SISTEMAS PARA IDENTIFICAÇÃO DE INVASÃO

Marcos Aurelio Pchek Laureano

Curso de Mestrado em Informática Aplicada
Pontifícia Universidade Católica do Paraná

Curitiba, abril de 2002.

Resumo

Este trabalho visa demonstrar alguns conceitos sobre sistemas de detecção de intrusos (IDS – Intrusion Detection System), como funciona esta tecnologia, quais os problemas que ela enfrenta atualmente e quais as perspectivas para o futuro.

1 – O Rápido Crescimento dos Ataques

Temos acompanhando nos noticiários e informativos técnicos, o rápido crescimento dos ataques aos computadores, ligados em rede ou não. Este crescimento é perceptível a partir de um estudo realizado pela CERT/CC.

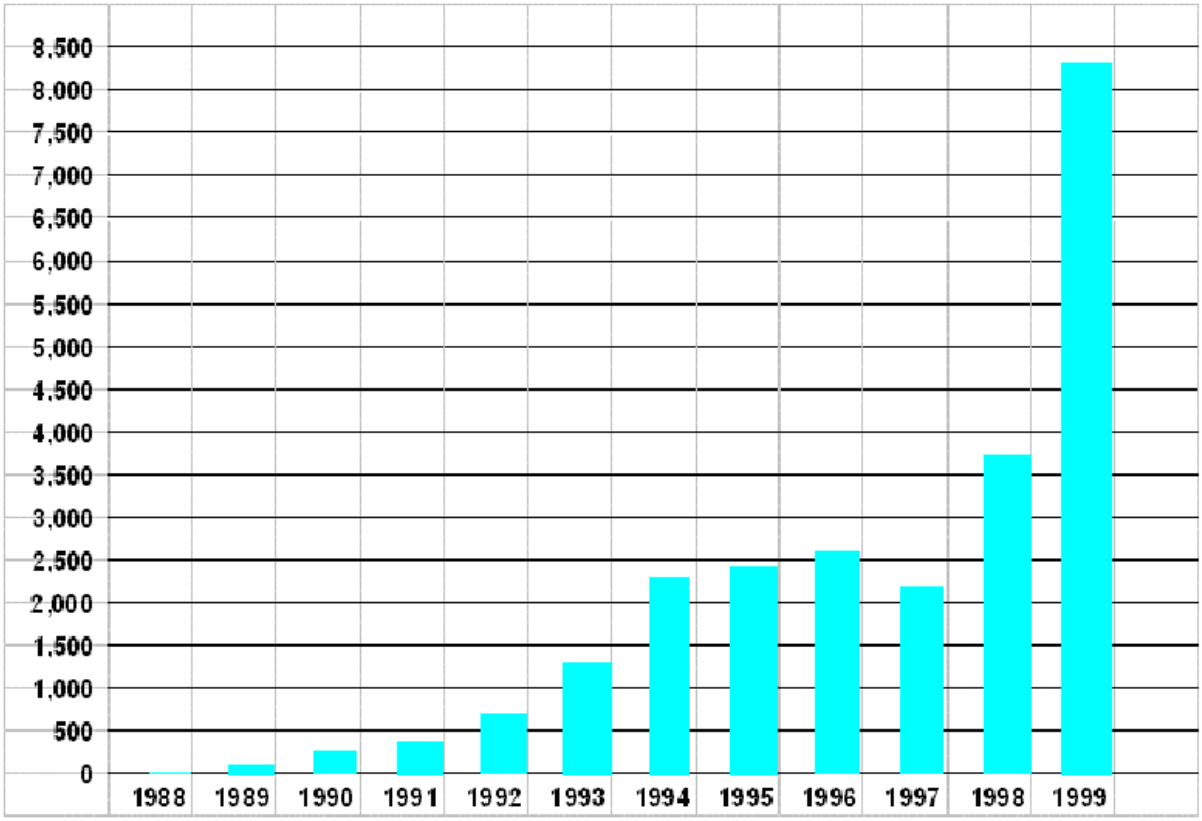

Figura 1 – Crescimento do número de incidentes controlado pela CERT/CC.

Nos anos 80 (embora as intrusões de computador ocorram desde os anos 60), as invasões eram realizadas somente por pessoas com alto conhecimento técnico. Com a popularização da internet, o perfil dos atacantes está mudando. É possível encontrar em vários *websites* ferramentas, *scripts* ou tutoriais que ensinam a burlar os sistemas de segurança atuais, o que possibilita que pessoas com pouco conhecimento possam efetuar ataques a computadores com pouca ou nenhuma proteção.

1.1 – Objetivos de um Ataque

Um atacante pode ter vários objetivos ao realizar um ataque, como causar dolo ou prejuízo ao atacante ou procurar vulnerabilidades no sistema para depois divulgar estas informações. Estas vulnerabilidades podem estar ligadas a um sistema mal-configurado ou a falhas na especificação de um software – como as falhas que possibilitam ataques de *buffer overflow*.

Como a indústria e governo ficam cada vez mais dependentes das redes para realizar negócios, intrusões de computador (com as metas de obter vantagens econômicas, competitivas, inteligência militar e políticas) são cada vez mais crescentes.

