos dentro de um sistema [PRIVACY, 2006].

Um exemplo comum é um usuário que acessa informações protegidas por meio da *Internet*. E para isto ele deverá enviar ao servidor uma senha ou outras informações que o identifique, para se acessar o que é desejado.

Para garantir a confidencialidade como descrito anteriormente é necessário que as informações que circulam pela rede sejam entendidas somente pelo transmissor e receptor.

Geralmente é utilizada a criptografia para que seja mantida a confidencialidade de uma comunicação. E nenhum método para garantir a confidencialidade pode ser considerado totalmente seguro.

2.1.2 - Integridade

A integridade dos dados é o que garante que a informação seja autêntica, completa e confiável para seu propósito. Não é somente questão de estar ou não "correto", mas questão do dado ser confiável [AVAILABILITY, 2001].

A integridade pode ser comprometida por algum erro no envio da informação ou por um usuário mal intencionado que intercepte mensagens entre transmissor e receptor para alterá-la. A segurança deste item é especialmente importante no comércio eletrônico e transações bancárias para evitar fraude nos preços e para que se evite debitar ou creditar erroneamente uma conta.

2.1.3 - Autenticação

Segundo [KUROSE E ROSS, 2006], autenticação é a propriedade de uma comunicação que assegura que o remetente é realmente a entidade que enviou a mensagem.

É um processo aonde se busca verificar a identidade digital do usuário do sistema.

A legitimidade do remetente é importante em transações na *Internet*, pois é preciso ter garantia de com quem se fala. Por exemplo, numa transação bancária o cliente deve ter certeza de que ele está trocando informações com o banco.

A falta desta propriedade leva a fraudes e golpes via *Internet*.

2.1.4 - Disponibilidade

É um objetivo crítico, pois em alguns sistemas é fundamental a resposta em tempo hábil.

A disponibilidade consiste na proteção dos serviços prestados pelo sistema de forma que eles não sejam degradados ou se tornem indisponíveis sem autorização, assegurando ao usuário o acesso aos dados sempre que deles precisar. Isto pode ser chamado também de continuidade dos serviços [SOUSA, 2009].

É importante que se evite os ataques de negação de serviço, que é um ataque no qual há tentativa em tornar os recursos de um sistema indisponíveis para seus utilizadores

para que o sistema esteja sempre disponível. Do contrário, podem ser geradas perdas tão graves quanto às causadas pela remoção das informações daquele sistema.

2.1.5 – Controle de acesso

O controle de acesso garante que somente pessoas autorizadas tenham o direito ao acesso às informações. É geralmente composto por processos de autenticação, autorização e auditoria. O controle de acesso pode ser entendido como a habilidade de permitir ou negar a utilização de um recurso.

2.1.6 – Não repúdio

Além destes aspectos citados por [KUROSE E ROSS, 2006], é importante ressaltar o não repúdio, que significa que a pessoa que detém o dado não tem como convencer outra pessoa que ela não o detém. Esta propriedade oferece provas irrefutáveis de que um usuário realizou alguma ação.

2.2 – Mecanismos de segurança

A seguir são apresentados alguns mecanismos para que se cumpram as propriedades de segurança de rede descritas anteriormente.

2.2.1 – Definições

De acordo com [ISO 7498-2, 1989], os mecanismos de segurança são divididos em dois tipos:

- mecanismos de segurança específicos (os que provém serviços de segurança fixos);
- mecanismos de segurança pervasivos (os que provém serviços de segurança individuais).

2.2.2 – Mecanismos de segurança específicos

São mecanismos de cifragem como mecanismos de assinaturas digitais, de controle de acesso, de integridade de dados, de autenticação, controle de rotas e validação [DENT E MITCHELL, 2004].

Os mecanismos de cifragem são mecanismos de segurança que envolvem a transformação dos dados em algum tipo de dados que não há modo de se ler [TAYLOR ET AL., 1996].

As transformações podem ser reversíveis ou irreversíveis. Um exemplo de transformação irreversível são as funções *hash* de criptografia.

As técnicas mais utilizadas são a criptografia de chaves simétricas e assimétricas, que serão abordadas com detalhes neste Capítulo.

2.2.3 – Mecanismos de segurança pervasivos

De acordo com [OPPLIGER, 2001], alguns dos mecanismos de segurança pervasivos podem ser vistos como aspectos da administração da rede. São enumerados cinco mecanismos de segurança pervasivos:

- conceito geral de funcionalidade confiável: pode ser usado para estabelecer a efetividade de outros mecanismos de segurança. Qualquer funcionalidade que provê diretamente, ou provê acesso a mecanismos de segurança devem ser confiáveis;
- recursos do sistema devem indicar o nível de segurança requerido. Muita das vezes é necessário indicar este nível enquanto este recurso está em trânsito na rede. Este indicador pode ser um dado adicional associado ao recurso transferido ou pode ser implícito pelo contexto, como por exemplo, do caminho que faz o recurso até chegar ao destino;
- a detecção de eventos pode ser usada para detectar aparentes violações de segurança;
- a auditoria de segurança se refere a um exame independente de registros e atividades para garantir que as políticas de segurança estabelecidas está em conformidade com os procedimentos operacionais;
- a "recuperação de segurança" diz respeito a mecanismos de manipulação de erros e funções de administração e tem ações de correção quando se aplicam um conjunto de regras.

2.2.4 – Criptografia

A criptografia consiste em um conjunto de práticas para codificar uma mensagem de modo que somente o transmissor e o receptor consigam decodificá-la para entender qual é o conteúdo da mensagem [WANG, 2008].

É uma técnica muito utilizada atualmente, pois diz respeito ao importante aspecto da confidencialidade, propriedade já discutida anteriormente. A seguir serão mostrados conceitos e algoritmos utilizados para este fim.

2.2.5 – Sistema de chave simétrica

Os algoritmos de chave simétrica são uma classe de algoritmos para criptografia que usam chaves relacionadas para as operações de codificação/decodificação (ou cifragem/decifragem) de uma mensagem [WANG, 2008].

A chave de cifragem pode ser idêntica à de decifragem ou poderá existir uma transformação simples entre as duas chaves. Os algoritmos DES, 3DES e AES utilizam este tipo de sistema.

2.2.5.1 - DES

O *Data Encryption Standard* (Padrão para Criptografia de Dados) foi publicado nos Estados Unidos em 1977 para uso comercial e não classificado do governo americano. É utilizada uma chave de 56 bits, que era suficiente para resistir a ataques de força bruta nos anos 70 e 80 [WANG. 2008].

Sua encriptação e decriptação são simétricas e com quatro operações básicas: ou exclusivo, permutação, substituição e troca circular. O DES codifica a mensagem em porções de 64 bits usando uma chave de 56 bits, sendo que oito bits são bits de paridade [DENT E MITCHELL, 2004].

2.2.5.2 - 3DES

O 3DES (Padrão para Criptografia de Dados Triplo) é baseado no DES, com a única diferença de aplicar-se 3 chaves de 48 bits para encriptação. É mais lento e oferece maior segurança que o DES.

2.2.5.3 – AES

O AES (*Advanced Encryption Standard* – Padrão Avançado de Criptografia) faz a cifragem em blocos e encripta com chaves de 128, 192 ou 256 bits. Deste modo a encriptação é muito mais segura do que a encriptação DES. Para comparação o DES tem chaves de 56 bits, o que nos dá aproximadamente $7{,}2 \times 10^{16}$ chaves diferentes. No padrão de 128 bits do AES há 10^{21} combinações a mais de chaves diferentes [WANG, 2008].

2.2.6 – Sistema de chaves assimétricas

Com o sistema de chaves assimétricas é um sistema aonde cada entidade possui duas chaves denominadas chave privada e chave pública [WANG, 2008].

A chave privada é mantida secreta, enquanto a pública pode ser distribuída. Mensagens são encriptadas com a chave pública e só podem ser decriptadas com a chave privada correspondente. Estas duas chaves são matematicamente relacionadas, e a chave privada não pode ser derivada da chave pública.

Estes esquemas de encriptação são provadamente seguros, pois a maioria deles são baseados na dificuldade de fatoração do produto de dois grandes números primos, que é a base do funcionamento da distribuição de chave. Esta segurança tem um significado matemático preciso.

Para que se estabeleça uma comunicação segura com o sistema de chaves assimétricas o transmissor obtém a chave pública do receptor para criptografar sua mensagem e envia a mensagem, e o receptor recebe e consegue descriptografar a mensagem usando a chave privada. Este processo é ilustrado pela Figura 2.2

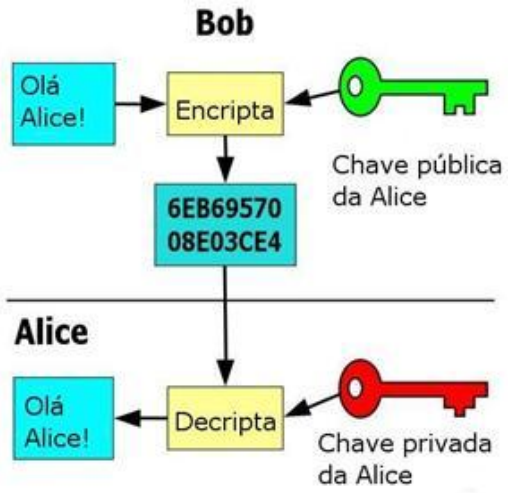

Figura 2.2 – Encriptação e decriptação utilizando o sistema de chaves assimétricas

A seguir são apresentados alguns exemplos de uso de chaves simétricas [WANG, 2008]:

- algoritmo de encriptação *ElGamal*;
- técnicas de desenhos elípticos;
- algoritmo de encriptação RSA;
- PGP (*Pretty Good Privacy* – Privacidade muito boa). É um programa de computador que prove privacidade ciptográfica e autenticação;

- GPG *(GNU Privacy Guard* – Guarda de privacidade GNU), implementação do OpenPGP;
- SSL (*Secure Socket Layer* - Camada de *Sockets* Segura), um protocolo de comunicação seguro;
- SSH (*Secure Shell – Shell* Seguro), um programa de computador e um protocolo de rede que permite a conexão com outro computador na rede, de forma a executar comandos de uma unidade remota.

