INSTITUTO DE PESQUISAS TECNOLÓGICAS DO ESTADO DE SÃO PAULO

VINICIUS AKIYAMA

Gerenciamento de Segurança de Servidores Web utilizando SNMP

São Paulo

2005

VINICIUS AKIYAMA

Gerenciamento de Segurança de Servidores Web utilizando SNMP

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, para obtenção do título de Mestre em Engenharia de Computação
Área de concentração: Redes de Computadores

Orientador: Prof. Dr. Wagner Luiz Zucchi

São Paulo
2005

Ficha Catalográfica
Elaborada pelo Centro de Informação Tecnológica do IPT

A315g Akiyama, Vinicius
Gerenciamento de segurança de servidores Web utilizando SNMP. / Vinicius Akiyama. São Paulo, 2005.
101p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. Wagner Luiz Zucchi

1. Gerenciamento de segurança 2. Ataque 3. Servidor Web 4. Sistema de detecção de intrusão 5. Intrusion Detection System - IDS 6. Simple Network Management Protocol - SNMP 7. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Centro de Aperfeiçoamento Tecnológico II. Título

05-32 CDU 004.056.53(043)

Dedicatória

À minha família, por iluminar a minha vida.

Agradecimentos

À minha família, por me dar todo o suporte que preciso e muito mais do que mereço ao longo de toda minha carreira.

À IBM do Brasil, por acreditar no meu potencial.

Ao meu orientador, Prof. Dr. Wagner Luiz Zucchi, pela paciência e suporte técnico durante o desenvolvimento deste trabalho.

Ao amigo Marcio, por me auxiliar a corrigir os erros no código do agente.

À equipe da secretaria do CENATEC, sempre prestativa durante todo o curso.

RESUMO

A corrida para participar da Internet trouxe novas e sérias ameaças aos negócios das empresas. Cada vez que é instalado um novo servidor Web, é aberta uma janela para a rede interna à qual toda a Internet poderá ter acesso.

Para possibilitar o crescimento íntegro deste novo meio de comunicação e comércio, muito tem sido investido no desenvolvimento de tecnologias de segurança.

Diversas ferramentas que auxiliam na prevenção e detecção de ataques, como firewalls, SDIs (Sistemas de Detecção de Intrusão) e programas antivírus foram desenvolvidas, apresentando, contudo o problema de estabelecerem plataformas proprietárias. Isto dificulta o trabalho do administrador de segurança, oferecendo poucas alternativas de customização além de permitir integração somente entre ferramentas do mesmo fabricante.

Este trabalho apresenta uma solução de gerenciamento de segurança para servidores Web baseada na plataforma SNMP através da especificação e implementação de uma MIB genérica especialmente projetada para esta finalidade.

Palavras-chave: Ataque, assinatura, SNMP, IDS, servidores web.

ABSTRACT

The run for the Internet brought new and serious threats to the business. Everytime a new Web server is installed, a window is opened to the internal network, that all the Internet users can reach.

To make possible the grow of the new means of communications and trade, high investments on security technologies research have being done.

Many detection and attack prevention tools have being developed, like firewalls, IDS (Intrusion Detection Systems) and anti-viruses, establishing, although proprietary platforms. The fact dificults the security administrator's work, presenting few customization alternatives and preventing the integration with other vendors' tools.

This paper presents a security management solution for Web servers based on SNMP platform through the specification and implementation of a generic MIB specially designed to accomplish this task.

Key-words: Attack, signature, SNMP, IDS, web servers.

Lista de figuras

Lista de tabelas

Lista de abreviaturas, siglas e símbolos

API	Application Program Interface
ASN.1	Abstract Syntax Notation One
CERT	Computer Emergency Response Team
CGI	Common Gateway Interface
DoS	Denial of Service
FTP	File Transfer Protocol
GUI	Graphical User Interface
GPL	General Public License
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IAB	Internet Architecture Board
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
MIB	Management Information Base
NIST	National Institute of Standards and Technology
OID	Object ID
OSI	Open Systems Interconnection
RFC	Request for Comment
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

Sumário

1 Introdução

No atual cenário econômico mundial, se uma empresa não possui um *site* Web, ela pode ser considerada excluída. Contudo, esta corrida para participar da Internet criou um ambiente onde *sites* Web e de comércio eletrônico multiplicam-se a uma taxa assustadora. Esta rápida proliferação de *sites* Web também criou novas ameaças aos negócios.

A partir do momento em que é instalado um servidor Web, é aberta uma janela para a rede interna à qual toda Internet poderá ter acesso. A maioria dos visitantes contenta-se em navegar, olhar, porém uma minoria tentará apoderar-se de elementos não destinados ao "consumo público". Outros, não satisfeitos em olhar e pegar, tentarão modificar o conteúdo. Os resultados destes acessos indevidos podem variar de uma simples situação embaraçosa ao descobrir que o conteúdo da página foi substituído por paródias obscenas, até danos, por exemplo, referentes ao roubo da base de dados completa de informações de clientes.

É uma máxima em sistemas de segurança, que softwares defeituosos abrem brechas na segurança (*security holes*). É uma máxima em desenvolvimento de software que programas extensos e complexos contêm defeitos. Infelizmente, servidores Web são extensos e complexos que podem (em diversos casos foi provado) conter falhas de segurança (abrir brechas na segurança - *security holes*). Além disto, a arquitetura aberta dos servidores Web permite que scripts arbitrários escritos em CGI sejam executados no servidor em resposta a solicitações remotas. Qualquer script CGI instalado no *site* pode conter erros e cada erro é uma brecha na segurança em potencial.

Do ponto de vista do administrador de redes, um servidor Web representa outra brecha em potencial à segurança da rede. A meta, de modo geral, dos sistemas de segurança da rede é manter estranhos do lado de fora. No entanto, a missão de um servidor Web é fornecer ao mundo acesso controlado à rede. Traçar o limiar entre os dois não é tarefa trivial. Um servidor Web mal configurado pode abrir uma bela brecha no mais bem desenhado sistema de *firewall*. Um *firewall* mal configurado pode impossibilitar o acesso ao serviço Web. A tarefa torna-se particularmente mais difícil em um ambiente intranet, em que o servidor Web deve tipicamente estar configurado para reconhecer e autenticar diversos grupos de usuários, cada qual com privilégios distintos.

Outro ponto bastante delicado é que o protocolo TCP/IP não foi desenvolvido objetivando-se segurança, portanto existe a vulnerabilidade à espionagem na rede. Quando documentos confidenciais são transmitidos de um servidor Web para um browser, ou quando

o usuário final envia informação privada para o servidor através do preenchimento de um formulário, alguém pode estar escutando.

Os riscos são muito grandes, porém necessários. Assim sendo, a necessidade de proteger os servidores de informação é real e muito séria. Para ilustrar a importância do tema segurança da informação, o documento Computer Crime and Security Survey (Relatório de Pesquisa de Segurança e Crimes em Sistemas de Computadores, que apresenta os resultados da pesquisa realizada com especialistas em segurança de 530 das mais importantes corporações norte-americanas) de 2003 [RIC2003], elaborado pelo CSI / FBI informa que as perdas totais relatadas pelos participantes da pesquisa no período de 2002 a 2003 representam um montante equivalente a US$ 201.797.340,00. Apesar do valor ter apresentado uma queda de 56% em relação ao período anterior, sugerindo que a severidade e os danos causados pelos ataques tiveram uma queda pela primeira vez desde 1999, talvez devido à maior preocupação das empresas em monitorar o acesso aos *sites*, notou-se que mesmo as organizações que investiram pesado em tecnologias de segurança sofreram perdas significativas. Pode-se concluir, desta forma, que os riscos de ataques e conseqüentes prejuízos continuam muito elevados.

As ameaças podem ser desde um funcionário descontente a um adolescente deslumbrado pela possibilidade de obter informações restritas, que não precisam gastar horas e horas procurando por vulnerabilidades nos sistemas, ou semanas estudando técnicas de invasão. Precisam somente de um mecanismo de procura na Internet como fonte de informações de invasão e em alguns instantes é possível obter os programas que exploram as vulnerabilidades dos sistemas operacionais e aplicações. Esta proliferação de informações na Internet sobre técnicas de invasão permite, mesmo a novatos, invadir sistemas, alterar *sites* Web, acessar informações confidenciais e comandar ataques de Denial of Service (DoS) para derrubar um servidor causando grandes perdas para seu proprietário.

Diversas ferramentas, ainda que por si só, insuficientes, foram desenvolvidas para auxiliar na prevenção e detecção de ataques (algumas delas extremamente poderosas, com capacidade de reconfigurar o sistema reagindo a um comportamento suspeito) como firewalls, IDS (sistemas de detecção de intrusão) e programas antivírus, porém existe o problema de estabelecerem uma plataforma proprietária, fechada, não permitindo a integração com sistemas e módulos auxiliares, como ferramentas de auxílio no tratamento dos eventos, que não sejam do mesmo fabricante, nem mesmo a alteração do seu código para melhor adequação às necessidades do ambiente a ser protegido.

1.1 Objetivo

Este trabalho tem como objetivo definir uma solução de gerenciamento de segurança de servidores Web, através de dados coletados via MIB SNMP especialmente projetada para este fim.