1.2 – Como Descobrir e se Proteger de um Ataque

A descoberta de um ataque bem-sucedido ou não, depende da análise do que está ocorrendo na rede. Pode-se utilizar *sniffers* que ficam “escutando” o tráfego de dados pela rede e comparando com um registro de assinaturas de possíveis ataques. Outros métodos de identificação analisam o comportamento dos sistemas, quando ocorre uma anomalia em relação ao comportamento usual, o sistema de identificação emite um alerta, neste caso, um comportamento não usual pode ser um comportamento válido (gerando uma mensagem de identificação falsa) e que deve ser registro para usos futuros.

É comum os administradores de sistemas disponibilizarem computadores como um chamariz para os atacantes – estes computadores também são chamados de "potes de mel". Os administradores analisam os ataques ocorridos e desenvolvem novas ferramentas e proteções contra esses ataques.

Muitas ferramentas de IDS realizam suas operações a partir da análise de padrões do sistema operacional e da rede tais como: utilização de CPU, I/O de disco, uso de memória, atividades dos usuários, número de tentativas de *login*, número de conexões, volume de dados trafegando no segmento de rede entre outros. Estes dados formam uma base de informação sobre a utilização do sistema em vários momentos ao longo do dia, outras já possuem bases com padrões de ataque (assinaturas) previamente montados permitindo também a configuração dos valores das bases bem como inclusão de novos parâmetros.

Com estas informações, a ferramenta de IDS pode identificar as tentativas de intrusão e até mesmo registrar a técnica utilizada. Uma ferramenta de IDS deve possuir algumas características, entre elas:

1. Deve rodar continuamente sem interação humana e deve ser segura o suficiente de forma a permitir sua operação em *background*, mas deve ser fácil compreensão e operação;
2. Deve ter tolerância a falhas, de forma a não ser afetada por uma falha do sistema, ou seja, sua base de conhecimento não deve ser perdida quando o sistema for reinicializado;
3. Deve resistir a tentativas de mudança (subversão) de sua base, ou seja, deve monitorar a si próprio de forma a garantir sua segurança;
4. Dever ter o mínimo de impacto no funcionamento do sistema;
5. Deve detectar mudanças no funcionamento normal;
6. Deve ser de fácil configuração, cada sistema possui padrões diferentes e a ferramenta de IDS deve ser adaptada de forma fácil aos diverso padrões.
7. Deve cobrir as mudanças do sistema durante o tempo, como no caso de uma nova aplicação que comece a fazer parte do sistema;
8. E deve ser difícil de ser enganada.

O último ponto faz referências aos prováveis erros que podem acontecer ao sistema. Estes podem ser classificados em: falso positivo, falso negativo e erros de subversão.

- Falso positivo – ocorre quando a ferramenta classifica uma ação como uma possível intrusão, quando na verdade trata-se de uma ação legítima; Um bom exemplo de falso positivo é o ataque de *SYN FLOOD*. O simples fato de acessar um determinado tipo de página pode gerar uma detecção da ocorrência de um ataque SYN FLOOD.
- Falso negativo – ocorre quando uma intrusão real acontece, mas a ferramenta permite que ela passe como se fosse uma ação legítima;
- Subversão – ocorre quando o intruso modifica a operação da ferramenta de IDS para forçar a ocorrência de falso negativo.

2 – As dificuldades

A maior dificuldade relativo a um sistema de detecção de invasão – *Intrusion Detection System* (IDS) – é identificar e classificar o que é realmente uma tentativa de acesso não autorizado ou simplesmente um erro eventual, ou uma distração para ocupar os administradores de sistemas enquanto o verdadeiro ataque ocorre.

Devido à complexidade das redes de computadores, suas ligações com softwares e outros hardwares, é muito difícil avaliar ferramentas de IDS. O rápido crescimento e diversidade de ataques que surgem a cada dia impedem que uma ferramenta de IDS seja

constantemente atualizada. Estas ferramentas acabam por agir somente sobre alguns tipos de ataques mais conhecidos ou comparam o tráfego de rede buscando padrões conhecidos de ataques.

O número de tentativas de invasão seria menor, se as ferramentas de IDS que são utilizadas fossem devidamente configuradas. Também não existe quantidade suficiente de pessoas com o conhecimento técnico adequado para configurar e trabalhar com ferramentas de IDS. Por exemplo, não é raro encontrar um *firewall* ou um *router* com a senha padrão de fábrica, o que facilita aos atacantes ter o acesso a estes equipamentos.

2.1 – Ferramentas de Ataque

O crescente desenvolvimento de ferramentas de ataques é um desafio às tecnologias de detecção de intrusos.

Alguns exemplos de ferramentas:

Ferramentas de Varredura – Permite um atacante determinar características do sistema atacado. Um exemplo conhecido é o *SATAN*;

Ferramentas de Administração Remota – Utilizado por um administrador de sistemas para acessar máquinas remotas. Pode causar danos significantes se utilizados como ferramenta de ataque. Um exemplo é o *Back Orifice*. Pode ser identificado através de programas antivírus.