No Capítulo 3 são apresentados os principais ataques aplicados na *Internet* e algumas medidas de defesa para a proteção contra essas ameaças.

Capítulo 3 – Ataques e defesa

Foi discutido anteriormente que a quantidade de incidentes de segurança tem aumentando nos últimos anos. Para o combate efetivo destas ameaças é necessário conhecer como o ataque é desencadeado, bem como seus principais motivos.

Neste Capítulo são abordadas técnicas de ataque e defesa em redes de computadores.

3.1 – Motivação

A adoção de uma boa política de segurança é fundamental para a proteção não somente contra ataques que vem de fora da rede local (ataques externos), mas também ocorrem ataques de dentro da rede, conhecidos como ataques internos. São feitos por funcionários que desejam atacar a sua própria rede por insatisfação ou sabotagem. Estes ataques são conhecidos como ataques diretos. Os ataques indiretos acontecem quando o usuário da rede não tem idéia de que sua ação pode prejudicar a rede, e esta ação pode tornar o ataque externo possível pela instalação de *trojans* [WANG, 2008].

Os motivos dos atacantes são vários, podendo ir de simples diversão até o roubo de informações confidenciais envolvendo desde perda de privacidade até crimes contra o Estado. E seja qual for motivo, é necessário ter métodos de defesa para a diminuição de ocorrências de ataque, e também para a minimização dos danos e rápida recuperação no caso do ataque que ocorreu com sucesso. Deste modo, é necessário proteger tanto os dados que estão sendo transmitidos na rede quanto os que estão armazenados nela.

Na subseção seguinte são mostrados os tipos mais comuns de ataque.

3.2 – Ataques

Quase todos os ataques conhecidos são construídos de alguma forma básica que são descritas a seguir ou uma combinação destes. Quanto mais elaborada a combinação, mais perigoso o ataque se torna [WANG, 2008].

Tradicionalmente os métodos de ataque diretos seguem passos como mostrado na Figura 3.1.

Figura 3.1 – Passos tradicionais para um ataque direto a uma rede[WANG, 2008]

No primeiro passo o atacante tenta aprender o máximo possível das vulnerabilidades da vítima. Uma vulnerabilidade pode ser definida como uma interseção de suscetibilidade ou falha do sistema, acesso a falha e a capacidade da falha ser explorada [TENETS, 2009].

São feitos *pings* e *traceroutes* para se mapear a rede a ser atacada, e um *port-scanner* pode revelar portas abertas e até mesmo o Sistema Operacional do alvo a ser atacado.

Feito isto o atacante entra no segundo passo, tentando comprometer o alvo de uma ou mais maneiras como, por exemplo, ataques com *worms* e/ou de força bruta.

O terceiro passo consiste em cobrir as evidências e estabelecimento de controle. Com isto o atacante pode instalar *rootkits* que são desenvolvidos para esconder a existência de alguns processos ou programas de métodos normais de detecção [HOGLUND E BUTLER, 2005].

Podem ocorrer outros tipos de ataques mais elaborados, como o de escrever um código para se aproveitar de uma vulnerabilidade específica [MCCLURE ET AL., 2005]. Este ataque é conhecido como *buffer overflow* (estouro de pilha) e historicamente é o tipo de vulnerabilidade mais explorada [FOSTER ET AL., 2005].

De acordo com [CERT, 2006], os métodos a seguir são os mais comuns utilizados por intrusos.

3.2.1 – Vírus

São programas maliciosos de computador que podem copiar a si mesmos, tal qual como um vírus biológico e infectar um computador. Devem ser executados no hospedeiro

para entrar em funcionamento, e a maioria dos vírus exploram as vulnerabilidades de um sistema. Os vírus podem se espalhar para outros computadores infectando arquivos numa rede ou sistema de arquivos que é acessado por outro computador, além de poderem ser obtidos pela *Internet*, dispositivo de armazenamento secundário ou instalação de programas de procedência desconhecida [HOGLUND E BUTLER, 2005].

3.2.2 – *Backdoor*

É uma aplicação maliciosa utilizada para administração remota. Faz uso de métodos para burlar uma autenticação normal.

Estes *softwares* podem permitir um atacante obter controle do computador atacado e usá-lo como ele desejar, sempre com objetivo de se manter indetectável. O uso desta ferramenta pode acarretar grandes prejuízos para a vítima, como por exemplo, a perda de seus dados confidenciais.

3.2.3 – *Bots* e *Botnets*

O *bot* é um código malicioso que uma vez que infecta o computador do usuário, permite que um atacante o controle remotamente. Estas instruções são geralmente utilizadas para se iniciar alguma atividade maliciosa, como por exemplo, ataques distribuídos de negação de serviço e envio de e-mails fraudulentos.

Geralmente o atacante usa um servidor IRC (*Internet Relay Chat*) para enviar comandos remotamente para o *bot*, que é programado para entrar em um canal do servidor e aguardar instruções do que ele deverá fazer.

Uma *botnet* consiste de um conjunto de computadores infectados com *bot*, conectados a *Internet* e comandados por um atacante.

3.2.4 – Cavalos de Tróia

Cavalos de Tróia (*Trojan Horse*) são aplicações benignas que contém em seu interior código malicioso, que é usado para instalar outras aplicações que podem abrir portas para futuras invasões. Os meios de propagação do cavalo de tróia são os mesmos dos vírus.

De acordo com [BITDEFENDER, 2009], os *trojans* estão se tornando uma forma de ataque cada vez mais comum. De janeiro a junho de 2009 foi feita uma pesquisa mostrando que 83% do total de detecções são referentes a este *malware*.

3.2.5 – *Keyloggers*

Programa de computador desenvolvido para capturar teclas digitadas pelo usuário do computador (*keystroke logging*). Atualmente a maioria dos *keyloggers* podem também capturar telas da vítima (*screen logging*).

Na maioria dos casos as telas são salvas de acordo com uma ação do usuário, como a de clicar no teclado virtual no *website* de um banco.

Estes softwares maliciosos são utilizados principalmente para roubo de senhas. As telas e teclas digitadas são enviadas para o atacante de tempos em tempos, geralmente por um *backdoor*.

A vítima pode o obter através da *Internet* ou de um cavalo de tróia.

3.2.6 – *Man-in-the-Middle*

O ataque *Man-in-the-Middle* (homem no meio) é uma forma de ataque na qual o atacante faz conexão com as vítimas e lê, insere e modifica o que ele deseja na comunicação entre as vítimas sem que nenhuma delas perceba, ou seja, as vitimas acreditam que estão se comunicando diretamente entre elas.

A seguir é exemplificado este tipo de ataque. A e B são duas vítimas que tentam trocar mensagens seguras através de uma criptografia de chave pública.

```
A envia mensagem a B pedindo por sua chave pública, que é
interceptada por C

C e entrega esta mensagem B (ele não tem como provar que a mensagem
não é de A)

C substitui a chave de B com a sua, e isto faz A acreditar que C é
B

A encripta a mensagem para que somente B leia, mas C também pode
ler

C pode modificar a mensagem do modo que desejar, re-encriptar e
mandar para B em nome de A
```

Figura 3.2 – O ataque *Man-in-the-Middle* [WANG, 2008]

3.2.7 – Negação de serviço

Segundo [KUROSE E ROSS, 2006], a negação de serviço (*Denial of Service* – DoS) é um ataque caracterizado por uma tentativa de fazer com que um recurso se torne indisponível na rede para seus usuários legítimos, por esforço concentrado de uma pessoa.

O ataque é geralmente feito com muitas requisições para comunicação com o alvo, o que faz com que o ele reserve memória para servir o atacante, julgando a requisição legítima. Deste modo o alvo é saturado e não responde mais as requisições legítimas.

3.2.8 – Negação de serviço distribuída

O ataque de negação de serviço distribuído (*Distributed Denial of Service* - DDoS) tem o mesmo conceito básico do ataque DoS, com a diferença de ter o esforço de centenas ou até milhares de computadores escravizados. A escravização é iniciada com uma varredura para se encontrar computadores vulneráveis. Quando este é achado, o atacante instala uma ferramenta para que o escravo seja controlado remotamente. Então os escravos aguardam a instrução para iniciar o ataque que terá poder muito maior do que um DoS.

3.2.9 – *Phishing*

Phishing é um processo fraudulento para se adquirir nomes de usuário, senhas e números de cartão de crédito em que o atacante se passa por uma entidade confiável. É feito na maioria das vezes por email ou mensageiro instantâneo, com uso de mensagens que chamam a atenção do usuário [KOON, 2006].

Exemplos de *phishings*: páginas que se parecem com a de um banco que pedem ao usuário para digitar seus dados bancários ou emails como o famoso golpe da Nigéria, aonde o usuário era convidado a fazer parte de uma transferência internacional de fundos. E para isto ele deveria contribuir antecipadamente uma quantia para no fim receber parte do dinheiro da transferência [INFORMATION, 2004].

3.2.10 – *Rootkits*

Composto por diversos programas maliciosos com a função de ocultar seu código para que não mostre que o sistema foi comprometido. Geralmente fazem interceptação nas APIs para que não sejam detectados. Uma função bastante explorada dos *rootkits* é enganar o usuário para que rodem aplicações não seguras em seus computadores, como por exemplo,

backdoors. Para isto é usado o monitoramento de processos, escondendo arquivos os dados do Sistema Operacional [BRUMLEY, 1999].

Além destas funções, o *rootkit* permite que o atacante remova evidências em arquivos de *log* e assumir outras funções como a de um *keylogger*.

3.3 – Defesa

Diante a variedade de ataques existentes, é necessário se precaver destes ataques. A adoção destas medidas defensivas torna o ambiente de rede mais seguro, mas não o torna invulnerável.

Os métodos mais utilizados para a defesa de redes são discutidos a seguir. Estes são os métodos mais comuns, e assim como nos ataques, são utilizados individualmente ou em conjunto com objetivo de uma melhor proteção para a rede.