A solução implementa o método de detecção de intrusão baseado em assinaturas dos ataques mais conhecidos e atividades suspeitas para inibir atacantes com conhecimento e recursos limitados. Estas assinaturas são registradas em arquivos padronizados que permitem atualização à medida que novos ataques são identificados.

A utilização da arquitetura SNMP para análise do problema proposto deve-se ao fato desta oferecer um conjunto de padrões (recomendados em RFCs) para gerenciamento de sistemas baseados em TCP/IP, incluindo um protocolo, especificação de estrutura para base de dados e um conjunto de objetos, amplamente aceitos pelo mercado.

1.2 Justificativa do tema

A proposta do novo enfoque para implementação de sistemas de detecção de intrusão justifica-se por diversos fatores, dentre os quais podem-se citar:

- Disponibilização do sistema sob formato de ferramenta de código aberto e distribuição livre, de acordo com os termos da licença GNU GPL [FSF1991], que poderá evoluir rapidamente para oferecer melhores resultados na detecção de atividades hostis em servidores Web;
- Uma grande vantagem decorrente da disponibilização do código aberto do sistema é que os defeitos poderão ser solucionados com o auxílio de diversos especialistas de todo o mundo, em um intervalo de tempo normalmente pequeno, pois estarão interessados na qualidade do sistema;
- Além do fator técnico, a disponibilização gratuita visa atender o maior número de interessados em melhorar a segurança de seu ambiente Web, sem onerar a empresa com custos de licenças;
- Baixo custo operacional, pois o sistema oferece recurso de atualização automática de novos padrões de ataques e eventuais manutenções não interferem na operação do servidor Web monitorado;

- A utilização da estrutura SNMP possibilita a fácil integração com a maioria dos sistemas de gerenciamento de redes, ou de forma mais abrangente, de sistemas disponíveis no mercado;
- Ocorrências de ataques poderão ser notificados à console central de eventos do sistema de gerenciamento de redes, ou sistemas, através de mensagens (Traps) padronizadas do protocolo de gerenciamento de redes TCP/IP, SNMP.

1.3 Organização da dissertação

Esta dissertação está organizada da seguinte forma, o Capítulo 2 apresenta a estrutura de um serviço Web, descrevendo seus componentes, mas destacando os servidores Web e a distribuição destes produtos no mercado.

O Capítulo 3 apresenta a arquitetura típica de segurança de ambientes Web, destacando a importância da solução agregando as funcionalidades de todos os componentes. Individualmente não podem proteger um ambiente Web de todos os tipos de ataques.

O Capítulo 4 apresenta os fundamentos sobre detecção de intrusão e detalha o componente da arquitetura de segurança responsável por realizar esta função, o Sistema de Detecção de Intrusão, ou IDS – Intrusion Detection System, fornecendo a base técnica para o entendimento da solução proposta neste estudo.

No Capítulo 5 é apresentada a ferramenta desenvolvida neste estudo, o Web-IDS, destacando sua estrutura, que é um dos diferenciais em relação às soluções existentes no mercado, e a operação de seus componentes.

O Capítulo 6 descreve os testes realizados para a homologação da ferramenta e apresenta os resultados obtidos. Complementando os testes, o Capítulo 7 apresenta o relatório da avaliação da ferramenta baseada no guia do IEEE apresentado em [AMO1998].

As conclusões, planos para evolução da ferramenta e sugestões para trabalhos futuros são apresentados no Capítulo 8.

2 Arquitetura de serviços Web

A arquitetura de serviços Web é muito similar ao modelo centralizado de computação, com clientes cujas atividades limitam-se basicamente a apresentação dos dados, se conectando a um robusto servidor central que faz todo o volume de processamento. O grande diferencial da arquitetura Web dos modelos de computação centralizada tradicional, é que elas dependem substancialmente da tecnologia HTML (HyperText Markup Language) e seu principal meio de transporte, o HTTP (HyperText Transfer Protocol). A Figura 2-1 ilustra os componentes típicos da arquitetura de serviços Web.

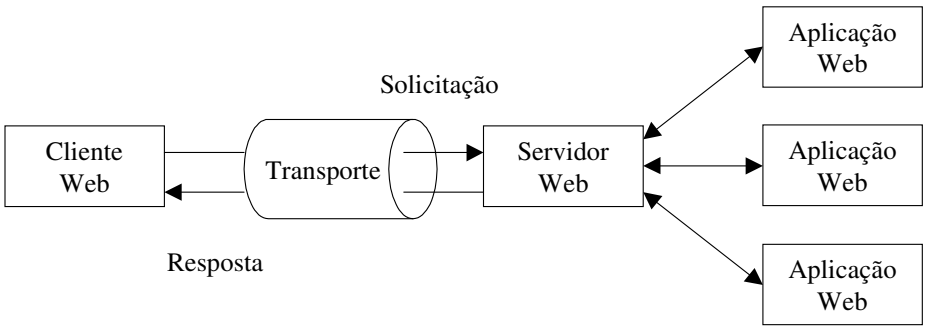


Figura 2-1- Componentes da arquitetura de serviços Web

2.1 Cliente Web

A aplicação cliente Web padrão é o navegador Web. Ele se comunica através do HTTP (além de outros protocolos) e interpreta as páginas, escritas em HTML e outras linguagens de markup, que apresentam os dados processados pelo servidor Web.

2.2 Transporte – Protocolo HTTP

O protocolo HTTP serve como meio de comunicação entre o cliente e o servidor. A versão 1.0 do protocolo, definido na RFC 1945 (a versão 1.1 é tratada na RFC 2616

[FIE1999]), é relativamente simples, stateless, baseado em ASCII e tipicamente opera na porta TCP 80 (entretanto, pode operar em qualquer outra porta).

A simplicidade do HTTP deriva de seu limitado conjunto de capacidades, solicitação e resposta. O HTTP define um mecanismo para solicitar um recurso, que o servidor retorna se for capaz. Os recursos são denominados URIs (Uniform Resource Identifiers) e podem variar de páginas estáticas de texto a conteúdo de vídeo dinâmico.

Nenhum conceito de estado de sessão é mantido pelo protocolo, considerado portanto, *stateless*. Isto é, cada solicitação é considerada separada e única.

O HTTP opera em uma porta TCP bem conhecida. Apesar de poder ser implementado em qualquer outra porta, todos os navegadores Web tentam conectar-se automaticamente à porta TCP 80 primeiro, portanto todos os servidores Web esperam por conexões nesta porta. De acordo com [SCA2002], isto tem inúmeras implicações na grande maioria das redes que se encontram atrás dos Firewalls, que supostamente os protegem das ameaças do mundo exterior. Firewalls são considerados praticamente sem defesa contra o hacking na Web quando configurados para permitir tráfego pela porta 80 através de um ou mais servidores.

2.3 Servidor Web

Segundo [SCA2002], o servidor Web pode ser considerado um serviço (na plataforma Unix, é denominado daemon) HTTP que recebe solicitações do cliente, realiza uma avaliação simples para garantir, entre outras verificações que o recurso existe e então o entrega à lógica da aplicação Web para processamento. Quando a lógica retorna uma resposta, o serviço HTTP a retorna ao cliente.

Em um estudo sobre aspectos qualitativos como a segurança, é importante considerar a distribuição dos produtos no mercado, isto é, identificar a participação de cada produto e, a partir destes valores direcionar o foco dos estudos para aqueles que detêm a maior parcela. As informações apresentadas a seguir foram extraídas do relatório de market-share de servidores Web elaborado e publicado mensalmente pela Netcraft. A Figura 2-1 apresenta o market-share entre setembro de 1995 a 2003 e a Tabela 2-1, valores somente considerando os *sites* ativos, de acordo com [NET2003].

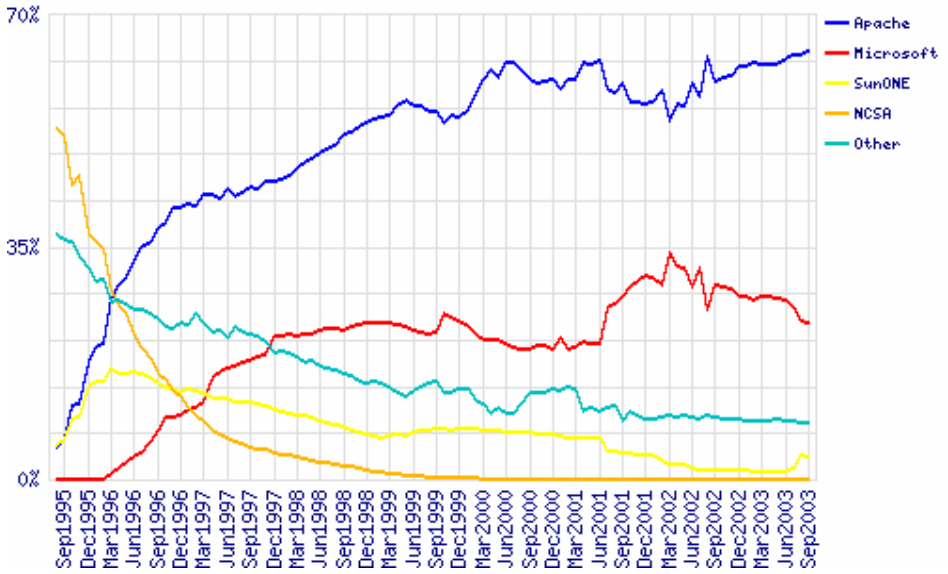


Figura 2-2 - Market-share dos servidores Web entre Setembro de 1995 e 2003

Tabela 2-1 - Distribuição dos servidores Web ativos

Fabricante	Quantidade (unidades)	Percentual
Apache	13.371.621	67,45
Microsoft	4.804.550	24,43
Zeus	266.220	1,34
SunONE	211.234	1,07

O SunONE contabiliza os servidores baseados em iPlanet-Enterprise, Netscape-Enterprise, Netscape-FastTrack, Netscape-Commerce, Netscape-Communications, Netsite-Commerce & Netsite-Communications.