Sniffers – Ferramentas que "cheiram" o tráfego de rede, permite a visualização dos pacotes que passam pela rede.

Vulnerabilidades novas surgirão com tempo, e conseqüentemente ferramentas que exploram estas vulnerabilidades.

2.2 – Criptografia

A criptografia dos pacotes é um problema, especialmente para IDS de redes. A prática de procurar padrões de assinatura não funciona em pacotes criptografados. E com o crescimento da utilização de criptografia, fica cada vez mais claro que um IDS deve ter a habilidade de verificar pacotes criptografados.

A utilização de criptografia por chave-pública elimina a necessidade da verificação, se os pacotes vierem assinados digitalmente, o que garante a origem e autenticidade dos dados.

2.3 – Modem

A utilização de um *modem* numa máquina ligada à rede aumenta consideravelmente o risco a segurança da rede. Um atacante pode utilizar este ponto de entrada para realizar um ataque a rede. Este tipo de ataque não pode ser barrado por um IDS baseado/localizado em um *firewall*.

Se a utilização de um modem for essencial, o sistema deve ser projetado para monitor a central telefônica e evitar ataques desta natureza.

2.4 – Códigos Móveis

A utilização de códigos móveis (*ActiveX* e *applets* Java) está se popularizando pela sua facilidade de uso. Estes códigos podem conter ataques embutidos e que tragam prejuízos para a vítima. A maioria destes programas possua um nível básico de segurança (um certificado digital), mas devemos lembrar que um atacante também pode obter um certificado digital. Alguns *firewalls* podem prevenir a entrada de *ActiveX* e *applets* Java na rede, através do reconhecimento dos tipos *MIME*, mas nenhum tem a funcionalidade de analisar o conteúdo destes programas (embora estejam ocorrendo esforços para se criar *firewalls* mais "inteligentes").

Um IDS deveria ser capaz de:

- Analisar o conteúdo e prever o comportamento de um código móvel;
- Verificar a localização de um código móvel a avaliar se aquele código poderia estar naquele local;
- Analisar a execução do código e emitir um alerta para ações suspeitas;
- Anotar as atividades do código.

2.4 – Redes – Complexidade e Escalabilidade

Os IDS atuais não conseguem funcionar em ambientes com tecnologias e políticas diferentes. Um IDS para estas redes deveria ser capaz de:

- Integração das informações de intrusão através de formatos e dados diferentes;
- Compartilhamento das informações de intrusão sensíveis com os ambientes não-confiáveis;
- Coordenação interdomínio (políticas e ferramentas);
- Segurança global da rede, mesmo que haja falhe de um IDS local.

2.5 – Sistemas Operacionais

Os sistemas operacionais não são projetados para operar de forma segura, esta é uma das razões pela alta demanda de IDS. O sistema operacional deveria ser projetado de forma que um administrador possa cadastrar regras e padrões de segurança. Existe um vasto conjunto de estudos referentes ao desenvolvimento de um sistema operacional seguro, mas como não foram implantados comercialmente, são considerados, na prática, fracassos.

2.6 – Padrões e Interoperabilidade

As ferramentas atuais não possuem padrões comuns e não se comunicam. O IDS ideal deve se capaz de se comunicar com outros IDS para compartilhamento de informações sobre intrusão.

2.7 – Fatores Humanos e Organizacionais

As empresas não trocam informações sobre ataques sofridos, o IDS deve ser projetado para permitir a troca de informações sobre ataques entre as empresas, o objetivo é o bem comum de todos.

As habilidades humanas não está sendo aproveitado de forma coerente para identificar ataques e tomar decisões, o IDS deve prover um módulo que permita um operador humano interagir com a detecção de intrusos, utilizando o poder computacional do computador para analisar o que se passa na rede e o poder humano de interpretar de dar significado e estabelecer relações entre ataques.

Ainda não temos sistemas capazes de analisar o perfil de um atacante e o real objetivo do ataque, não existe uma base histórica sobre ataques anteriores deste atacante.

A quantidade de terminologias e produtos IDS dificulta o aprendizado e assimilação das informações que surgem a todo instante, as constantes modificações dos IDS comerciais dificulta a escolha do produto.

2.8 – Aspectos Funcionais

Os IDS não conseguem detectar um ataque nos seus estágios iniciais. Não são detectadas sondagens iniciais em busca de vulnerabilidades no sistema.

Os IDS não estão preparados adequadamente para responder automaticamente a um ataque. Uma defesa automática e ideal não deve requerer intervenção humana, tem que ser mais rápido que um humano e pode ser customizado com políticas de segurança específicas.

Os sistemas não provêem pouco ou nenhum apoio para a recuperação de danos causados por um ataque. Uma forma de recuperação de dados seria assinar criptograficamente os arquivos de comparar regularmente estas assinaturas (como o *Tripwire* faz).