3.3.1 – *Firewall*

É um software comum para manter a comunicação de uma rede segura contra intrusos, pois bloqueia tráfego de acordo com a necessidade em uma rede privada. É também capaz de alertar sobre tráfego suspeito baseado num conjunto de regras.

Pode também existir na forma de hardware ou combinação entre software e hardware. Os *firewalls* podem ser divididos em [VACCA, 2009]:

- filtros de pacotes: fazem análise individual dos pacotes à medida que são transmitidos. Baseiam-se em regras para aceite ou descarte do fluxo transmitido;
- gateways de aplicação: Tratam as requisições como se fosse uma aplicação. As respostas sempre são analisadas antes de serem entregues ao solicitante;
- firewall de aplicação: Este tipo de firewall analisa particularidades de cada protocolo para tomar decisões com mais eficiência do que um simples filtro de pacotes. Deste modo é possível analisar hipertexto encriptado além de mapear transações com padrão específico na rede;
- firewall de estado de sessão: Guarda o estado das transações para evitar tráfegos ilegítimos. Algumas versões usam uma tabela de conexões legítimas, deste modo novas regras podem ser adicionadas mais facilmente.

3.3.2 – Antivirus

Software que tenta identificar e eliminar softwares maliciosos. Identifica arquivos maliciosos checando sua estrutura e comparando com seu banco de ameaças conhecidas e identifica ações suspeitas de um computador infectado. É importante manter atualizado este banco de ameaças conhecidas, pois novos vírus surgem a cada dia [VACCA, 2009].

Atualmente é a forma mais usada para combater códigos maliciosos que já estão executando na vítima. Pode ser obtido gratuitamente na *Internet*.

3.3.2 – NAT

O NAT (*Network Address Translation* – Tradução de Endereços de Rede) é uma técnica de reescrita de endereço IP de pacotes que passam por um roteador ou *firewall*. Originalmente surgiu como forma dos computadores de uma rede privada receberem resposta a pedidos feitos fora da rede, mas também tem a utilidade de esconder os computadores de ataques vindo de fora da rede privada.

3.3.3 – *Proxy*

Trata-se de um servidor que atende a requisições numa rede e repassando para frente. O cliente se conecta ao *proxy* para requisitar algum serviço por meio dele.

Os usos mais comuns do *proxy* são armazenar *cache*, ou seja, quando um cliente faz requisições HTTP por meio dele, a resposta do servidor *web* pode ser salva para futuras consultas, evitando-se o consumo de banda de *Internet*. Além de poder bloquear determinados acessos dos clientes que necessitam de uma conexão com *proxy* para acessar a *Internet*.

No ponto de vista de segurança o *proxy* ajuda a prevenir ataque, pois os pedidos são feitos de forma indireta, além do administrador da rede poder criar mais regras de segurança que determinam como o pacote é tratado pelo servidor.

3.4 – Políticas de segurança

Para que ataques sejam evitados é necessário que se implemente uma política de segurança na rede. Isto é feito definindo-se responsabilidades e direitos para cada usuário da rede. É um instrumento importante para proteger uma organização contra ameaças à segurança da informação que ela pertence [CERT.BR , 2003].

Os objetivos de uma política de segurança variam para cada rede. A instituição deve ficar a cargo de colocar regras que reflitam a realidade da segurança na instituição.

A política deve especificar os mecanismos através dos quais requisitos pré-determinados. Com isso o administrador da rede pode identificar e tomar decisões para combater o não respeito a regras que já foram estabelecidas. O próximo Capítulo mais um mecanismo de segurança, o detector de intrusões.

3.5 – Conclusão

Neste Capítulo foram vistos diversos mecanismos preventivos para proteger uma rede de ataques externos e internos. O próximo Capítulo mais um mecanismo de segurança, o detector de intrusões.

Capítulo 4 – Sistemas de Detecção de Intrusão

Na grande maioria dos casos é necessário para proteger a rede de atividades de intrusão, pois como já foi dito nos Capítulos anteriores a quantidade e complexidade de ataques em redes tem crescido a cada ano.

Para que ataques sejam combatidos já existem diversas ferramentas que controlam a rede. Mas quase todas estas medidas são preventivas, e não são suficientes para que se combata ou evite o comprometimento de uma rede antes que ocorra um ataque.

Para isso é mais comum que se utilize um Sistema de Detecção de Intrusão (*Intrusion Detection System* - IDS), que escuta todas as atividades em uma rede. Um IDS faz política de tráfego identificando pacotes que são predefinidos como um padrão não usual de atividade com objetivo de detectar tentativas de invasão ou se já existe alguma em andamento [VACCA, 2009].

4.1 – Funcionamento

O funcionamento básico de um detector de intrusão é salvar informações sobre eventos numa rede e analisá-los usando métodos pré-definidos. A detecção de intrusão pode ser feita tanto em nível de rede, que é denominada por NBD (*network-based detection* detecção baseada em rede) quanto dentro de alguma estação na rede, conhecida como HBD (*host-based detection* – detecção baseada em *host*).

Uma analogia a detecção de intrusão é um sistema de alarme contra ladrões, pois este sistema também é uma tentativa de monitorar fluxos para detectar intrusos e avisar ao responsável o acontecimento.

O detector de intrusão é configurado para observar pacotes na rede, e alguns destes sistemas podem reagir a invasões. É possível também armazenar informações sobre todo tráfego da rede para, por exemplo, uma análise de um ataque que não foi combatido ou para que seja descoberta a origem de um ataque.

A seguir são mostrados com detalhes o funcionamento de cada tipo de detector de intrusão e algumas características de suas regras.

A Figura 4.1 ilustra o funcionamento geral de alguns detectores de intrusão.

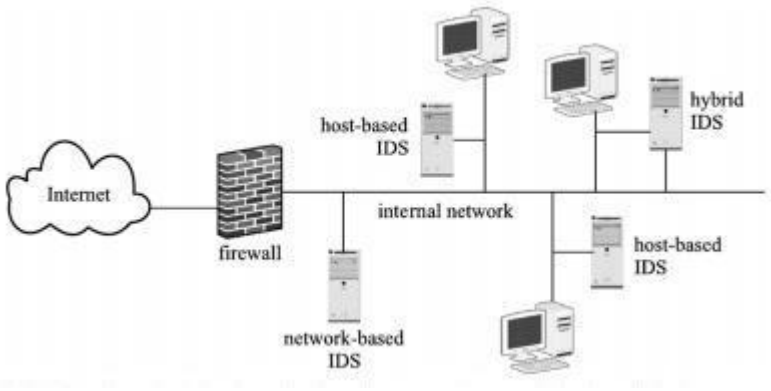

Figura 4.1 – Esquema de um sistema de detecção de intrusão e firewall [WANG, 2008]

Um detector de intrusão pode também agir ativamente. Conforme [ABSOLUTA, 1999], o detector de intrusão pode aplicar algumas medidas de defesa quando identifica um ataque:

- reconfiguração de firewalls e roteadores;
- ativação de alertas nas estações de gerência via mensagens SNMP (*Simple Network Management Protocol* – Protocolo Simples de Gerenciamento de Rede);
- encerramento da conexão.

4.2 – Histórico

A Dra. Dorothy E. Denning do *Stanford Research Institute* (Instituto de Pesquisa Stanford) começou a trabalhar em 1983 para o governo dos Estados Unidos em um projeto de detecção de intrusão para mainframes do governo. Denning e Peter Neuman publicaram em 1986 um modelo de detecção de intrusão que ainda hoje é base de muitos sistemas [DENNING, 1986]. Este era usado principalmente para obter estatísticas para uso anormal da rede, executado em estações Sun. Até 1988 a detecção era muito difícil, pois era necessário procurar pela informação em grandes quantidades de dados. Então surgiu o projeto *Haystack* (palheiro). Foi dado este nome, pois um membro do projeto dizia que "procurar uma utilização inadequada do sistema era como procurar uma agulha no palheiro". O projeto produziu um IDS que fazia comparações de atividades com padrões pré-estabelecidos, tornando muito mais fácil a análise

Em 1988 foi criado o MIDAS (*Multics Intrusion Detection and Alerting System*) baseado no trabalho de Denning e Neuman.

Em 1990 a *Time-based inductive machine* detectava anomalias na rede. Era executado no VAX 3500, e no mesmo ano foi lançado um protótipo inovador que incluía estatísticas e um sistema avançado de detecção. Mas o uso de estatísticas para auditoria foi colocado em prática somente pela AT&T com o *Computer Watch* [TENG ET AL., 1990].

Em 1991, a Universidade da Califórnia criou um protótipo de IDS distribuído. No mesmo ano foi criado em *Los Alamos National Laboratory's Integrated Computing Network* (ICN) que usava estatísticas de anomalias na rede. No início dos anos 90 a *Haystack Labs* foi a primeira vendedora comercial destes sistemas. Somente em 1997 o uso comercial deste tipo de software ganhou popularidade [LAUFER, 2003].

Mudanças significativas só ocorreram em 2001 com o ADAM – *Audit data analysis and mining* que usou o *tcpdump* para construir perfis de regras para classificação.

4.3 – Políticas de detecção

Na prática, o uso de um IDS em ambiente comercial são combinações de IDS's de rede, *host* e/ou aplicação. A classificação pela funcionalidade dos IDS's pode ser feita em três categorias principais [KOHLENBERG ET AL., 2007]:

- NIDS: *Network-based Intrusion Detection System* (Sistema de detecção de intrusão baseado em rede);
- HIDS: *Host-based Intrusion Detection System* (Sistema de detecção de intrusão baseado em *host*);
- DIDS - *Distributed Intrusion Detection System* (Sistema de detecção de intrusão distribuído).

4.4 – IDS baseado em rede

Um NIDS funciona monitorando um segmento completo de rede. No computador onde o NIDS está instalado, a interface de rede é configurada para funcionar em modo promíscuo, isto é, a interface não capta somente os pacotes destinados a ela, e sim todos os pacotes que estão trafegando na rede. Uma vantagem direta do NIDS é que ele não interfere de modo algum no tráfego. O trajeto original do pacote é mantido, além de não ser detectável por atacantes.