Considerou-se Microsoft a soma dos servidores rodando Microsoft-Internet-Information-Server, Microsoft-IIS-W, Microsoft-PWS-95 e Microsoft-PWS.

2.4 Aplicação Web

A aplicação Web é o componente lógico principal de um Web *site* moderno. Apesar de a lógica executada pelo cliente ter evoluído bastante, exigindo maior grau de processamento, ainda o trabalho executado pelo servidor é o ponto principal.

O bloco Aplicação do diagrama pode ser compreendido como uma arquitetura de vários níveis, transformando em um mecanismo dinâmico de funcionalidade que possibilita grande interação com o usuário.

Existe uma enorme diversidade de técnicas e tecnologias utilizadas para criar a lógica Web de vários níveis, como ASP, o Java e o PHP. Isto contribui para a complexidade da arquitetura Web geral, o que por sua vez, aumenta o risco de exposições da segurança.

A exploração destes aspectos relativos às aplicações Web, apesar de oferecer um vasto campo de estudo, está fora do escopo deste trabalho.

3 Arquitetura de segurança de ambientes Web

Conforme define [STE1999], ameaça é "qualquer circunstância, ou evento que pode causar danos à uma organização através da exposição, alteração, ou destruição de informação, ou negação de serviços críticos."

Uma vez que a Internet constituiu-se importante meio de comunicação e negócios, através do comércio eletrônico, os ataques à infra-estrutura Web tornaram-se verdadeiras ameaças às empresas.

Ataques a provedores de informação, isto é, a qualquer servidor Web constituem um problema crescente. Nas últimas duas décadas, o crescimento do número de ataques à infra-estrutura Web reportados a instituições como o CERT Coordination Center, equivalem ao crescimento da Internet. No entanto, o que causa maior preocupação é a sofisticação destes ataques.

À medida que os *sites* de comércio eletrônico tornam-se alvos preferenciais e os ataques mudam suas características de invasão para negação de serviço (Denial of Service, DoS), o cenário torna-se mais grave. A maioria dos invasores pioneiros queria somente provar que conseguiam invadir os sistemas, no entanto, este panorama vem mudando progressivamente para invasões motivadas por objetivos financeiros, políticos e até militares.

Na década de 1980, a maioria dos invasores era especialista com alto grau de conhecimento e cada um desenvolvia seus próprios métodos para invadir os sistemas. Raramente utilizavam ferramentas automatizadas e scripts de exploração. Atualmente, qualquer pessoa pode atacar os *sites* da Internet utilizando ferramentas de invasão e scripts que exploram vulnerabilidades amplamente divulgadas, facilmente encontradas principalmente na própria Internet. A Figura 3-1 ilustra a relação entre a sofisticação das ferramentas de ataque e o conhecimento dos atacantes a partir da década de 1980.

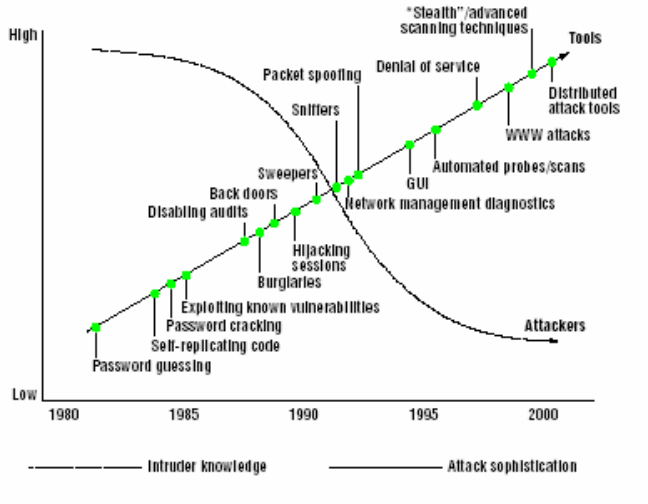


Figura 3-1 - Crescimento da sofisticação das ferramentas de ataque em relação ao conhecimento técnico do atacante [AL2000]

Atualmente, invasões devastadoras podem acontecer em questão de segundos. Os atacantes mascaram sua presença instalando versões alteradas de ferramentas e comandos de monitoração e administração do sistema e encerram a atividade apagando seus rastros em arquivos de registros (logs) e auditoria. Este tipo de comportamento dos atacantes é apresentado no capítulo 4. Na década de 1980 até o início da seguinte os ataques de negação de serviço eram raros e desconsiderados. Hoje, ataques bem sucedidos de negação de serviço podem deixar inoperantes organizações baseadas em comércio eletrônico.

Para poder responder à altura da complexidade dos ataques, é necessário adotar uma postura defensiva estruturada e bem implementada, denominada por [MCH2000] como Defesa Intensa.

3.1 Postura de Defesa Intensa

Postura defensiva estruturada em camadas em que cada camada corresponde a um elemento. Começa pelo estabelecimento de uma política de segurança apropriada e efetiva.

Políticas efetivas auxiliam a entender quais ameaças podem ocorrer aos componentes críticos do ambiente e quais medidas devem ser tomadas quando um incidente acontecer. Sempre que possível, a política deve refletir a missão da organização. Desta forma

poderão auxiliar na configuração de todos os elementos da arquitetura de segurança (firewalls, anti-vírus, IDS).

Estabelecer uma arquitetura de segurança em camadas é vantajoso, independente da implementação ou não de uma delas. Deve-se conhecer bem a importância do ambiente Web para o negócio da empresa, pois conforme [FRA1997], um axioma da segurança é que o custo para proteger algo contra uma ameaça deve ser menor que o custo da reconstituição dos danos causados por essa ameaça. Desta forma, é recomendável que quanto maior a importância do ambiente, maior o número de camadas da arquitetura que devem ser implementadas. A Figura 3-2 ilustra alguns aspectos da arquitetura de segurança.

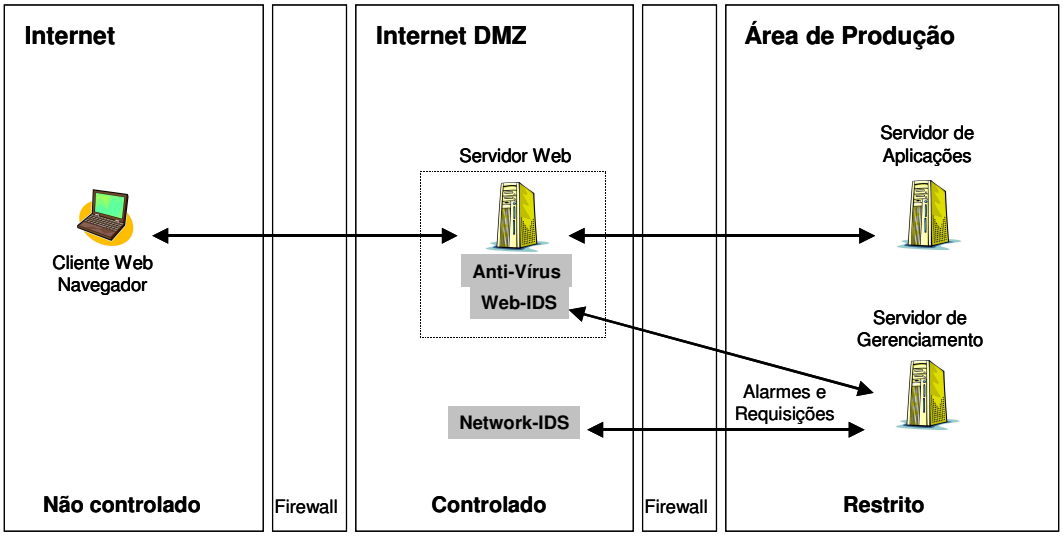


Figura 3-2 - Arquitetura típica de segurança de ambiente Web

Além de elaborar a política de segurança, as etapas essenciais para o desenvolvimento da arquitetura de segurança são:

- Identificar as áreas de acesso;
- Definir autenticação dos usuários e controle de acesso;
- Separar os serviços;
- Identificar os serviços desnecessários;
- Levantar todas as correções dos serviços para eliminação de vulnerabilidades conhecidas;
- Definir os firewalls;

- Definir os antivírus;
- Definir as ferramentas de rastreamento de vulnerabilidades (scanner);
- Definir os IDS.

A seguir estão detalhados cada componente:

3.1.1 Identificação das áreas de acesso

O gerenciamento da segurança do ambiente torna-se mais fácil ao se definir áreas de acesso, isto é, áreas com diferentes níveis de restrição de acesso, separadas por firewalls.