O tráfego constante de dados passando pela rede ainda é um obstáculo para os IDS, já que eles não conseguem manter o ritmo constante de avaliação dos pacotes de dados em redes muito rápidas ou fortemente carregada. Também não está preparado para ataques de consumo de recursos da máquina, ou seja, um atacante sobrecarrega a CPU, memória ou canal de comunicação e força que o IDS perca um tempo considerável analisando pacotes desnecessariamente.

Infelizmente, os sistemas atuais não possuem “inteligência” o suficiente para reconhecer e aprender sobre novos ataques, um sistema eficaz deveria demonstrar maior adaptabilidade a novas ameaças.

Devido a grande quantidade de IDS comerciais, não é possível validar a suficiência e eficácia das assinaturas de ataques existentes. Isto não ocorre com sistemas de domínio público.

Não existe um parecer sobre o desempenho ideal de um IDS, o ideal é detectar a maior quantidade de ataques possíveis e emitir a menor quantidade possível de falso avisos.

Faltam efetivamente ferramentas para análises de auditoria e prover provas para uso forense (legal), e ferramentas de apoio para análise humana de dados de intrusão que estão relacionados entre si.

É comum ocorrer diagnósticos inexatos devido a informações insuficientes e análises simplistas. O sistema precisa melhorar o diagnóstico e a forma (algoritmos) que são utilizados para se chegar nele.

3 - Ferramentas Existentes

A tecnologia de IDS é imatura e dinâmica (está continuamente em desenvolvimento). Um IDS não utiliza medidas preventivas, quando um ataque é descoberto age como um informante. A maneira mais comum para descobrir intrusões é a utilização dos dados das auditorias gerados pelos sistemas operacionais e ordenados em ordem cronológica de acontecimento, sendo possível à inspeção manual destes registros, o que não é uma prática viável, pois estes arquivos de *logs* apresentam tamanhos consideráveis.

O IDS automatiza a tarefa de analisar estes dados da auditoria. Estes dados são extremamente úteis, pois podem ser usados para estabelecer a culpabilidade do atacante e na maioria das vezes é o único modo de descobrir uma atividade sem autorização, detectar a extensão dos danos e prevenir tal ataque no futuro, tornando desta forma o IDS uma ferramenta extremamente valiosa para análises em tempo real e também após a ocorrência de um ataque.

Existem dois tipos de implementação de ferramentas IDS:

a) *Host Based*: são instalados em servidores para alertar e identificar ataques e tentativas de acesso indevido à própria máquina, sendo mais empregados nos casos em que a segurança está focada em informações contidas em um servidor e os usuários não precisam ser monitorados. Também é aplicada em redes onde a velocidade de transmissão é muito alta como em redes "*Gigabit Ethernet*" ou quando não se confia na segurança corporativa da rede em que o servidor está instalado.

b) *Network Based*: são instalados em máquinas responsáveis por identificar ataques direcionados a toda a rede, monitorando o conteúdo dos pacotes ou do tráfego e seus detalhes como informações de cabeçalhos e protocolos.

O IDS tem como um dos objetivos principais detectar se alguém está tentando entrar no seu sistema ou se algum usuário legitimo está fazendo mau uso do mesmo. Esta ferramenta é executada constantemente em *background* e somente gera uma notificação quando detecta alguma ocorrência que seja suspeita ou ilegal. A verificação passiva está dentro dos sensores e são os padrões de ataques e assinaturas, ou seja, os ataques conhecidos previamente, a verificação ativa também está dentro dos sensores e são caracterizadas por possuírem inteligência para aprender com o comportamento da rede e com isso identificar novos padrões ou mutação dos padrões existentes.

Um sensor é composto por um conjunto de componentes, entre eles: sub-sensor estático, sub-sensor inteligente e aprendiz. O sub-sensor estático deve inicialmente ser configurado de acordo com a política de segurança além de possuir as assinaturas dos ataques conhecidos, isso caracteriza a verificação passiva, já o sub-sensor inteligente inicialmente passa por um período de adaptação e aprendizado fase em que o sensor aprende e reconhece o padrão de funcionamento da rede, este período pode ser variável dependendo do volume de tráfego, após esta fase estes sub-sensores inteligentes estariam em condições de reconhecer padrões que fogem da normalidade da rede e tomarem ações.

Os sensores podem ser de dois tipos:

- Os sensores de rede devem localizar-se em segmentos estratégicos observando o tráfego da rede e os formatos de pacotes;
- Os sensores de *hosts* são localizados nos servidores críticos observando as ações realizadas no sistema operacional, as ações dos serviços e o comportamento do tráfego TCP/IP.

Os sensores devem interagir entre si a fim de construírem uma matriz de eventos que tem por objetivo a qualificação do padrão de ataque, minimizando desta forma a ocorrência de alertas falsos (falso positivo). Outras características fundamentais são: o gerenciamento centralizado, a possibilidade do sensor interagir com outros elementos de rede como *firewall*, roteadores e consoles de gerência; e a possibilidade de construir uma base de conhecimento centralizada de forma a permitir uma visão ampla do nível de segurança da rede.