É possível aplicar regras com visão mais ampla do que num IDS baseado em *host*, que será definido adiante, além do reconhecimento de padrões que podem inibir um ataque. Para aprimorar a detecção é possível usar sensores espalhados na rede.

Uma desvantagem é que quando o tráfego é muito alto o NIDS pode não conseguir capturar todos os pacotes ou ficar saturado. A utilização de *switches* na rede podem impedir ou restringir os sensores de obter todos os pacotes, pois enviam pacotes de acordo com os endereços de interface diretamente. A figura 4.2 ilustra um IDS baseado em rede.

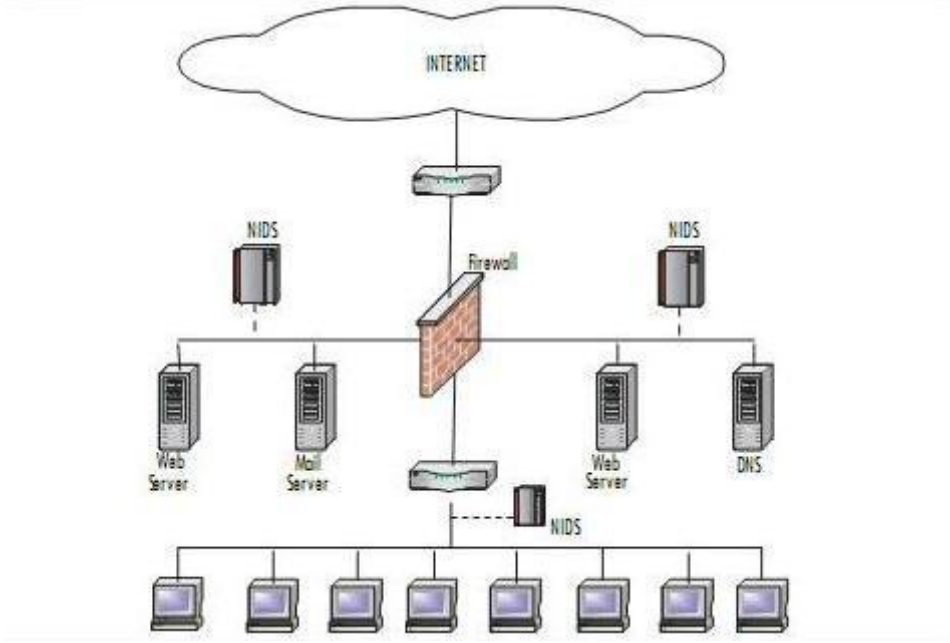

Figura 4.2 – IDS baseado em rede [BEALE, 2004]

Outra desvantagem é que quando é usada a criptografia nos pacotes torna-se difícil a análise para determinar se este pacote faz parte de um ataque.

4.5 – IDS baseado em *host*

Como mostrado na Figura 4.3, o IDS baseado em host (HIDS) é um detector de intrusão que protege o sistema aonde ele reside. Isto ajuda a proteger os eventos locais, como alterações no sistema de arquivos. Eles podem também ter regras bem específicas para um host em particular e detectar atividades maliciosas que usam criptografia.

A maior desvantagem de um HIDS é de poder ser desativado. O gerenciamento pode se tornar complicado dependendo do número de computadores na rede e pode influenciar o desempenho do computador no qual está instalado, pois consome recursos.

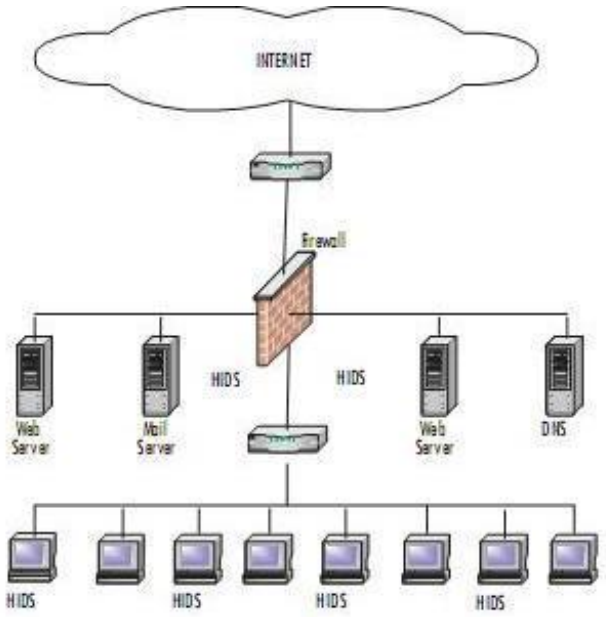

Figura 4.3 – IDS baseado em *host* [BEALE, 2004]

4.6 – IDS distribuído

O IDS distribuído (DIDS) é a combinação dos dois sistemas de detecção de intrusão citados anteriormente, com a existência de um servidor para que os sensores remotos reportem qualquer comportamento estranho.

A complexidade e as consequentes vantagens e desvantagens são dependentes de como o DIDS é codificado, pois os sensores podem agir como NIDS, HIDS ou ambos.

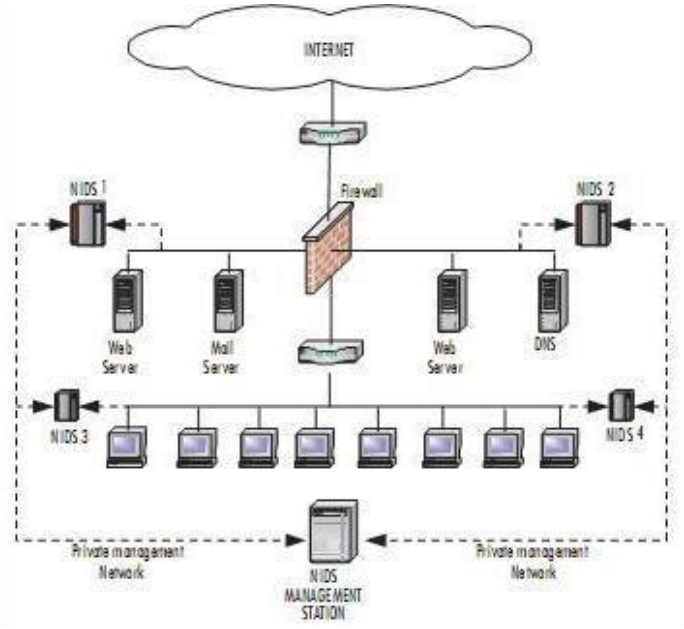

Figura 4.3 – IDS distribuído [BEALE, 2004]

4.7 – Identificando ataques

O detector de intrusão deve possuir um algoritmo eficiente para dar um bom suporte a decisão do administrador, além de estar configurado para se adaptar com facilidade ao comportamento da rede.

Após a instalação do IDS se dá início ao processo adaptativo do detector. Este processo é caracterizado do aprendizado pela distinção de pacotes considerados corretamente bons ou ruins.

Os pacotes que são marcados erroneamente são divididos em falsos positivos e falsos negativos, que serão definidos a seguir.

4.7.1 – Falso positivo

O falso positivo (ou erro do tipo 1) é descrito como o erro de rejeitar uma hipótese nula (hipótese considerada verdadeira até que seja provado o contrário). É quando se espera uma verdade que não aconteceu, e pode ser visto como excesso de credulidade [NEYMAN E PEARSON, 1928].

No caso específico do IDS um falso positivo ocorre quando pacotes do tráfego são marcados como ataque quando na verdade são atividades consideradas normais na rede.

A presença de falsos positivos pode ser diminuída através do ajuste de regras existentes e desabilitando os serviços não oferecidos na rede. Este erro acarreta em mais tráfego para análise, logo este problema não é tão grande quanto a presença de falsos negativos.

4.7.2 – Falso negativo

Também conhecido como erro do tipo 2, é o erro da falha ao rejeitar uma hipótese nula quando ela é verdadeira. É quando se espera uma verdade aonde nunca existiu, e pode ser visto como excesso de ceticismo [NEYMAN E PEARSON, 1928].

No caso do IDS é quando há falhas na detecção de comportamentos de ataque, ou seja, o IDS vê como tráfego normal uma tentativa de invasão à rede.

O falso negativo ocorre por desconhecimento do tipo do ataque, por estar sobrecarregado ou mal configurado.

O erro tipo 2 é um erro considerado mais grave do que o erro tipo 1 num IDS, pois a presença deste erro indica que o detector tratou uma ameaça como tráfego benigno, logo perdeu um possível ataque.

4.8 – Detecção por assinaturas

Também conhecido como detecção operacional, é responsável por inspecionar eventos correntes e decidir se estes tem comportamento aceitável. Segundo [WANG, 2008], um conjunto de regras de comportamentos pode ser especificado como:

- arquivos de sistema, particularmente arquivos de senhas não devem ser copiados pelos usuários;
- discos devem ser acessados somente pelos utilitários do Sistema Operacional, usuários não o devem fazer;
- usuários não devem ter acesso a diretórios de outros usuários;
- usuários não devem modificar arquivos de outros usuários;
- usuários não devem continuar tentando se autenticar após três falhas;
- usuários com maior nível de autorização não devem copiar arquivos de diretórios com menor nível de acesso para um diretório com maior nível de acesso;
- usuários com níveis de autorização mais baixos não devem ler arquivos em diretórios com menor nível de acesso.

4.9 – Assinaturas de rede

Assinaturas são definidas como características de um pacote que permitem identificá-lo, mostrando comportamento de pacotes que podem afetar a execução normal do sistema. Consistem de assinaturas de cabeçalho (*header signatures*) e assinaturas de conteúdo (*content signatures*) [WANG, 2008].

As assinaturas de conteúdo, como o nome indica, verificam o que há dentro de um pacote e determina se este é aceitável ou não. A checagem de cabeçalho comumente identifica pacotes maliciosos, como o de ataques em *broadcast*. Além disso, o IDS pode verificar mais algumas características para se detectar ataques, como portas de origem e destino, número de sequência dos pacotes e sequência ilegal de *flags* TCP.