- Área não controlada:

Qualquer usuário conectado à Internet pode solicitar o serviço de um servidor Web, desta forma, não existe uma forma de controlar estas requisições.

- Área controlada:

Também denominada DMZ (DeMilitarized Zone), é a área em que os servidores de conteúdo público, como os servidores Web, FTP e de correio eletrônico, são instalados. O acesso a esta área é restrito por firewalls, permitindo somente requisições às portas em que os serviços estão operando.

Esta configuração oferece um nível maior de segurança aos servidores de acesso público, em relação à configuração demonstrada em [STE2002], denominada "ovelha para sacrifício", em que não existe *firewall* de primeiro nível.

- Área Restrita:

Os servidores de aplicação podem conter informações críticas, portanto não devem ser acessadas por qualquer usuário. Normalmente somente os servidores Web têm acesso a estes servidores. Esta área é denominada de acesso Restrito.

3.1.2 Autenticação e controle de acesso

Para *sites* de comércio eletrônico ou conteúdo personalizado, é importante um sistema de autenticação dos usuários. Este sistema irá previnir acessos não autorizados à informação e determinará quais privilégios o usuário autorizado tem sobre o conteúdo.

Existem diversos métodos de autenticação, alternativos ou suplementares, cada um com vantagens e desvantagens, que por não serem escopo do trabalho, não serão discutidos, somente mencionados. Os principais métodos são: senhas, token e biometria.

3.1.3 Separação dos serviços

Conforme [FRA1997], os serviços que um *site* irá fornecer, normalmente requerem diferentes níveis de acesso. Serviços essenciais à operação do *site* devem ser implementados em servidores dedicados com acesso limitado (somente ao serviço oferecido).

3.1.4 Identificação dos serviços desnecessários

Todos os serviços que não são disponibilizados pelo servidor devem ser desabilitados para evitar que os atacantes explorem possíveis vulnerabilidades que comprometam sua operação.

Disponibilizar os serviços críticos FTP e HTTP no mesmo servidor pode ter conseqüências catastróficas. Por exemplo, se um usuário enviar um programa hostil através do serviço FTP e fazer o servidor HTTP executá-lo.

3.1.5 Levantamento das correções dos serviços

A maioria dos ataques explora vulnerabilidades conhecidas dos serviços e para a maioria destas vulnerabilidades existe uma correção. Portanto é possível evitar que estes ataques se consolidem se todas as correções foram aplicadas.

3.1.6 Firewalls

Os firewalls são dispositivos que têm a função de separar redes. Um *firewall* pode ser um roteador, um computador, uma estação de trabalho, ou vários destes elementos combinados, que protegem uma rede, ou um segmento dela do acesso a determinados serviços e protocolos.

A Figura 3-2 apresenta a implementação de firewalls para a arquitetura de segurança de ambientes Web, com dois dispositivos separando as três areas de acesso: não controlada, controlada e restrita, pois cada uma possui requerimentos específicos, conforme mencionado anteriormente.

São dispositivos de grande importância na arquitetura de segurança de um ambiente Web, protegendo-o de uma variedade de tipos de ataques. No entanto, é importante compreender que são apenas parte da solução.

Para ataques que exploram vulnerabilidades do serviço Web pouco podem fazer, exceto em soluções coordenadas de IDS com capacidade de reconfiguração do *firewall*. Neste caso o IDS identificaria o atacante e poderia configurar o *firewall* para não permitir o acesso do atacante à rede.

3.1.7 Antivírus

O antivírus também é parte importante na arquitetura de segurança, pois é praticamente impossível um ambiente que esteja totalmente livre da infecção de programas hostis, conhecidos como vírus. Alguns destes vírus substituem programas verdadeiros e ao

serem executados instalam serviços clandestinos que permitem aos atacantes invadir o servidor infectado.

Existem diversos enfoques que podem ser utilizados para escolha do antivírus, porém independente da escolha é importante que o produto tenha possibilidade de integração ao sistema de detecção de intrusão do host.

3.1.8 Ferramentas de rastreamento de vulnerabilidades

As ferramentas de rastreamento de vulnerabilidades podem complementar a solução de segurança como uma facilidade para os administradores de segurança do ambiente Web.

Conhecendo as principais vulnerabilidades e formas de ataques aos servidores Web, estas ferramentas realizam uma série de testes em busca de pontos em que os sistemas possam ser alvo de ataques. Os principais produtos desta categoria são os gratuitos e muito eficientes Whisker e o Nikto – que foi utilizado para avaliar o IDS desenvolvido neste estudo, e os comerciais Stealh HTTP Scanner, AppScan.

3.1.9 IDS

Segundo [MCH2000], apesar da tecnologia ainda imatura e não poderem ser considerados soluções completas, os Sistemas de Detecção de Intrusão, assim como os firewalls, são componentes importantes da arquitetura de segurança de ambientes Web.

Os IDS de forma bastante simples, podem ser considerados como sistemas de alarme. Alarmes, por si só, não implementam segurança, somente indicam que algum tipo de atividade hostil está ocorrendo. Os fundamentos técnicos de IDS estão apresentados no capítulo 4.

4 Detecção de Intrusão

Antes de discutir sobre detecção de intrusão, é importante definir o que é intrusão. De acordo com [HEB1990], intrusão pode ser definida como qualquer conjunto de ações que tenham como intenção comprometer a integridade, confidencialidade, ou disponibilidade de um recurso.

Todas intrusões são definidas relativas a uma política de segurança e a menos que se saiba o que é e o que não é permitido no sistema, não há como tentar detectá-las.

Em linhas gerais, pois não é escopo deste trabalho, a política de segurança define o que é permitido e o que é negado em um sistema.

Existem duas filosofias que, embora sejam diametricamente opostas, servem de base para todas políticas de segurança:

- Proibitiva: tudo que não é declarado permitido é negado;
- Permissiva: tudo que não é declarado negado é permitido.

Estas filosofias podem ser misturadas para definir a política de segurança de uma empresa, aplicando-as a diferentes porções do ambiente, dependendo dos requerimentos funcionais, controle administrativo, etc.

Continuando com a definição dos conceitos de intrusão, quanto à categoria as intrusões podem ser divididas em duas principais classes:

- Mau-uso: são ataques bem definidos a pontos conhecidamente vulneráveis de um sistema. Podem ser detectados através da monitoração de certas ações executadas em determinados objetos;
- Anomalia: pode-se identificar um ataque através de observações de desvios em relação ao padrão normal de utilização do sistema (comportamento anômalo). O método para detecção deste tipo de intrusão é através da definição do padrão de utilização do sistema a ser monitorado, detectando significantes desvios a este padrão.

Intrusões por mau-uso seguem padrões bem-definidos que podem ser detectados através da busca pelo padrão nos registros de atividades do sistema. Como exemplo mais claro para este estudo, uma tentativa de execução de comandos através do servidor Web

pode ser detectada examinando as mensagens no arquivo de registro de atividades (arquivo de log).

O modelo clássico para detecção de intrusões por anomalia foi proposto por Denning em [DEN1987]. Na abordagem utilizada por Denning, um modelo é construído a partir de métricas derivadas da operação do sistema. Uma métrica é definida como uma variável qualquer *x* que representa uma medida quantitativa acumulada durante um período.

Estas métricas são contabilizadas a partir de parâmetros disponíveis no sistema como média de CPU, número de conexões de rede por minuto, número de processos por usuário, etc.

Uma anomalia pode ser um sintoma de uma possível intrusão. Dado um conjunto de métricas que podem definir um padrão de utilização normal do sistema, [DEN1987] assume que a exploração de vulnerabilidades de um sistema implica na sua utilização anormal; portanto, violações da segurança podem ser detectadas por padrões anormais de utilização do sistema.

A detecção de anomalias pode ser realizada por outros mecanismos, como redes neurais, técnicas de classificação aprendidas pela máquina (inteligência artificial), ou mesmo métodos que simulam o sistema imunológico dos seres vivos.

As intrusões anômalas são difíceis de se detectar. Não existem padrões para monitoração e uma abordagem mista deve ser utilizada. Idealmente um sistema deve combinar a capacidade humana de identificação de padrões com a rigorosidade de um programa de computador. Desta forma estaria sempre monitorando o sistema em busca de ataques em potencial, porém seria capaz de ignorar falsos-positivos resultantes de ações legítimas dos usuários.