Desta forma quando algum ataque for detectado pelos sensores torna-se possível ação de conta-ataque que podem ser: envio de e-mail para o administrador, envio de mensagem via *pager*, ativação de alertas nas estações de gerência via SNMP, re-configuração de elementos de rede como *firewall* e roteadores, e até mesmo o encerramento da conexão.

A quantidade de ferramentas existentes para avaliação de intrusão nos sistemas é grande, alguns são feitos por estudantes para praticar conceitos apreendidos em sala de aula, nestes casos, as ferramentas não tem continuidade. Dentre as ferramentas existentes, podemos citar o EMERALD, NetSTAT, BRO e outros exemplos comerciais.

3.1 - EMERALD

O EMERALD - *Event Monitoring Enabling Responses to Anomalous Live Disturbances* (Monitoração de Eventos que Ativa Resposta as Perturbações Anômalas Vivas), que está sendo desenvolvida pela SRI Internacional, verifica se houve uma intrusão baseando em desvios de comportamento do usuário (anomalias) e padrões de intrusão conhecidos (assinaturas). A meta principal do EMERALD é trabalhar com redes de empresas grandes (heterogêneas). Estes ambientes são difíceis de monitorar e de se analisar devido à diversificação da informação que trafega pela rede. O EMERALD estrutura os usuários em

um conjunto de domínios independentemente administrados. Cada conjunto provê uma cobrança de serviços de rede (*ftp*, *http*, *telnet*) que podem ter relações de confiança e políticas de segurança diferentes entre si.

A estrutura hierárquica provê três níveis de análise: monitores de serviço, domínio e empresa. Estes monitores possuem a mesma arquitetura básica: um conjunto de mecanismos de perfil (para descobertas de anomalia), mecanismos de assinatura e um componente determinador que integra os resultados gerados pelos mecanismos. É possível configurar e personalizar cada nível.

No mais baixo nível, o monitor de serviço suporta a detecção de intrusão para os componentes individuais e serviços de rede dentro de um domínio, sondando ou verificando *logs* e eventos, e verificando assinaturas e realizando análises estatísticas. Os monitores de domínio integram a informação do monitor de serviço para prover uma visão de invasões, enquanto os monitores de empresa executam uma análise interdomínio para avaliar as ameaças de uma perspectiva global.

O NIDES demonstrou técnicas de análise estatísticas que poderiam ser efetivas com usuários ou aplicações. A monitoração de aplicações (*ftp* anônimo, por exemplo), era efetiva se menos perfis de aplicação fossem exigidos. O EMERALD generaliza a técnica de perfil pela abstração do que é um perfil, separando gerenciamento de perfil de análise de perfil.

O EMERALD é um trabalho em progresso. Provê um exemplo da direção que as ferramentas de IDS em desenvolvimento podem tomar.

3.2 - NetSTAT

O NetSTAT é a mais recente ferramenta de uma linha de ferramentas de investigação "STAT" produzida pela Universidade da Califórnia em Santa Bárbara. A atividade STAT explora o uso da análise de transição de estados para descobrir a intrusão em tempo real.

Sistemas de IDS baseados em *hosts* analisam se houve uma invasão a partir da análise de trilhas de auditoria. Porém, na análise STAT a informação de trilha de auditoria é transformada por um "analisador de trilha de auditoria" que filtra e abstrai as informações que são recolhidas pela trilha de auditoria. Estas abstrações, que são mais adequados para análise, portabilidade e compreensão humana, são chamadas de assinaturas e é a principal aproximação para o STAT. As ações de assinatura movem o sistema para uma seqüência de estados, cada mudança de estado deixa o sistema perto de uma configuração acertada. Seqüências de intrusão são definidas pelos diversos estados que são capturados num sistema baseados em regras.

A aplicação inicial do método era um sistema baseado em *host*, desenvolvido para Unix e chamado de USTAT. Era composto de um pré-processador, uma base de conhecimento (ações e regras), um mecanismo de inferência e um mecanismo de decisão.

A mais recente ferramenta, o NetSTAT está atualmente sob desenvolvimento e difere dos sistemas baseados em endereçamento de intrusão de rede. O NetSTAT é composto de sondas que agem remotamente em cada sub-rede, caso alguma sonda identifique algum componente de intrusão, um evento é enviado as outras sondas interessadas para adquirir mais detalhes sobre a intrusão. Desta forma é possível identificar intrusões em sub-redes. As sondas são suportadas por um analisador, que é responsável pela administração da base de dados (base de conhecimento). É o analisador que determinam quais e como os eventos serão monitorados.

3.3 - BRO

O BRO é uma ferramenta de investigação que esta sendo desenvolvida por *Lawrence Livermore National Laboratory*. Está sendo construído, em parte, para explorar as emissões

relacionadas à robustez de ferramentas de IDS, isto é, avaliando quais características fazem um IDS resistir a ataques contra si mesmo. As metas do projeto abrangem:

- Monitoração de *high-load* (capacidade de controlar altos tráfegos de rede);
- Notificação em tempo real;
- Separação de políticas de filtros, identificação e reação aos eventos. Facilita a aplicação e manutenção do sistema;
- Um amplo banco de dados com relação a ataques conhecidos e habilidade de acrescentar novos ataques a esta base;
- Habilidade para repelir ataques contra si mesmo.