O exemplo a seguir mostra um intruso (computador 1) que tenta usar o FTP para baixar o arquivo `/etc/passwd` de um computador remoto (computador 2).

```
Computador 1 -> Computador 2 ETHER TYPE=0800 (IP), SIZE = 68 bytes

Computador 1 -> Computador 2 IP D=129.63.8.1 S=129.63.8.12 LEN=54,
ID=4434U

Computador 1 -> Computador 2 TCP D=21 S=28613 ACK=2132480783
SEQ=1358787809 LEN=14 WIN=61329+

Computador 1 -> Computador 2 FTP C PORT=28113 SITE exec
cat+\verb+/etc/passwd\r\n
```

Figura 4.5 – Tentativa de intrusão usando FTP [WANG, 2008]

4.10 – Detectando anomalias

Outro modo de detecção é o baseado em anomalias, ou seja, o detector procura por intrusões e mau uso monitorando a atividade da rede. É fundamental neste tipo de detecção o IDS conhecer a priori o que é uma atividade normal. Em geral é feito algum tipo de processo adaptativo para atingir este objetivo. Para que o administrador reconheça anomalias é necessário o uso de estatística, como por exemplo, o número de vezes que um evento ocorre num período de tempo [WANG, 2008].

Baseado neste e em outros parâmetros pode-se medir quantidade de eventos e utilização dos recursos de hardware.

4.11 – Conclusão

A detecção de intrusão tem espaço ativo na pesquisa. A necessidade de técnicas para melhor detecção e análise de tráfego vem crescendo a cada dia.

No próximo Capítulo é mostrada uma técnica que é uma tentativa para uma detecção de intrusão mais completa.

Capítulo 5 – Detecção de intrusão com utilização de sistemas imunológicos artificiais

No Capítulo 4 foram mostradas várias técnicas para que pacotes considerados maliciosos sejam detectados. Com isto é possível tomar ações combativas contra atividades prejudiciais a uma rede. No entanto, estas técnicas não prevêem adaptações de ataques já conhecidos. De acordo com [LIU, 2009] a construção de um sistema de detecção de intrusão com adaptabilidade e robustez é uma necessidade.

Antes de apresentar o sistema de detecção de intrusão que utiliza idéias do sistema imunológico (SI), é necessário definir alguns conceitos básicos do funcionamento de sistemas imunológicos bem como a sua modelagem.

5.1 – Conceitos de Sistemas Imunológicos

Todas as plantas e animais possuem um sistema imunológico que os confere a capacidade de resistir a agentes que causam doenças, conhecidos como patógenos. Basicamente o sistema imunológico é uma rede de células que trabalham em conjunto para defender o organismo identificando e matando patógenos. Ele é complexo e pode detectar uma gama muito grande de agentes patogênicos, desde vírus a parasitas [LITMAN, 2005].

O sucesso da identificação de patógenos deve-se a habilidade do sistema imunológico de reconhecer o que é parte do corpo (próprio) e células que não são do corpo (não próprio).

Quando o sistema imunológico identifica algum conjunto de células não próprias, acontece uma resposta imune, conhecida como antígeno.

5.1.1 – Células Imunes

O sistema imunológico é composto por vários tipos de células e cada um destes tipos faz um trabalho específico no mecanismo de resposta imune. São armazenadas algumas células para reconhecer milhões de patógenos possíveis. Quando este patógeno aparece, estas células se multiplicam e se comunicam com citoquinas, que são substâncias secretadas que carregam sinais entre células. Estes sinais fazem com que estas células se tornem células específicas do sistema imunológico. A seguir são descritos os vários tipos de células específicas [MEDICINENET, 1999].

5.1.2 – Células B

As células B são linfócitos (um tipo de célula branca do sangue) que tem como principal função produzir anticorpos para combater antígenos. Os anticorpos são os identificadores de corpos não próprios. A estrutura dos anticorpos é similar umas as outras, com exceção da chamada "região hipervariável". Deste modo as variações desta região hipervariável se encaixam com diferentes tipos de antígeno, como mostra a Figura 5.1 [HEALTH, 2003].

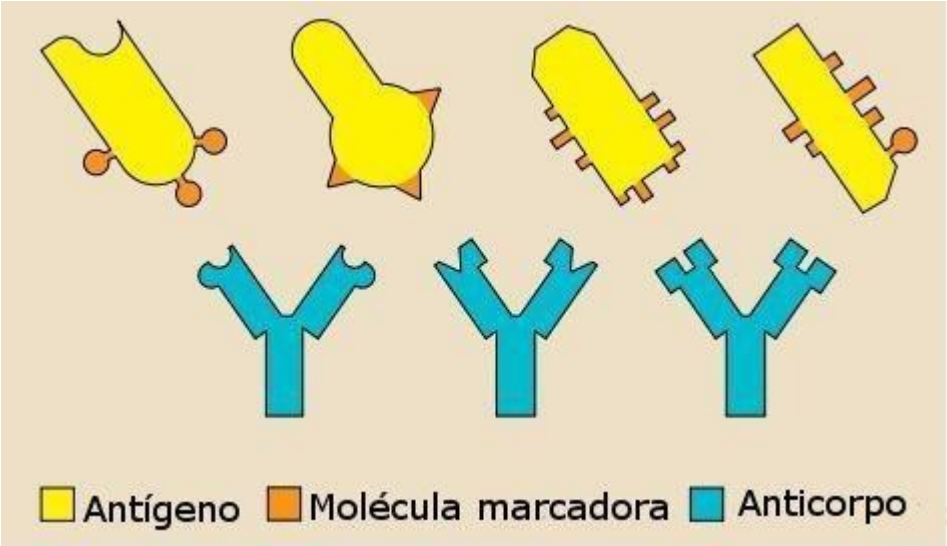

Figura 5.1 – Anticorpos e região hipervariável [HEALTH, 2003]

Quando uma célula B encontra o antígeno e recebe o sinal de uma célula T auxiliar, então ela se especializa numa célula B de plasma (ou plasmócito) que irá produzir mais anticorpos para aquele antígeno ou uma célula B de memória, que irá guardar a configuração do antígeno. Isto é útil para o sistema imunológico, pois quando o próximo ataque deste mesmo antígeno ocorrer a célula já tem uma memória para a defesa e a resposta imune acontece num período menor de tempo. Este princípio é o mesmo da vacinação [HEALTH, 2003].

5.1.3 – Células T

As células T também são linfócitos, mas não tem a capacidade de reconhecimento como as células B. Células T contribuem para a defesa do organismo de modo a regular a resposta imune ou atacando diretamente células infectadas. Assim como as células B, as células T também se especializam como será mostrado a seguir [KIMBALL, 2009]:

- *T Helper* (Linfócito T auxiliar): São as coordenadoras das respostas imunes, pois elas estimulam as células B para fabricação de anticorpos;
- *Cytotoxic T* (Linfócito T citotóxico): São as células que propriamente destroem células infectadas;
- *Regulatory T* (Linfócito T regulatório): Esta especialização das células T mantém a tolerância imunológica, ou seja, ela faz com que as células T não destruam um corpo próprio;
- *Natural Killer T* (Linfócito T matador natural): São células que atacam diretamente o antígeno, usando grânulos com substâncias químicas potentes.

5.2 – Sistemas imunológicos artificiais para um detector de intrusão de rede

De acordo com [SOMAYAJI, 1997], os princípios para um modelo de detecção para um sistema imunológico artificial são:

- proteção distribuída: o sistema imunológico consiste de milhões de agentes distribuídos pelo corpo. Estes componentes interagem localmente para prover proteção de modo distribuído. Não há controle ou coordenação centralizada;
- diversidade: existe muita diversidade entre sistemas imunológicos. Cada membro de uma população tem um sistema imunológico único. Isto provê robustez à população, pois um antígeno não afeta a diferentes membros da população do mesmo modo;
- robustez: a perda de alguns elementos do sistema imunológico tem pouco efeito. Este aspecto combinado com a proteção distribuída faz com que o sistema imunológico seja mais tolerante a falhas;
- adaptabilidade: o sistema imunológico tem capacidade de reconhecer patógenos com acurácia crescente, e isto aumenta a chance de uma resposta mais efetiva;
- memória (para detecções de intrusão baseadas em assinaturas): para a adaptabilidade ter sucesso, é necessário que exista uma memória imunológica. Esta memória corresponde aos sistemas de detecção baseados em assinatura;
- política de especificação implícita: a definição de próprio usado pelo sistema imunológico é definida empiricamente. Esta característica é implicitamente determinada pelo "monitoramento" de proteínas correntes no corpo. A

vantagem é que o “próprio” é considerado um estado normal. Em outras palavras, o sistema imunológico não sofre problemas quando tenta especificar corretamente uma política de segurança;

- flexibilidade: o sistema imunológico é flexível na alocação de recursos para proteção do corpo. Quando uma infecção séria ameaça o corpo, o sistema imunológico gera mais componentes, e em outras ocasiões ele usa menos recursos;
- escalabilidade: visto de uma perspectiva de processamento distribuído, o sistema imunológico é escalável. A comunicação e interação entre os componentes é local, então o *overhead* é baixo quando se aumenta o número de componentes;
- detecção de anomalias: o sistema imunológico tem a habilidade de detectar patógenos ao qual nunca foi exposto.

5.3 – Integração do detector de intrusão com o sistema imune

No início do funcionamento do sistema imune é modelado um determinado número de componentes que irão gerar uma resposta adaptativa das células T, ou seja, são escritas manualmente as regras de detecção que gerarão as novas em um primeiro momento.

Os pacotes ao serem analisados pelo detector de intrusão recebem sinais do sistema imunológico artificial para indicar o contexto deles. Este contexto pode ser seguro, contexto de perigo (indica comportamento patológico) e o contexto de PAMP (*Pathogen Associated Molecular Pattern* – Patógeno Associado a um Padrão de Moléculas), que é um fluxo considerado perigoso e que deve ser analisado na superfície de uma célula B. Nesta análise é provado se ocorreu alguma variação de ataques conhecidos.