4.1 Sistemas de Detecção de Intrusão

Sistema de Detecção de Intrusão – Intrusion Detection System (IDS) é uma ferramenta de segurança que tenta detectar e identificar um invasor acessando o sistema, ou um usuário legítimo fazendo mau-uso de seus recursos. O IDS é executado durante toda a operação do sistema, de forma transparente em segundo plano, somente notificando os administradores quando detecta alguma atividade suspeita ou ilegal. A Figura 4-1 ilustra a estrutura geral de um IDS:

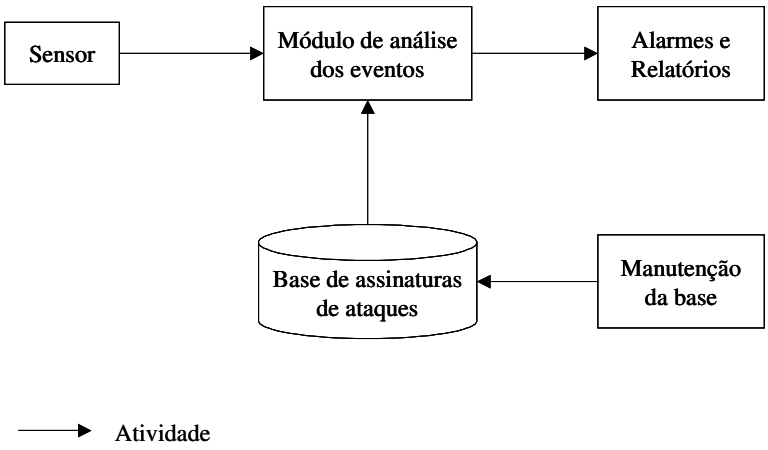


Figura 4-1- Estrutura geral de um IDS

A fonte de informações, que pode ser um segmento de rede, aplicação, ou recursos do sistema operacional, é monitorada pelo Sensor. Todas informações de atividades são enviadas pelo Sensor ao Módulo de análise de eventos, que por sua vez consulta a Base de Assinaturas de Ataques, tentando identificar algum padrão de hostilidade. Caso seja identificado como ataque, o Módulo de análise de eventos dispara um alarme, ou relatório sobre a ocorrência.

Um aspecto muito importante em um IDS é a manutenção da Base de Assinaturas de Ataques, pois conforme mencionado no capítulo anterior, novos ataques surgem a todos instantes. Mesmo que o IDS esteja preparado para suportar os mais recentes tipos de ataques, estará um passo atrás dos atacantes. No entanto, a manutenção da Base de Assinaturas de Ataques atualizada auxilia na redução desta diferença.

A fonte de informações é uma forma de classificar os IDS. Existem dois tipos básicos de IDS: baseados em host e baseados em rede. Apesar da estrutura ser a mesma, apresentam enfoques distintos na monitoração dos dados: os IDS baseados em host examinam os dados no computador, enquanto que os baseados em rede analisam os dados das conversações entre computadores. As principais características de cada tipo estão apresentadas a seguir:

4.1.1 IDS baseados em host

Foram os primeiros IDS a serem desenvolvidos e implementados. Estes sistemas coletam e analisam os dados que estão no computador que hospeda um serviço, como um servidor Web. Uma vez que estes dados são agregados para um computador, eles podem ser analisados localmente, ou enviados para uma central de análise.

De forma simplista, estes IDS são programas que operam em um computador recebendo informações dos arquivos de registros de atividades (logs) do sistema operacional, ou aplicações.

4.1.2 IDS baseados em rede

Este tipo de IDS analisa os pacotes de dados que circular pela rede. Estes pacotes são examinados e às vezes comparados com dados históricos para verificar sua natureza: hostil, ou inofensiva. Pelo fato de que são responsáveis por monitorar um segmento de rede e não um único computador, os IDS baseados em rede tendem a ser mais distribuídos que os baseados em host.

Independente do formato, software, ou hardware em alguns casos, eles podem estar instalados em em mais de um sistema conectado à rede e é utilizado para analisar dados contidos em unidades de transmissão da rede, comumente denominados pacotes.

Utilizam técnicas como “packet-sniffing”, que capturam todos os dados que trafegam no segmento de rede em que o sensor está instalado. São destinados principalmente para detectar as seguintes atividades:

- Acesso não autorizado: o IDS baseado em rede consegue identificar os pacotes à medida que a tentativa de acesso ocorre. Para casos em que o usuário consegue acesso, o IDS baseado em host é mais eficiente;
- Roubo de banda / negação de serviço: são ataques externos à rede que tentam abusar ou esgotar seus recursos. Os pacotes que iniciam, ou contém informações de ataque podem ser detectados pelos IDS baseados em rede.

4.2 Técnicas utilizadas para detecção

Conforme apresentado anteriormente, existem dois tipos de ataques: aqueles que provocam anomalias e os que fazem mau-uso dos recursos do sistema. As técnicas utilizadas para detecção tentam identificar estes dois tipos de ataques.

4.2.1 Detecção de anomalia

Desenvolvido para descobrir padrões anormais de comportamento, o IDS estabelece uma referência de padrões normais de utilização e qualquer comportamento que desvie muito desta referência é identificado como possível intrusão. O que é considerado comportamento normal varia para cada sistema, porém geralmente qualquer incidente que ocorra com freqüência maior que dois desvios da curva estatística deve ser analisado em maior detalhe.

4.2.2 Detecção de mau-uso ou assinatura

Mais conhecido como detecção por assinatura, este método utiliza padrões específicos e bem conhecidos de atividades não autorizadas para prever e detectar tentativas similares subseqüentes. Estes padrões específicos são denominados assinaturas.

Um exemplo para IDS baseados em host, é a seqüência de falhas de autenticação de um mesmo usuário. Para os IDS baseado em rede, uma assinatura pode ser tão simples quanto um padrão de um segmento de um pacote de dados.

A ocorrência de um evento que corresponda a uma assinatura pode não significar uma tentativa real de acesso não autorizado (falso positivo, que será tratado mais adiante), porém é recomendável analisar o alerta.

4.3 Questões que um IDS deve endereçar

Um sistema de detecção de intrusão deve endereçar as seguintes questões, independente da técnica que se baseia:

- Deve executar ininterruptamente sem supervisão humana. O sistema deve ser confiável o suficiente para ser executado em segundo plano no sistema monitorado;
- Deve ser tolerante a falhas de forma a resistir a um desastre no sistema não havendo a necessidade de recuperar toda sua base de conhecimento;
- Deve impor sobrecarga mínima ao sistema. Um IDS que torna o sistema excessivamente lento não deve ser usado;
- Deve observar desvios do comportamento padrão;
- Dever ser facilmente adaptado ao sistema monitorado. Cada sistema tem um padrão diferente de utilização e o mecanismo de defesa deve se adaptar facilmente a estes padrões;
- Deve acompanhar as mudanças de comportamento do sistema ao longo do tempo, a medida que novas aplicações são instaladas. O perfil do sistema vai mudar ao longo do tempo, e o IDS deve ser capaz de se adaptar;
- E finalmente, deve ser difícil enganá-lo.

O último ponto levanta uma questão sobre os tipos de problemas que muito provavelmente serão encontrados no sistema. Podem ser classificados como falsos positivos, falsos negativos, ou erros subversivos.

Um falso positivo acontece quando o sistema classifica uma ação como anômala, ou hostil (possível intrusão) quando na realidade é uma ação legítima.

A ocorrência freqüente de falsos positivos fará com que o IDS perca credibilidade e seus usuários ignorem os alarmes, pois considera ações legítimas como intrusões. Uma vez que não é possível eliminá-la totalmente, a ocorrência deste tipo de erro deve ser minimizada para fornecer informação correta e precisa aos operadores.

Falso negativo ocorre quando uma ação hostil aconteceu, mas o sistema considerou como ação legítima, não gerando alarme ao operador. Este tipo de erro é mais crítico que os falsos positivos, pois induzem a uma falsa sensação de segurança.

Erros subversivos acontecem quando um invasor modifica a operação do IDS para forçar a ocorrência de falsos negativos. É o tipo de erro mais complexo e trabalha em conjunto com os falsos negativos. Um invasor pode utilizar seu conhecimento sobre o IDS

para alterar sua operação, de forma a permitir que atividades hostis sejam literalmente desconsideradas e tratadas como atividades legítimas.

Pode-se descobrir a irregularidade na operação do IDS através da análise dos seus registros de atividades, ou ainda verificando a alteração de seus arquivos. No entanto, como não existem sintomas aparentes até que os ataques provoquem danos visíveis, o IDS continuará parecendo estar operando normalmente.

Uma forma de avaliar a eficiência do IDS em atender os requisitos acima descritos foi desenvolvida e publicada em [AMO1998] e utilizada para avaliar o IDS desenvolvido neste estudo.

Em última análise, de acordo com [MCH2000], apesar da tecnologia de detecção de intrusão ser imatura e não poder ser considerada como um sistema completo de defesa, os IDS podem ter uma função muito importante na arquitetura global de segurança.

5 Descritivo da solução

A abordagem para o tema escolhido, Gerenciamento de Segurança de Servidores Web utilizando SNMP foi centralizada no desenvolvimento de um Sistema de Detecção de Intrusão, especializado em servidores Web, baseado na arquitetura de gerenciamento de redes TCP/IP denominada SNMP.

Em linhas gerais, este Web IDS monitora em tempo real os arquivos de registros de atividades (logs) de servidores Web, em busca de indicadores de invasão, utilizando como base a estrutura do SNMP.

Toda a operação do agente é controlada através da manipulação de objetos da MIB genérica, especialmente desenvolvida para a solução. Todos os acessos considerados hostis serão relatados pelo agente ao gerenciador da rede através de Traps SNMP.

A base para o desenvolvimento da solução foi o agente Net-SNMP, cujas principais características técnicas são a portabilidade e extensibilidade. A distribuição gratuita incluindo o código fonte, torna o Net-SNMP ainda mais atraente para que novos módulos sejam desenvolvidos.

O IDS desenvolvido é tratado como uma extensão do agente, aproveitando o poderoso recurso do agente Net-SNMP.