O BRO trabalha com uma hierarquia de três níveis, na camada mais baixa é utilizado um utilitário chamado *libpcap*, este utilitário extrai pacotes da rede associados aos protocolos *finger*, *ftp*, *portmapper* e *telnet* que são os protocolos sobre os quais o BRO trabalha. A camada de evento executa verificações de integridade dos cabeçalhos (*headers*) dos pacotes, que verifica se deve ser feita uma análise mais profunda do pacote ou não. Na terceira camada os pacotes passam por um *script* que verifica as políticas de segurança.

Atualmente o BRO monitora quatro aplicações (*finger*, *ftp*, *portmapper* e *telnet*), novas aplicações podem ser adicionadas a partir de uma derivação de uma classe em C++, somente devem ser acrescentado algumas informações que correspondem à nova aplicação que se deseja monitorar. Sobre condições de alto tráfego (uma rede FDDI de 25 megabits com uma análise de 200 pacotes por segundo).

3.4 - Outros exemplos comerciais

CMDS (*Computer Misuse Detection System*) – Sistema baseado em *host* que verifica invasões de mudança de perfil (anomalia) ou assinatura.

NetProwler – Sistema baseado em *host* que verifica assinaturas. Permite que o usuário adicione novas assinaturas de ataque.

NetRanger – É um IDS baseado em rede. Opera em tempo real e é escalável. Os sensores são espalhados pela rede e conversam com o software principal. Permite a análise de três categorias de ataques: ataques nomeados, gerais e extraordinários (com algo grau de complexidade).

Centrax – Sistema baseado em *host* que permite verificar pacotes criptografados. Pode reagir localmente a ameaças em tempo real e cada ataque pode ter um padrão de resposta diferente (como encerramento de uma conexão a máquina atacada).

RealSecure – Outro exemplo de IDS em tempo real. Baseado numa arquitetura de três níveis: um mecanismo de reconhecimento baseado em *host*, outro baseado em rede e o terceiro é um módulo administrador.

3.5 - Exemplos de Ferramentas de Domínio Público

Pelo fator de não possuir empresas por trás destas ferramentas, a instalação e manutenção destas ferramentas fica comprometida. Mas é válido avaliar estas aplicações para entender como funciona a tecnologia de IDS.

Shadow – Utiliza chamadas a sensores (que ficam espalhados pela rede) e estações de análise. A filosofia do *Shadow* é não emitir alertas, simplesmente são criados arquivos de *log*. Utiliza o utilitário *libpcap* para prover uma capacidade de *sniffer* básica.

NFR (*Network Flight Recorder*) – IDS disponível em uma versão comercial e outra de domínio público (mais antiga). Utiliza o *libpcap* para extrair de forma aleatória e passiva os pacotes da rede para análise. Pode agir fora de um *firewall* para descobrir ameaças em pontos mais distantes da rede. Possui uma linguagem de programação completa que permite desenvolver *scripts* de análise mais completos.

Tripwire – Como o NFR, existe uma versão comercial e outra de domínio público (códigos fontes disponíveis para UNIX). *Tripwire* trabalha de forma diferente de outros IDS, ele verifica os arquivos de sistema em busca de mudanças. É feito um *checksum* entre o arquivo de sistema e as informações que estão armazenadas num sistema seguro. Com o *Tripwire* é possível restaurar os arquivos modificados. Trabalha com o conceito de assinatura criptográfica.

3.6 - Produtos GOTS (*Government Off-the-Shelf*) – Produtos que não estão a venda

Enquanto as empresas utilizando IDS visando proteger a sua rede para obter lucro (não perdendo dados e produtos) e/ou não ser responsável por uma invasão que possa ocorrer com seus parceiros comerciais ou acionistas, os órgãos governamentais também estão interessados em proteger a sua rede, mas com um outro foco e objetivo: A proteção da soberania nacional.

No seminário "*Detection of Malicious Code, Intrusions, and Anomalous Activity*" realizado em 1999 e patrocinado pelo Departamento de Energia, Conselho de Segurança Nacional e Departamento de Ciência e Políticas Tecnológicas e que teve a participação de especialistas ligados a setores do governo e comércio, chegou-se a algumas considerações:

- Qualquer atacante que tenha o apoio de uma nação terá maiores recursos e ferramentas que um atacante comum;
- Um IDS para fins governamentais deve ter a capacidade de avaliar e armazenar a maior quantidade de dados possíveis para análise e uso futuro – identificação de intenção;
- O objetivo de um IDS numa empresa é barrar o ataque e evitar qualquer prejuízo, para a nação é descobrir as respostas para as questões: Quem ? O que ? Por quê ? Quando ? E como ? Um IDS comercial não tem a necessidade de responder a estas questões – e para sua própria sobrevivência, dificilmente virá a responder estas questões;
- Qualquer pessoa pode adquirir um IDS comercial, neste caso um atacante sabendo que um governo está utilizando este IDS pode adquirir e descobrir como derrotá-lo.