O funcionamento do detector de intrusão com algoritmos do sistema imune é feito pós processando este alerta vindo de um sensor. A idéia é que estas informações sejam salvas para se construir cenários que representam uma ameaça. Deste modo é mais fácil o sistema imune processar um alerta, pois as ameaças a rede ocorrem em estágios, como foi discutido no Capítulo 3.

Estes cenários construídos são chamados de correlação de alertas de intrusão.

5.3.1 – Correlação de alertas de intrusão

A correlação de alertas tem como objetivo geral descobrir variações de novos ataques, que funciona basicamente remontando o grafo de ataque preexistente [TEDESCO, 2006].

Segundo [NING, 2003], este modelo de correlação é um modelo formal de ataques conhecidos como tipos de hiper alerta. Este tipo de hiper alerta é uma 3-tupla que representa (fato, pré-requisito, consequência), aonde o fato é um conjunto de alertas, pré-requisito é uma fórmula relacionada com o fato. E a consequência é um conjunto de fórmulas calculado com as variáveis livres da consequência.

É importante dizer que a reconstrução do cenário não captura a essência do ataque, somente reflete como ele foi desencadeado.

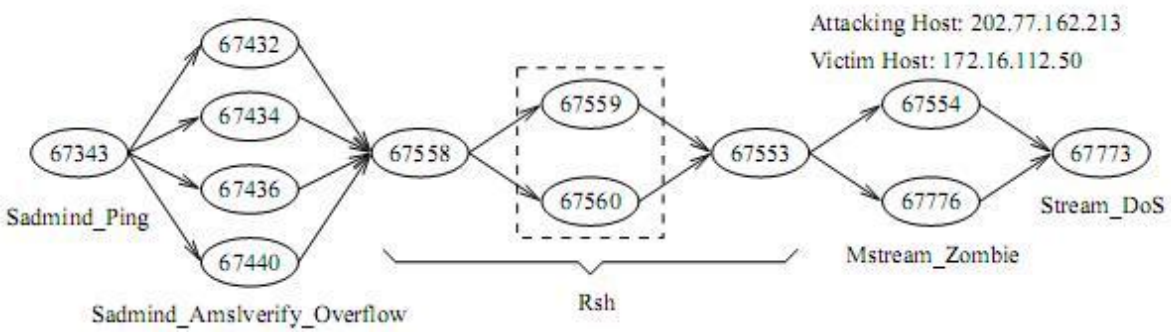

Figura 5.2 – Grafo de correlações de hiper alerta [NING, 2004]

5.4 – Grafos de ataque

É necessário que se extraia automaticamente as estratégias de ataque dos alertas discutidos anteriormente. O grafo de ataque tem este objetivo, além de salvar configurações de ataques para que no futuro não seja necessário refazer toda a correlação para ameaças invariantes.

Estes grafos são acíclicos diretos, e representam vários tipos de requisitos e consequências para um ataque [TEDESCO, 2006].

A estratégia que [NING, 2003] propõe para criar o grafo de ataque é a de preparar relações entre alertas e ataques já conhecidos. A representação do ataque neste momento é transformada de modo que haja algumas condições iniciais comuns, ou seja, há rearranjo nas arestas do grafo de ataque quando há a percepção de ataques parecidos.

De acordo com [DASGUPTA, 1999] este grafo pode não funcionar bem quando há presença de falsos negativos. Alguns sistemas podem ter a habilidade de processar este

alerta hipotético. Este processamento não é simples, e pode significar que o detector perdeu o pacote por problemas na rede ou porque não conhece a variação deste ataque.

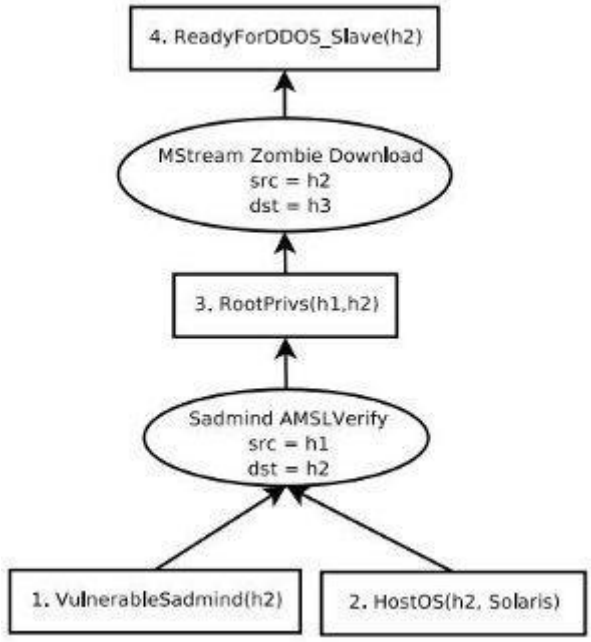

Figura 5.3 – Exemplo de grafo de ataque [TEDESCO, 2006]

Quando se monta o grafo de ataque a análise do tráfego se torna mais simples. A definição formal desta estratégia é mostrada a seguir.

> Dado um par de tipos de hiper alerta (T_1, T_2) uma *constraint* de igualdade para (T_1, T_2) é a conjunção de igualdades na forma de $u_1 = v_1 \wedge \ldots \wedge u_n = v_n$, aonde u_1, ..., u_n são nomes de atributos de T_2, logo existe $p(u_1, \ldots, u_n)$ e $p(v_1, \ldots, v_n)$ que são os mesmos predicados com argumentos possivelmente diferentes nos pré-requisitos de T_1 e T_2. Dado um tipo T_1 de hiper alerta h_1 e um tipo T_2 de hiper alerta h_2, h_1 e h_2 satisfazem a *constraint* de igualdade se t_1 pertencente a h_1 e t_2 pertencente a h_2 de modo que $t_1.u_1 = t_2v_1 \wedge \ldots \wedge t_1.u_n = t_2.v_n$

Figura 5.4 – Definição da formação do grafo de ataque [NING, 2004]

5.5 – Variações de ataques

O modelo que usa somente as técnicas descritas acima ainda não é capaz de reconhecer variações de ataques em um fluxo de pacotes. O algoritmo que gera e analisa o grafo de ataque gera um grafo diferente para cada sequencia de ataque quando são usados ataques equivalentes em tipo de ameaça, mas diferentes entre si. A modificação que

geralmente é feita é a de generalizar sintaticamente as diferenças entre tipos de hiper alertas, como por exemplo, se generaliza ataques de estouro de pilha [DASGUPTA, 1999].

Se esta generalização não for cuidadosamente feita, causa perda de informação sobre o ataque, além da perda da capacidade de reconhecimento de novos ataques. Além disto, com este modelo o atacante pode obter sucesso com um ataque completamente novo.

Para a geração automática desta generalização de ataques, os pré-requisitos para o ataque são expandidos de modo que predicados implícitos ao ataque também sejam levados em conta na geração do grafo [NING, 2003].

5.6 – O algoritmo imune

Para testar o funcionamento do detector será usada a biblioteca *libtissue* [TWYCROSS, 2006], que modelará o problema. Esta biblioteca foi escolhida, pois ela faz parte de um projeto chamado *Danger Theory* (Teoria do Perigo), que propõe um Sistema Imunológico Artificial que responde quando há infecção em algum corpo. Esta biblioteca vem sendo usada em testes de detecção de intrusão.

A *libtissue* é implementada como sistemas multiagentes de células, antígenos e sinais que interagem com o tecido (*tissue*). O tecido são os grafos montados a partir dos ataques descritos anteriormente. É um *software* que implementa e avalia Sistemas Imunológicos Artificiais monitorando e controlando problemas. É toda codificada na linguagem C e tem uma entrada para dados relativamente simples, e isto faz com que esta biblioteca tenha um propósito mais geral.

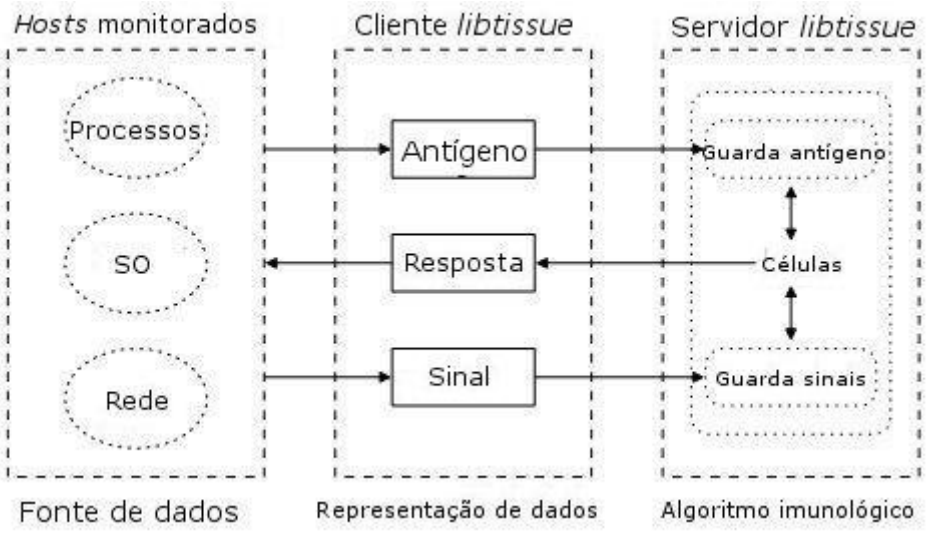

Figura 5.5 – Arquitetura do *libtissue*

A *libtissue* tem arquitetura cliente/servidor, aonde o servidor tem um algoritmo imunológico implementado que é executado como várias *threads* assíncronas, e o cliente provê a entrada ao algoritmo, além dos mecanismos de resposta que mudam de acordo com o sistema monitorado. Dentro dela o tecido se normaliza e cresce para popular o algoritmo com as células B. O servidor é o agente que contém as células, que possuem vários receptores que interagem com o antígeno. E os clientes são geralmente os sinais enviados, sinais e respostas (responsável pelo armazenamento de *logs*).

O papel desta biblioteca é analisar o fluxo que o detector captura para fazer a marcação do pacote de acordo com o contexto, além do pós-processamento deste fluxo.

5.7 – Resultados experimentais

Foi utilizado o *software* NIDS Firestorm versão 0.55 [TEDESCO, 2009], que é um detector de intrusão baseado em assinaturas que utiliza as mesmas assinaturas do Snort, um famoso *software* livre de detecção de intrusão para rede (NIDS). O Snort pode ser baixando em http://www.snort.org/.

Na sua instalação é criado um usuário exclusivo para o detector, para possibilitar a sua execução como serviço.

Este detector de intrusão assim como a *libtissue* também vem sendo utilizado para se testar a eficiência do Sistema Imunológico Artificial.

Para a instalação e execução correta do Firestorm é necessária a biblioteca *libpcap* para a captura de pacotes. As instruções para instalação estão contidas no arquivo README.

```
rodrigo@t311:~$ firestorm-nids

1259448544.593857 info: Firestorm NIDS v0.5.5
1259448544.593888 info: Copyright (c) 2002-2004 Gianni Tedesco
1259448544.593894 info: This program is free software; released
under the GNU GPL v2 (see: COPYING)
1259448544.594047 linux: Fast live capture for Linux
```

Figura 5.6 – Tela inicial do Firestorm

O Firestorm possui um arquivo de configuração facilmente modificável, na instalação padrão o arquivo se encontra em `/etc/firestorm.conf`, aonde se carregam tanto as regras nativas quanto as do Snort. Para adicionar uma regra do Snort basta adicionar uma linha neste arquivo com a seguinte sintaxe:

```
signatures snort <caminho-do-arquivo>
```

Figura 5.7 – Carregando uma regra do Snort no Firestorm

Para demonstrar e testar o funcionamento do Sistema Imunológico Artificial integrado ao Sistema de Detecção de Intrusão o Firestorm foi instalado e configurado em um servidor de modo que clientes pudessem enviar livremente tráfego com conteúdo malicioso.

O ataque foi feito através de uma variação de um *exploit* FTP com o comando "SITE EXEC" similar ao mostrado na Figura 4.5. O Firestorm foi carregado com as assinaturas e testado para ter certeza de que o *exploit* não foi detectado.

A modificação do *exploit* foi feita comparando-se padrões de regras do Snort com o código compilado do *exploit*. Deste modo foi possível modificar a porção exata do código fonte do programa malicioso que o detector reconhece de acordo com as regras especificadas, mas não foi possível determinar com exatidão se o *exploit* continuava malicioso.

O cenário de ataque foi estabelecido. Um cliente da mesma rede do servidor aonde foi instalado o detector de intrusão enviou pacotes do ataque juntamente com tráfego HTTP e FTP ao servidor (ruído na rede) para se configurar um cenário realista de ataque e também testar a eficiência da geração dos grafos de ataque.

Para a geração de tráfego na rede foi codificado um *script* em PHP para fazer requisições HTTP ao servidor Apache e outro *script* para listar diretórios em intervalos de tempos pré-definidos ao servidor FTP.

Na segunda tentativa todas as regras foram excluídas e foi repetido o mesmo processo.

A figura 5.7 mostra os resultados dos testes, aonde o total de pacotes foi o tráfego total analisado, os pacotes suspeitos são os pacotes que foram marcados para análise e os pacotes de saída se referem aos pacotes que realmente são os de variação de ataques.

Teste	Total de pacotes	Pacotes suspeitos	Pacotes de saída
1	20000	9000	26
2	50000	24000	83

Figura 5.8 – Resultados experimentais

A relação entre pacotes benignos e malignos não foi controlada nos testes para que o ambiente se aproximasse ao tráfego real.

Os resultados obtidos não foram comparados com outros detectores de intrusão, pois para esta análise seria necessário que os grafos de ataque fossem construídos de forma semelhante, além de ter que se considerar variáveis que estão fora do escopo deste trabalho.

A avaliação das técnicas ainda não tem completa acurácia por motivos de complexidade da análise e do entendimento completo do processo. Também há certa dificuldade para se encontrar dados para avaliação do detector e avaliar a qualidade do grafo de ataque gerado.

Estes testes podem ser considerados um ponto de início para a melhoria deste tipo de detector de intrusão.

Na próxima seção estes resultados e suas implicações teóricas são discutidos. Desta discussão serão apontados problemas e pontos positivos deste sistema.

5.8 – Conclusão

Foram mostrados alguns conceitos básicos do funcionamento de um sistema imunológico e como estes conceitos se aplicam em segurança de redes. Logo após foram identificadas premissas para um bom desempenho desta aplicação além de um exemplo de como um detector pode ser integrado a um Sistema Imunológico Artificial. Esta integração foi feita para se mostrar o funcionamento desta técnica.

Após estes testes é possível fazer uma melhor análise de quais componentes do Sistema Imunológico Artificial tiveram um bom funcionamento para a futura construção de um detector de intrusão mais sofisticado.

A análise dos resultados mostra que a detecção com sistema imune tem um alto número de falsos positivos. Alguns dos motivos conhecidos serão descritos a seguir.

Atualmente não é possível empregar técnicas que exijam um processamento mais complexo do fluxo, pois a análise é feita em um nível muito baixo de abstração. O que é possível de se fazer com o fluxo tem que ser feito no momento da entrada dele, pois se o detector não identificou nada neste momento então não há quase nada que se possa ser feito depois por ele. Deste modo não é possível fazer uma análise mais completa esperando-se mais tempo para montar mensagens que façam mais sentido. O desenvolvimento de técnicas mais avançadas para este processamento é um importante objeto de estudo para a melhoria do detector, sobretudo quando há tráfegos diferentes na rede.

Outro ponto importante é a generalização de hiper alertas que foi discutida anteriormente. Ela tem objetivo de esconder diferenças desnecessárias entre ataques diferentes. Isso simplifica a construção de regras e o trabalho do detector, mas pode também

deixar de encontrar detalhes críticos de ataques. O atacante que conhece a construção destas regras pode tomar vantagem deste aspecto. Esta generalização é fundamental no ponto de vista da eficiência do sistema imunológico. Atualmente sem esta generalização o detector se mostra com um custo computacional muito elevado comparado com detectores inatos, o que torna seu uso impraticável comercialmente. Uma causa que provavelmente limitou o teste foi a do ataque ter partido de somente um endereço IP.

Ainda existem vários outros aspectos a serem analisados para a melhoria do desempenho neste tipo de detector de intrusão. Existem muitos fatores que influenciam diretamente na execução destas aplicações que não foram mencionados ou sequer estudados por se tratar de uma área relativamente nova na Computação. Deste modo pode-se afirmar que é uma área promissora. Entre outros assuntos será abordado no próximo e último Capítulo o trabalho que poderá ser feito no futuro para o aprimoramento da detecção.

Capítulo 6 – Considerações finais

Como mostrado neste trabalho, junto com o crescimento da *Internet* cresceram as ameaças as redes. Estas ameaças podem comprometer seriamente o funcionamento da rede, que atualmente é utilizada para os mais diversos fins. O aumento destes ataques está relacionado principalmente à facilidade crescente de se realizar um ataque, além do fato de que qualquer usuário pode se tornar um atacante, alvo ou os dois ao mesmo tempo, pois mesmo que alguém não queira atacar outro na rede ele o faz devido os dados transmitidos pela *Internet* passarem por equipamentos controlados por outras pessoas.

Para manter a segurança foi visto que se deve atender a requisitos básicos, que são a confidencialidade, a autenticidade, a integridade, a disponibilidade, o controle de acesso e o não repúdio. Além destes aspectos foi mostrado que a criptografia é importante para manter o sigilo das informações, a verificar autoria e a garantir a sua integridade.

Foi mostrado ainda que as ameaças a rede podem vir de dentro da rede, e isto é um fator fundamental a ser levado em conta pelo administrador da rede. Estas ameaças podem ser desencadeadas por um simples clique em um *link* por um usuário desta rede. Os atacantes se aproveitam tanto deste aspecto quanto o de falhas existentes na rede ou em aplicativos que são executados computadores internos a rede. A adoção de uma política de segurança é fundamental para proteger a rede destes ataques.

Este estudo foi motivação para se mostrar os diversos tipos de ataques existentes. É importante que o administrador da rede conheça o comportamento das várias ameaças para poder se prevenir delas. Alguns métodos de defesa também foram discutidos.

Em relação a estes ataques foi visto que nem sempre uma atitude passiva de defesa é suficiente para a segurança da rede. Em alguns casos é necessário agir ativamente e tomar decisões para que não se comprometa a rede. O sistema de detecção de intrusão tem este papel, e foi mostrado que existem dois tipos básicos desta ferramenta, IDS baseado em *host* e IDS baseado em rede. Cada uma destas ferramentas apresenta um escopo de detecção diferente, com suas vantagens e desvantagens. Esta ferramenta combinada com outras ferramentas defensivas contribui para maior segurança na rede.

Um conceito relativamente novo de detecção de intrusão foi mostrado, que é a integração do detector com um sistema imunológico artificial. O objetivo o trabalho foi de mostrar o funcionamento desta ferramenta e discutir sobre. Para isso foi mostrado conceitos de sistemas imunológicos e como eles são aplicados em segurança de redes. Algumas premissas para o bom funcionamento deste sistema foram enunciadas e foi feito um teste para analisar o desempenho desta ferramenta.

Os resultados indicam que a pesquisa nesta área é promissora. E esta pesquisa é fundamental desde o início da construção de detectores de intrusão, pois foi deste modo que se deu a evolução na detecção e análise de ameaças.

Durante o desenvolvimento do trabalho foi notado que cada vez mais é necessário que os administradores de redes conheçam a fundo o funcionamento do detector para que possam ficar a par do que acontece na intrusão e o que se pode fazer para aumentar a segurança da rede, algo crucial nos dias atuais.

Muitas outras abordagens podem ser utilizadas para o estudo e descoberta de novas técnicas. O desenvolvimento dos testes permitiu um aprendizado aprofundado no assunto, e testes com maior abrangência podem ajudar a entender melhor as ameaças e o comportamento do sistema imunológico artificial para o desenvolvimento de trabalhos futuros.