Um governo sempre terá uma necessidade de que nenhum produto comercial irá atender, para estas necessidades é essencial que o governo continue a desenvolver IDS GOTS.

3.7 - Exemplos de produtos GOTS

3.7.1 - CIDDS (*Common Intrusion Detection Director System*)

Também conhecido como *CID Director*, é um sistema operacional, hardware e software dedicado, é utilizado pela Força Área Americana. Está baseado em:

- C, C++ e Java;
- Funções, *procedures* e *scripts* SQL do banco de dados Oracle (além da própria estrutura de banco de dados);
- *Scripts Shell* (*Born Shell*) e arquivos de configurações.

Estes componentes se comunicam com um sensor *ASIM* que recebe conexões de um sensor *host*. Estas informações são armazenadas numa base de dados local e através de uma ferramenta gráfica *user-friendly* é possível analisar e estudar indicadores de intrusos na rede.

Estas análises tentam identificar atividades suspeitas que possam acontecer em longos períodos de tempo, verificar os alvos, analisar a tendência e propósitos destas atividades.

O *CIDDS* combinado com os sensores *ASIM* provem uma capacidade pró-ativa e informa o mais próximo do tempo-real uma informação de descoberta de um incidente.

3.7.2 - ASIM (*Automated Security Incident Measurement*)

É um software baseado nas mesmas linguagens de programação do *CIDDS*. Captura, filtra e analise pacotes de dados *FDDI*. O *ASIM* é um *sniffer* e analisador de pacote de dados promíscuo. Verifica o protocolo *TCP*, *UDP* e *ICMP*, trabalhando em dois modos de operação (lote e tempo real).

Em tempo real o *ASIM* analisa e identifica tentativas de intrusão, neste caso o módulo Diretor associado é notificado. Em lote, o *ASIM* captura o tráfego da rede para uma análise mais detalhada. Os dados monitorados são criptografados e enviados para um analista humano. O analista irá determinar se as atividades suspeitas são realmente tentativas de intrusão.

3.8 - O futuro para um IDS

Permanece em aberto se a tecnologia de descoberta de intrusão pode cumprir a promessa de identificar ataques com precisão, são muitas as propostas, mas poucos resultados na prática. A tecnologia atual utiliza um universo pequeno de técnicas para descobrir ataques de invasão. A tecnologia de invasão está evoluindo mais rapidamente que a de detecção.

As estratégias de ataque evoluem mais rápido do que IDS. Um ataque pode ser dividido em 7 (sete) fases:

- Reconhecimento – O atacante sonda a provável vítima, procurando identificar em qual parte do servidor sofrerá o ataque;
- Identificação de vulnerabilidade – Identificar quais serviços ou produtos estão vulneráveis a um ataque;
- Penetração – Derrota de qualquer perímetro do limite de segurança de um *firewall*;
- Controle – Ganhar o controle da rede e remover os sinais de invasão;
- Incrustar – Manter o controle sobre a vítima (mesmo se descoberto) através de código malicioso;
- Extração de Dados – Retirada de informação em baixas taxas, utilizando ou não um protocolo comum para a retirada dos dados. Os dados podem ser mascarados em um formato de arquivo comum;
- Transmissão de ataques – Utilizar a vítima para realizar outros ataques.

Com a velocidade que os ataques evoluem, são necessárias mais ferramentas com tecnologias sofisticadas e adaptáveis, a fusão entre múltiplas fontes de dados (arquivos de

assinatura), uma interação entre o computador e o homem de forma mais integrada e treinamento em políticas de segurança mais efetivas.

Um documento da ICSA intitulado "*An Introduction to Intrusion Detection and Assessment*" identifica o que um IDS poderia fazer:

- Conceder um maior grau de segurança a infra-estrutura da rede;
- Utilizar freqüentemente informações de fontes obtusas, enquanto narra o que realmente está acontecendo em seu sistema;
- Liberar um sistema de monitoração que rastreie a Internet a procura de novos ataques;
- Realizar o gerenciamento de segurança dos seus sistemas com assessoria de um possível não perito;
- Conter diretrizes que ajudem a estabelecer uma política de segurança;
- Investigar a atividade de um usuário do ponto de entrada até o ponto de saída ou impacto;
- Reconhecer padrões de atividade que refletem ataques conhecidos e alerte as pessoas apropriadas;
- Análise estatística para padrões de atividades anormais;
- Gerenciamento de uma trilha de auditoria, com reconhecimento de atividades de usuário que violem as políticas de segurança.