Como trabalhos futuros propõem-se a realização de estudos sobre uma análise mais completa e eficiente do fluxo de pacotes, além de uma generalização mais eficiente de hiper alertas. A análise de resultados é um fator importante para a descoberta de novos objetos de estudo.

O estudo desta ferramenta foi muito importante para compreender tanto conceitos do sistema imunológico artificial quanto a do detector de intrusão, e permitiu a assimilação do conteúdo teórico pela prática demandada.

Referências

ABSOLUTA, Verdade. 1999. **Sistema de Detecção de Intrusão e Aspectos Legais**. Disponível em: http://www.absoluta.org/seguranca/seg_ids.htm. Acesso em: 4 de novembro de 2009.

AVAILABILITY, Confidentiality, Integrity and. **The information security glossary**. 2001. Disponível em http://www.yourwindow.to/information-security/gl_confidentialityintegrityandavailabili.htm. Acesso em: 8 de outubro de 2009.

BEALE, Jay; BAKER, Andrew; CASWELL, Brian; POOR, Mike. **Snort 2.1 Intrusion Detection**. 2 ed. Syngress. 2004. 731 páginas.

BITDEFENDER, Survey. **BitDefender Malware and Spam Survey finds E-Threats Adapting to Online Behavioral Trends**. 2009. Disponível em: http://news.bitdefender.com/NW1094-en--BitDefender-Malware-and-Spam-Survey-finds-E-Threats-Adapting-to-Online-Behavioral-Trends.html Acesso em: 17 de novembro de 2009.

BRUMLEY, David. **Invisible intruders: rootkits in practice**. 1999. Disponível em http://www.usenix.org/publications/login/1999-9/features/rootkits.html. Acesso em: 17 de outubro de 2009.

CERT. **Home Network Security**. 2006. Disponível em http://www.cert.org/tech_tips/home_networks.html#III-B-2. Acesso em: 22 de outubro de 2009.

CERT.BR. **Estatísticas dos Incidentes Reportados ao CERT.br**. 2009. Disponível em http://www.cert.br/stats/incidentes/. Acesso em: 17 de outubro de 2009.

CERT.BR. **Práticas de Segurança para Administradores de Redes Internet**. 2006. Disponível em http://www.cert.br/docs/seg-adm-redes/seg-adm-redes.html. Acesso em: 22 de outubro de 2009.

DASGUPTA, Dipankar. **Immunity-Based Intrusion Detection System: A General Framework**. National Information Systems Security (22: 1999: Baltimore, EUA). 1999.

DENNING, Dorothy E. **An Intrusion Detection Model.** Proceedings of the Seventh IEEE Symposium on Security and Privacy (7: 1986: Oakland, EUA). Anais. 1986. 13v número 2.

DENT, Alexander W; MITCHELL, Chris J. 2004 **User's Guide to Cryptography and Standards**. 1.ed. Artech House Publishers, 2004, 382 páginas.

FOSTER, James; OSIPOV, Vitaly, BHALLA, Nish. 2005. **Buffer Overflow Attacks: Detect, Exploit, Prevent.** 1 ed**.** Syngress, 304 páginas.

GRIFFITHS, Richard T. **From ARPANET to World Wide Web**. 2002. Disponível em http://www.let.leidenuniv.nl/history/ivh/chap2.htm. Acesso em: 15 de setembro de 2009.

INFORMATION, Consumer. **Nigerian Money offer scams**. 2004. Disponível em: http://www.aarp.org/money/consumer/articles/FraudsNigerianMoneyOffer.html. Acesso em 17 de novembro de 2009.

INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION (1: 1972: Washington, EUA). Anais. Washington, EUA, 1972. 5v.

ISO 7498-2. **Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture**. 1989.

HEALTH, National Institutes of,. **Understanding the immune system, how it works.** National Cancer Institute. 2003. 60 páginas.

HOGLUND, Greg; BUTLER, Jamie. 2005 **Rootkits: Subverting the Windows Kernel**. 1 ed. Addison-Wesley Professional. 2005. 352 páginas.

HOWE, Walt. **A Brief History of the Internet by Walt Howe**. 2009. Disponível em http://www.walthowe.com/navnet/history.html. Acesso em: 14 de outubro de 2009.

KIMBALL, John W. **B Cells and T Cells**. Disponível em http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/B/B_and_Tcells.html. Acesso em: 4 de novembro de 2009.

KLEINROCK, Leonard. **The Birth of the Internet**. 1996. Disponível em http://www.cs.ucla.edu/~lk/LK/Inet/birth.html. Acesso em: 17 de outubro de 2009.

KUROSE, James F.; ROSS, Keith W. 2006. **Redes de Computadores e a Internet: Uma abordagem top-down**. 3.ed. Pearson Addison Wesley. 2006. 659 páginas.

KOON, Tan. **Phishing and Spamming via IM (SPIM)**. 2006. Disponível em http://isc.sans.org/diary.html?storyid=1905. Acesso em: 22 de outubro de 2009.

KOHLENBERG, Toby; BEALE, Jay; BAKER, Andrew R. **Snort IDS and IPS Toolkit**. 1 ed. Syngress. 2007. 730 páginas.

LAUFER, Rafael P. 2003. **Introdução a Sistemas de Detecção de Intrusão**. Disponível em: http://www.gta.ufrj.br/grad/03_1/sdi/sdi-1.htm. Acesso em: 4 de novembro de 2009.

LICKLIDER, Joseph Carl Robnett. **Memorandum For Members and Affiliates of the Intergalactic Computer Network.** Washington, DC: 23 de abril de 1963.

LITMAN, Gary; CANNON, John; DISHAW, Larry. **Reconstructing immune phylogeny: new perspectives**. Nature reviews: immunology. São Petersburgo – EUA. p. 263 – 294. 2005.

LIU, Nian; LIU, Sunjun; LI, Rui; LIU, Yong. **A network intrusion detection model based on immune multi-agent.** International Journal of Communications, Network and System Sciences, Chengdu – China, v. 2, n. 6, p. 569 – 574, 2009.

MCCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. 2001. **Hacking Exposed.** McGraw-Hill, 2001. 1 ed. 700 páginas.

MEDICINENET. **Definition of cytokine**. 1999. Disponível em http://www.medterms.com/script/main/art.asp?articlekey=11937. Acesso em: 7 de novembro de 2009

NEYMAN, Jerzy; PEARSON, Egon. **On the use of certain test criteria for purposes of statistical inference**. 20A, p. 263 – 294. 1928.

NING, Peng; XU, Dingbang. **Learning Attack Strategies from Intrusion Alerts**. 10th ACM Conference on Computer and Communications Security, Washington – EUA. p. 200 – 209. 2003.

NING, Peng; CUI, Yun; REEVES, Douglas; XU, Dingbang. **Hypothesizing and reasoning about attacks missed by intrusion detection systems**. ACM Transaction Information System Security, Irvine – EUA. v. 7, n. 6, p. 591 – 627. 2004.

ONENESS, Global. **J.C.R. Licklider - Role in Early Computer Science Research.** 2009. Disponível em: http://www.experiencefestival.com/a/JCR_Licklider_-_Role_in_Early_Computer_Science_Research/id/1519297 Acesso em: 17 de novembro de 2009.

OPPLIGER, Rolf. 2001. **Internet And Intranet Security**. Artech House Publishers, 2001.

PRIVACY, **Data Protection Project**. 2006. Disponível em http://privacy.med.miami.edu/glossary/xd_confidentiality_integrity_availability.htm. Acesso em: 8 de outubro de 2009.

ROBERTS, Lawrence G. **Internet Chronology**. 2009. Disponível em http://www.packet.cc/internet.html. Acesso em: 17 de novembro de 2009.

STATS, World Internet Usage of, **Internet Usage Statistics**. Disponível em http://www.internetworldstats.com/stats.htm. Acesso em: 17 de novembro de 2009.

SOMAYAJI, Anil; HOFMEYR, Steven; FORREST, Stephanie. **Principles of a computer immune system.** New security paradigms workshop, Langdale – Reino Unido, p. 23 – 26, 1997.

SOUSA, Rafael T. de.; PUTTINI, Ricardo S. **Principais aspectos de segurança em redes**. 2009. Disponível em http://www.redes.unb.br/security/introducao/aspectos.html. Acesso em 11 de outubro de 2009.

SUTTON, Chris. **Internet Began 35 Years Ago at UCLA with First Message Ever Sent Between Two Computers**. 2004. Acesso em: 14 de setembro de 2009.

TAYLOR, Mark S; WAUNG, William; BANAN, Mohseh. **Internetwork Mobility – The CDPD Approach**. 1996. Disponível em http://www.leapforum.org/published/internetworkMobility/split/node96.html. Acesso em: 17 de outubro de 2009.

TEDESCO, Gianni; UWE, Aickelin. **An immune inspired Network Intrusion Detection System utilising correlation**. AISB '06: adaptation in artificial and biological systems. Society for the Study of Artificial Intelligence and the Simulation of Behaviour, Bristol – Reino Unido, 2006.

TEDESCO, Gianni. **Firestorm Network Intrusion Detection System**. Disponível em http://www.scaramanga.co.uk/firestorm/. 2009.

TENETS, Three. **The three tenets of cyber security**. 2009. Disponível em http://www.spi.dod.mil/tenets.htm. Acesso em: 19 de outubro de 2009.

TENG, Henry S.; CHEN, Kaihu; LU, Stephen C-Y. **Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns.** IEEE Symposium on Security and Privacy, Oakland, p. 278 – 284, 1990.

TWYCROSS, Jamie; AICKELIN, Uwe. **Libtissue - Implementing Innate Immunity**. IEEE Congress on Evolutionary Computation. Vancouver – Canadá, 2006

WANG, Jie, 2008, **Computer Network Security – Theory and Practice.** Springer, 2008. 384 páginas.

VACCA, John R., 2009, **Computer and Information Security Handbook**. 1 ed. Morgan Kaufmann. 2009. 928 páginas.