Uma outra discussão promovida pela *Computer Security Institute* (CSI) em 1998 com peritos sobre IDS oferecem as seguintes perspectivas:

- Deve-se esperar de um IDS que ele descubra ataques comuns de forma oportuna;
- IDS atuais tem a habilidade de visualizar a rede e suas atividades em tempo real, identificam atividades sem autorização e provem uma resposta mais próxima possível do tempo real. Possuem a capacidade de analisar as atividades ocorridas e identificar tendências e problemas futuros. IDS bons serão projetados para operar num nível técnico, porém ainda irão requerer de análises consideráveis para entender os dados e saber o que fazer com a resposta;
- Um IDS teria que ter ferramentas de detecção para guiar uma investigação, deveria ter um procedimento operacional que junte informações adicionais para ajustar a rede e o processo;
- Deve ficar claro para os usuários que um IDS não irá proteger a sua rede em 100% dos casos, e sim diminuir significamente o risco de uma invasão;
- Você pode aprender mais sobre o que está acontecendo em sua rede, recolher dados sobre o que está enviado para suas redes remotas, e usar estas informações para tomar decisões sobre os controles de segurança que precisa ser desenvolvido;

3.9 - Observações sobre as ferramentas de IDS existentes

O CERT após alguns testes realizados com alguns IDS's (*RealSecure*, *NetRanger*, *Shadow* e *NFR*) chegou a algumas conclusões:

- A decisão do local de instalação do IDS é importante, face às implicações de segurança existentes. A maioria dos IDS's requeria duas interfaces: uma insegura para monitoração e outra segura para administração e comunicação com o IDS. Esta não é uma solução considerada segura (já que existe o mesmo nível de atenção e confiança que se deve ter num *firewall*). O ideal era utilizar um protocolo seguro (como o *SSH*) para realizar estas comunicações.

- Das ferramentas avaliadas, nenhum tinha uma configuração compreensível e de fácil utilização. As assinaturas são ajustadas individualmente. O ideal era a utilização de perfis (para agrupamento de ataques semelhantes).
- Não foi encontrada nenhuma indicação de integração entre os *scanners* de vulnerabilidades e as ferramentas de configuração (de forma a adicionar automaticamente novas assinaturas).
- A maioria das ferramentas tem a capacidade aumentar a monitoração da rede (embora nenhuma ferramenta deixe claro esta posição). Basicamente, o IDS pode monitorar pacotes por política de *firewall*, máquinas sem *patchs* instalados para corrigir vulnerabilidades específicas e serviços de redes específicos.

Glossário

Ataque – O ato de tentar desviar dos controles de segurança de um sistema. Um ataque pode ser ativo, tendo por resultado a alteração dos dados; ou passivo, tendo por resultado a liberação dos dados. O fato de um ataque estar acontecendo não significa necessariamente que ele terá sucesso. O nível de sucesso depende da vulnerabilidade do sistema ou da atividade e da eficácia de contramedidas existentes.

Auditoria – Revisão e exame dos registros e das atividades do sistema para avaliar sobre sua confiabilidade, executados com independência.

Bastion Host – Um sistema montado para resistir a ataques que é instalado numa rede potencialmente sujeita a ataques. *Bastion hosts* são freqüentemente componentes de *firewalls*, ou podem ser servidores *web* externos ou sistemas de acesso público.

Buffer overflow – Vulnerabilidade que ocorre quando um programa recebe mais dados do que consegue processar.

Checksum – Um valor calculado a partir de parte de dados que pode ser usado para verificar que o dado não foi alterado.

Denial of service – Negação de Serviço – O impedimento do acesso autorizado aos recursos ou o retardamento de operações críticas por tempo.

DNS Spoofing – Usar o endereço DNS de um outro sistema corrompendo o *cache* do sistema da vítima, ou comprometendo um *domain name server* para um domínio válido.

Flood – Do inglês inundação, transbordar, normalmente ataque associado à técnica de "entupir" o sistema com comandos ou pacotes específicos visando à interrupção ou queda do sistema.

FDDI – *Fiber Distributed Data Interface* – é um padrão designado pelo *National Standards Institute* (ANSI) comitê X3T9.5, com a participação de várias empresas de produtos e serviços de computação e telecomunicações. É uma rede em duplo anel usando fibra ótica como meio físico para transmissão de dados a uma taxa de 100 Mbps.

Firewall - Um sistema ou combinação de sistemas que protege a fronteira entre duas ou mais redes

ICMP – *Internet Control Message Protocol* – Parte integrante do Protocolo de Internet (IP) que lida com mensagens de erro e de controle. Especificamente, roteadores e *hosts* usam ICMP para enviar relatórios de problemas relativos aos datagramas ao ponto original que os enviou. O ICMP também inclui uma solicitação/resposta de eco, usada para verificar se um destino é alcançado e está respondendo.

IP Spoofing – Um ataque em que um sistema assume ilicitamente a personalidade de outro sistema usando seu endereço de rede

Libpcap – Utilitário desenvolvido por *Lawrence Berkeley Laboratories Network Research Group*.

Patch – Do inglês remendo ou curativo, é uma correção ou aprimoramento de um aplicativo ou sistema.

Spoofing – Tentativa de ganhar acesso ao sistema iludindo ser um usuário autorizado.

SSH – *Secure Shell*.

SSL – *Secure Sockets Layer* – protocolo que possibilita realizar comunicações seguras através de criptografia e autenticação.

Trilha de auditoria – Histórico das transações de sistemas que estão disponíveis para a avaliação a fim a provar a correção de sua execução comparada com os procedimentos ditados pela política de segurança. Relaciona-se a uma chave ou transação que permite que as quebras na segurança sejam detectáveis.