Tomando-se como base a fonte monitorada, o IDS é classificado na categoria de Host IDS, com especialização em servidores Web.

No entanto, conforme mencionado no capítulo 2, dentro desta especialização existem diversos tipos de servidores para diversas plataformas. O objetivo de atender a parte desta grande diversidade veio de encontro ao recurso de portabilidade do Net-SNMP. Desta forma, foi possível desenvolver um código genérico, isto é, independente do servidor a ser gerenciado, para os módulos componentes do agente estendido (tratamento dos objetos da MIB, detecção e atualização automática) variando-se somente a base de ataques conhecidos.

Os servidores Web suportados inicialmente pelo IDS são aqueles que detêm maior parcela do mercado, conforme **Error! Reference source not found.**:

- Apache;
- Microsoft Internet Information Server;
- Zeus;
- SunONE.

Cada implementação de servidor Web pode registrar os acessos em arquivos com formatos diferentes definidos em sua configuração, sendo os principais:

- W3C Extended Log Format:

Formato proposto pelo W3C em [HAL1996], utilizado principalmente pelo Microsoft Internet Information Server, tem a seguinte estrutura:

```
date time c-ip cs-username s-ip cs-method cs-uri-stem cs-uri-query sc-status sc-bytes
cs-bytes time-taken cs-version cs(User-Agent) cs(Cookie) cs(Referrer)
```

Exemplificando:

```
1998-11-19 22:48:39 206.175.82.5 - 208.201.133.173 GET
/global/images/navlineboards.gif - 200 540 324 157 HTTP/1.0
Mozilla/4.0+(compatible;+MSIE+4.01;+Windows+95) USERID=CustomerA;+IMPID=01234
http://yourturn.rollingstone.com/webx?98@@webx1.html
```

Os campos que compõem o registro são:

date
Data em que a transação foi completada;

time
Horário em que a transação foi completada;

c-ip
Endereço IP do cliente;

cs-username
Identificador do usuário;

s-ip
Endereço IP do servidor;

cs-method
Método utilizado na requisição do usuário;

cs-uri-stem

URI solicitada pelo cliente;

cs-uri-query
Porção de pesquisa da URI;

sc-status
Código de estado do protocolo HTTP que o servidor apresenta ao cliente;

sc-bytes
Quantidade de Bytes transferidos pelo servidor;

cs-bytes
Quantidade de Bytes transferidos pelo cliente;

time-taken
Tempo decorrido para completar a transação em segundos;

cs-version
Versão do protocolo HTTP;

cs(User-Agent)
Navegador utilizado pelo cliente;

cs(Cookie)
Cookie apresentado pelo cliente;

cs(Referrer)
URL que fez referência à presente URL.

- NCSA Log Format / Common Log Format:

Formato suportado pela maioria dos servidores Web está especificado em [APA2003] e seus registros têm a estrutura conforme o exemplo abaixo:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
```

Detalhando cada componente dos registros:

127.0.0.1 (%h)
Endereço IP do cliente que fez a requisição ao servidor;

- (%l)
O hífen indica que a informação não está disponível. No caso, a informação que não está disponível é a identidade do cliente, definida na RFC 1413.

frank (%u)
É o identificador de usuário da pessoa que está requisitando o documento conforme determinado pela autenticação HTTP;

[10/Oct/2000:13:55:36 – 0700] (%t)
Tempo em que o servidor completou o processamento da requisição;

"GET /apache_pb.gif HTTP/1.0"
Requisição do cliente, composta pelo método, no caso, GET, em seguida pelo recurso, /apache.gif e finalizando, o protocolo: HTTP, versão 1.0;

200 (%>s)
Código de estado do protocolo HTTP que o servidor apresenta ao cliente. Especificado na RFC2616, é utilizado pelo IDS para verificar se o servidor aceitou a requisição.

2326 (%b)
O último campo indica o tamanho em Bytes do objeto retornado ao cliente.

O objeto wwwsecLogFormat da MIB pode assumir um dos dois formatos, que informará ao agente o formato de log a ser tratada, isto é, qual a posição dos campos significativos nos registros. Em ambos casos as seguintes informações são consideradas para o Módulo de Análise dos Eventos:

- Endereço IP do cliente: nem sempre existe a necessidade de autenticação para acesso ao conteúdo desejado. Desta forma, o campo de identificador do usuário nem sempre contém valor. O endereço IP do cliente sempre aparece nas mensagens, portanto é,

de forma geral, a melhor forma de identificar o atacante. Desconsideraram-se os casos de intrusão que utilizam técnicas de forjamento de endereço IP (IP spoofing).

- URI: informação utilizada para a detecção de intrusão. O agente irá comparar o conteúdo deste campo com todas as entradas do arquivo de assinaturas de ataques aplicáveis ao servidor em busca de um padrão que classifique o acesso como hostil.
- Código de estado do protocolo HTTP: informa se o servidor Web aceitou a solicitação do cliente.

De acordo com a definição de operação de um IDS, deve-se alarmar em qualquer tentativa. No entanto, para minimizar a ocorrência de falsos positivos, o agente pode trabalhar em dois modos quanto aos acessos monitorados:

- Alarme em qualquer tentativa. Neste modo, o IDS considera um ataque qualquer tentativa de acesso que explore alguma das vulnerabilidades específicas da implementação do servidor e que seja conhecida pelo IDS.
- Alarme somente em ataques efetivos, isto é, em acessos hostis cujas requisições foram aceitas pelo servidor.

O protocolo HTTP trabalha com mensagens de requisição e resposta. As requisições (sempre solicitando URIs) são feitas pelo usuário através do navegador (cliente), que as traduz em mensagens HTTP.

O servidor deverá responder com mensagens contendo a informação solicitada (conteúdo das URIs) sempre que possível. Segundo a RFC 2616 [FIE1999], que padroniza a versão 1.1 do protocolo HTTP, a resposta contém também um campo destinado ao código de estado do protocolo, que é um código de retorno composto por três dígitos da tentativa do servidor em compreender e satisfazer a requisição. O primeiro dígito define a classe da resposta e está apresentado na Tabela 5-1.

Tabela 5-1 - Códigos de estado do protocolo HTTP [FIE1999]

Código	Tipo de ocorrência	Descrição
1xx	Informativo	Requisição recebida, continuando o processo;
2xx	Sucesso	A solicitação foi recebida com sucesso, compreendida e aceita;
3xx	Redirecionamento	É necessário ação adicional para completar o processamento da requisição;
4xx	Erro do cliente	A requisição contém sintaxe errada e não pode ser atendida;
5xx	Erro do servidor	O servidor não conseguiu atender uma requisição aparentemente válida.

Operando neste modo, o IDS primeiramente verificará o código de estado de cada requisição no arquivo de registro de atividades do servidor em busca de códigos de classes 1xx até 3xx.

Ao retornar códigos destas classes representa que o servidor aceitou e processou as requisições, que podem ser inofensivas, como hostis (ataques). Respondendo a acessos hostis, indica que o servidor está vulnerável ao tipo específico de ataque.

Cada modo tem suas vantagens. O primeiro, por monitorar todo tipo de acesso, é menos seletivo, podendo identificar um invasor que não tenha informações sobre o servidor e por isso esteja tentando diversas formas de ataques. Pode ainda identificar um invasor com pouco conhecimento técnico e esteja operando uma aplicação que rastreia vulnerabilidades (scanner) do servidor. A grande desvantagem é que os alarmes serão disparados, mesmo o servidor não estando vulnerável ao ataque.

O segundo método implementa um nível maior de inteligência, somente alarmando os acessos hostis efetivamente aceitos pelo servidor. A desvantagem é que as tentativas não aceitas pelo servidor não serão alarmadas. A identificação de atacantes utilizando rastreadores (scanners) pode tornar-se inviável caso a base de ataques do rastreador seja inócua para o servidor.

O IDS inicialmente opera no segundo modo, mas pode ser alternada pelo administrador para o primeiro modo.

O código de todos os módulos, desenvolvidos nas linguagens Perl e C, estão apresentados no anexo C.

5.1 Definição da MIB

Toda a operação do Web-IDS é controlada através dos objetos da MIB desenvolvida neste estudo. Para alterar, ou consultar os valores devem-se utilizar operações SNMP, como *get* e *set*. A Figura 5-1 apresenta a sua estrutura.

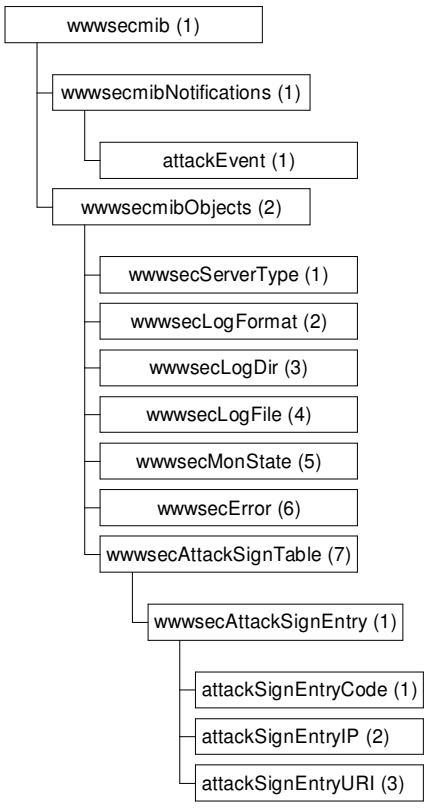


Figura 5-1 - Estrutura da MIB wwwsecmib

A Tabela 5-2 descreve todos os objetos da MIB e seu código, escrito em SMIv2, é apresentado no anexo B. Os OIDs dos objetos são relativos ao *enterprise* akiyama, sendo que para este estudo adotou-se o OID fictício .1.3.6.1.4.1.900.

Tabela 5-2 - Descrição dos objetos da MIB wwwsecmib

OID Relativo	Objeto	Função	Descrição
1.2.1	wwwsecServerType	Tipo do servidor	Especifica o tipo de servidor que será monitorado. Os seguintes servidores são suportados: - Apache; - Microsoft IIS; - SunOne; - Zeus.
1.2.2	wwwsecLogFormat	Tipo / formato da log	Especifica o formato da log a ser monitorada. Os seguintes formatos são suportados: - Common Log Format; - NCSA Extended.
1.2.3	wwwsecLogDir	Diretório da log	Identifica o diretório onde o servidor armazena a log.
1.2.4	wwwsecLogFile	Nome da log	Identifica o nome do arquivo de log.
1.2.5	wwwsecMonState	Estado do monitor	Controla o estado do monitor, ativando (valor 1), ou desativando (valor 0).
1.2.6	wwwsecError	Código de erro do agente	Informa a ocorrência de algum problema que comprometa o funcionamento do agente. O código é apresentado em formato decimal, representando um código de 2 bits, cujas funções são: - Bit 0: erro na localização do diretório; - Bit 1: erro na localização do arquivo.
1.2.7	wwwsecAttackSignTable	Tabela dos ataques	Tabela que registra todos os ataques ocorridos.
1.2.7.1	wwwsecAttackSignEntry	Registro da tabela	Identifica os registros dos ataques ocorridos.
1.2.7.1.1	attackSignEntryCode	Código do ataque	Contém o identificador do ataque.
1.2.7.1.2	attackSignEntryIP	Endereço IP	Armazena o IP do atacante.
1.2.7.1.3	attackSignEntryURI	URI	Contém o objeto do site solicitado pelo atacante.
1.1	attackEvent	Notificação de ataque	Além de armazenar na tabela, são enviados Traps ao gerenciador SNMP com os mesmos objetos dos seus registros.

5.2 Assinaturas de ataques

Conforme descrito no capítulo 4, existem abordagens para identificação de intrusão. O IDS desenvolvido baseia-se em assinaturas para identificar ataques. Todos os registros de acesso ao servidor são comparados os com padrões de ataques relacionados nos arquivos de assinaturas.

No entanto, sabe-se que sistema oferece uma solução definitiva, pois novos tipos de ataques são criados a todo instante. Assim sendo, existe um intervalo de tempo considerável para que os padrões de ataque sejam identificados, disponibilizados para conhecimento público e integrado às ferramentas de detecção. Isto torna a atualização freqüente da base de assinaturas de ataques extremamente importante para garantir a eficiência da ferramenta.

Existem instituições muito conceituadas como o CERT Coordination Center, da universidade Carnegie Mellon [CER2004] e o SecurityFocus [SEC2003], que realizam atividades de levantamento das vulnerabilidades em ambientes Web, alvo dos ataques. Estas vulnerabilidades são consolidadas em relatórios como o ICAT Metabase, do Computer Security Division, do NIST [NIS2003] e servem de base de assinaturas para as ferramentas de detecção de ataques e scanners.

A base de assinaturas de ataques do sistema de detecção de intrusão desenvolvido neste estudo é composta pela base do rastreador Nikto [CIR2004], que foi escolhido por sua grande abrangência tanto de tipos de servidores quanto de ataques reconhecidos e a freqüência com que é atualizada.

Para facilitar a manutenção e operação do agente, a base assinaturas de ataques foi dividida em quatro arquivos em formato texto, sendo um para cada tipo de servidor (Apache, Microsoft Internet Information Server, SunONE e Zeus). Cada ataque reconhecido pelo agente possui uma única entrada de uma linha no arquivo de assinaturas, sendo cada linha composta por três campos separados por vírgula “,“, conforme Tabela 5-3:

Tabela 5-3 - Descrição dos campos dos arquivos de assinaturas

Campo	Descrição
Índice	Número identificador do ataque e que é armazenado na tabela de ataques ocorridos;
Assinatura	Conjunto de caracteres, ou padrão que identifica unicamente um ataque;
Descrição	Descrição do ataque;
Tipo	Tipo do servidor Web que pode ser vulnerável ao ataque. Somente os servidores mais conhecidos no mercado foram considerados: Apache, Microsoft Internet Information Server, SunOne e Zeus.

Os ataques contidos no arquivo de assinaturas inicial estão relacionados no anexo C.

Os arquivos que compõem a base de assinaturas são:

- Apache: Web_ids_apache.db
- IIS: web_ids_iis.db
- SunONE: web_ids_sun.db
- Zeus: web_ids_zeus.db

5.3 Módulo de atualização automática de assinaturas de ataques

Conforme exposto inicialmente, a base de assinaturas deve ser atualizada freqüentemente para garantir o menor tempo possível de exposição aos novos ataques.

A atualização da base de assinaturas pode ser realizada de forma manual, ou automática e uma grande virtude do IDS é que não há a necessidade de desativar o agente, evitando que o serviço Web monitorado fique desassistido durante o processo de atualização.

A atualização manual dos arquivos é realizada incluindo-se uma linha para cada novo padrão reconhecido, de acordo com o formato descrito anteriormente.

O processo automático, similar ao disponibilizado pelos principais produtos do mercado, é executado em horários pré-determinados, buscando alterações na base de ataques do Nikto. Caso encontre alguma diferença entre a base anterior e a atual, o módulo de atualização automática salva a base atual a partir do *site* do Nikto, e separa os padrões para cada plataforma gravando os registros nos arquivos acima relacionados.

É importante salientar que é necessário que o módulo de atualização automática de assinaturas tenha acesso ao *site* do Nikto através do protocolo HTTP, operando na porta padrão, TCP 80.

O código do módulo de atualização automática de assinaturas está listado no anexo B.

É recomendado que seja utilizado preferencialmente o método automático, pois garante que os ataques detectados tenham sido analisados por instituições competentes. O método manual pode ser utilizado de forma complementar, ou em situações emergenciais.

Estas situações emergenciais são caracterizadas por descobertas de novos ataques, que representem real ameaça ao ambiente que o sistema de detecção de intrusão protege.

6 Testes de homologação

Para testar a eficiência do Web IDS desenvolvido neste estudo, foi utilizado o Nikto que é uma ferramenta bastante conhecida no mercado, tanto por sua forma de desenvolvimento e disponibilização, quanto por seu real valor técnico.

As qualidades técnicas do Nikto renderam-lhe uma posição de destaque na lista das 75 melhores ferramentas de segurança elaborada e publicada em [FYO2003].

O Nikto [CIR2004] é um scanner de servidores Web de filosofia Open Source (disponibilizado sob o formato GPL), que executa diversos testes em servidores Web, procurando por vulnerabilidades em execução de arquivos/CGIs potencialmente perigosos, além de problemas específicos das implementações dos servidores.

O ambiente de testes foi construído de acordo com a arquitetura típica de segurança de um ambiente Web, conforme apresentado no capítulo 3, com um servidor Web com uma página simples, estática, codificada em HTML, configurado para atender requisições na porta padrão (80) e um navegador como cliente para comprovar a acessibilidade do serviço.

Não foi implementado um *firewall* no ambiente, pois os testes foram direcionados somente à porta do serviço Web. Ataques direcionados a uma porta que oferece serviço não podem ser defendidos por *firewall*.

A estação com navegador (cliente Web) acumulou as funções de gerenciador SNMP e scanner Web. AFigura 6-1 ilustra a estrutura do ambiente de testes:

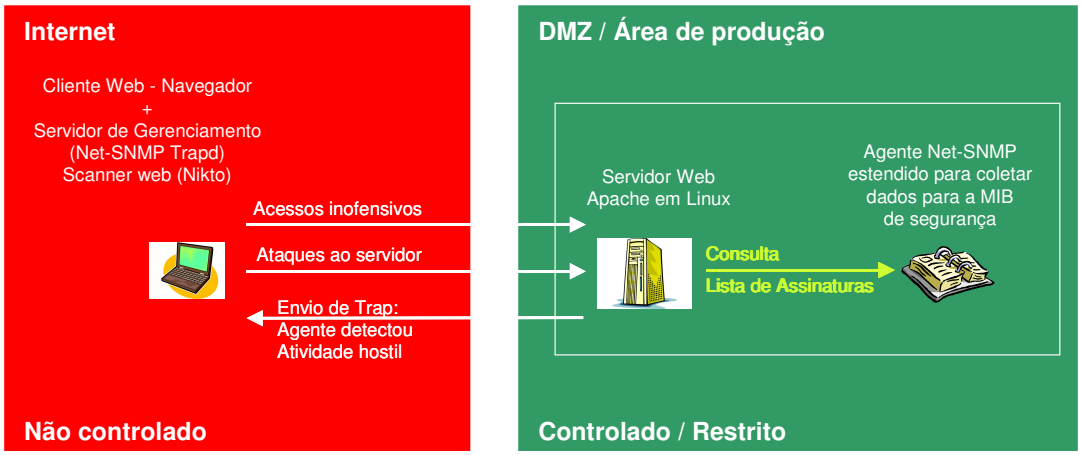


Figura 6-1 - Estrutura do ambiente de testes de homologação

6.1 Detalhamento dos componentes

Servidor Web:

- Sistema Operacional: SuSE Linux Enterprise Server 8, kernel 2.4.21-138;
- Serviço Web: Apache 2.0.52, a versão mais estável até outubro de 2004;
- Agente SNMP: net-snmp versão 5.1.2, disponibilizada em 09 de agosto de 2004.

Observação: Não foram testados outros servidores Web, nem plataformas suportadas, devido a limitação de tempo.

Procedimento utilizado para certificação da disponibilidade do servidor Web:

- Verificando se o processo httpd está em execução, conforme Figura 6-2:

```
dot:/usr/local/apache2/logs # ps -ef | grep httpd
root     22336     1  0 07:40 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   22337 22336  0 07:40 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   22338 22336  0 07:40 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   22339 22336  0 07:40 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   22340 22336  0 07:40 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   22341 22336  0 07:40 ?        00:00:00 /usr/local/apache2/bin/httpd -k start
nobody   22352 22336  0 07:41 ?        00:00:00 /usr/local/apache2/bin/httpd -k
```

Figura 6-2 - Verificação dos processos HTTP

- Verificando se o servidor está com a porta 80 (http) esperando por conexões, isto é, em estado LISTEN, conforme Figura 6-3:

```
dot:/usr/local/apache2/logs # netstat -napt | grep LISTEN | grep 80
tcp        0      0 :::80                   :::*                    LISTEN      22336/httpd
```

Figura 6-3 - Verificação do estado da porta 80

- Verificando se os arquivos de log, access_log e error_log foram criados no diretório /usr/local/apache2/logs. O arquivo access_log, que registra cada acesso ao servidor Apache, é utilizado pelo IDS como fonte de informações para identificação de acessos hostis. A Figura 6-4 mostra o diretório de logs:

```
dot:/usr/local/apache2/logs # ls -al
total 532
drwxr-xr-x    2 root     root          112 Oct 18 05:36 .
drwxr-xr-x   15 root     root          360 Oct  6 07:39 ..
-rw-r--r--    1 root     root       261939 Oct 18 05:22 access_log
-rw-r--r--    1 root     root       280034 Oct 18 05:36 error_log
```

Figura 6-4 - Arquivos de log do Apache

Cliente Web / Gerenciador SNMP / Scanner:

- Sistema Operacional: SuSE Linux Desktop 9.0, kernel 2.4.19;
- Cliente Web: Mozilla Firefox 1.0;
- Scanner de servidores Web: Nikto 1.34;
- Agente SNMP: net-snmp versão 5.1.2, disponibilizada em 09 de agosto de 2004;
- Gerenciador SNMP: snmptrapd, integrante do pacote Net-SNMP, do qual também faz parte o agente utilizado na solução.

A partir do navegador instalado no cliente, foi acessada a página do servidor Apache demonstrando que o serviço estava disponível aos clientes. A Figura 6-5 ilustra a página de testes acessada:

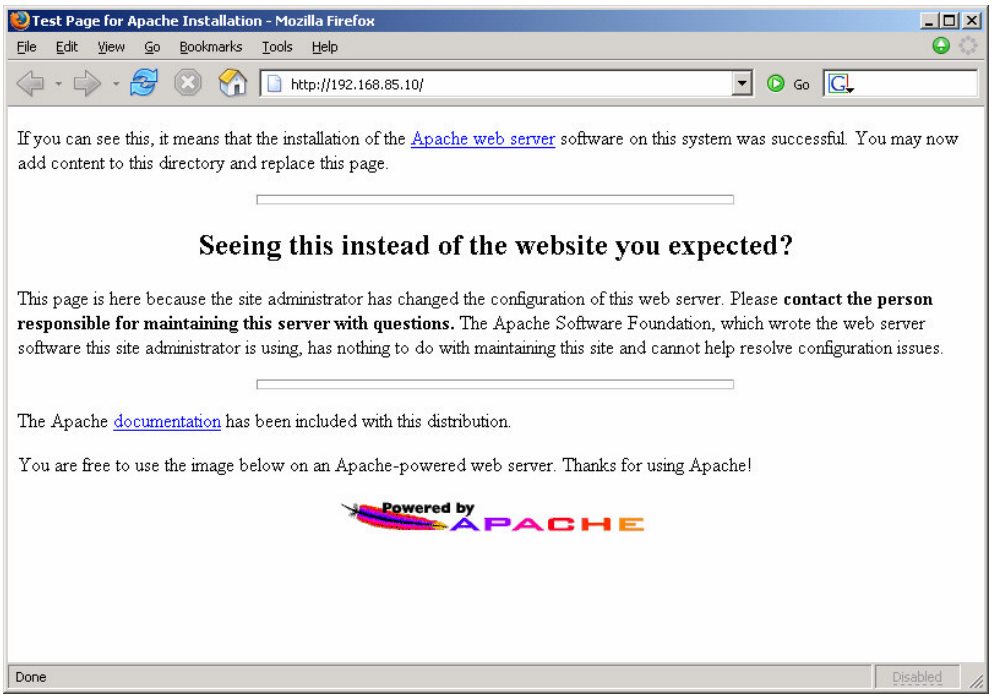


Figura 6-5 - Página de testes acessada pelo cliente

6.2 Detalhamento dos testes

Os testes consistiram na execução do scanner Nikto, que comandou uma série de testes que exploram as vulnerabilidades do servidor implementado, simulando as requisições utilizadas em ataques reais.

Cada requisição do Nikto foi registrada no arquivo access_log do servidor Apache. O Web IDS, em tempo real, analisou cada registro de acordo com o formato definido pelo objeto da MIB wwwsecLogFormat, comparando a URI com sua base de ataques para servidores Apache.

O Nikto foi configurado para apresentar um arquivo contendo o resultado dos testes. O conteúdo do arquivo está apresentado na Figura 6-6.

```
- Nikto v1.34/1.29
---------------------------------------------------------------------------
+ Target IP:       127.0.0.1
+ Target Hostname: localhost
+ Target Port:     80
+ Start Time:      Thu Jan  6 16:16:28 2005
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache/2.0.52 (Unix)
- Server did not understand HTTP 1.1, switching to HTTP 1.0
+ Server does not respond with '404' for error messages (uses '400').
+     This may increase false-positives.
+ IIS may reveal its internal IP in the Content-Location header. The value is
"index.html.en". http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0649.
+ Allowed HTTP Methods: GET,HEAD,POST,OPTIONS,TRACE
+ HTTP method 'TRACE' is typically only used for debugging. It should be disabled.
+ 2.0.52 (Unix) - TelCondex Simpleserver 2.13.31027 Build 3289 and below allow directory
traversal with '/.../' entries.
+ /~root - Enumeration of users is possible by requesting ~username (responds with
Forbidden for real users, not found for non-existent users) (GET).
+ / - Appears to be a default Apache install. (GET)
+ / - Appears to be a default Apache install. (GET)
+ /icons/ - Directory indexing is enabled, it should only be enabled for specific
directories (if required). If indexing is not used all, the /icons directory should be
removed. (GET)
+ /index.html.de - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.dk - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.en - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.es - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.et - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.fr - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.pt - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.pt-br - Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information. (GET)
+ /index.html.var - Apache default foreign language file found. All default files should
be removed from the web server as they may give an attacker additional system
information. (GET)
+ /manual/images/ - Apache 2.0 directory indexing is enabled, it should only be enabled
for specific directories (if required). Apache's manual should be removed and directory
indexing disabled. (GET)
+ / - TRACE option appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACE)
+ /manual/ - Web server manual? tsk tsk. (GET)

+ Over 30 "OK" messages, this may be a by-product of the
           +     server answering all requests with a "200 OK" message. You should
           +     manually verify your results.
+ 2449 items checked - 33 item(s) found on remote host(s)
+ End Time:        Thu Jan  6 16:19:18 2005 (170 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

Test Options: -h localhost -output apache2.report
---------------------------------------------------------------------------
```

Figura 6-6 - Trecho do relatório emitido pelo Nikto

Informações registradas pelo IDS:

Os ataques identificados pelo IDS são informados ao gerenciador SNMP através de mensagens do tipo Trap. Conforme definido na estrutura do ambiente de testes, o próprio servidor acumulou a função de gerenciador SNMP.

De acordo com a definição da MIB WWWSEC, os Traps contêm as seguintes informações:

- Código do ataque;
- IP do invasor;
- URI acessado.

Os Traps recebidos durante os testes foram registrados, de acordo com a configuração do snmptrapd, no arquivo trapd.log. Parte do conteúdo do arquivo, resultante dos testes está apresentado na Figura 6-7.

Como os Traps informam apenas o Código do ataque e não seu significado, foi habilitado durante estes testes o registro das informações do ataque no arquivo contataque.log. Parte do arquivo está apresentado na Figura 6-8.

```
2005-01-06 16:11:12 NET-SNMP version 5.1.2 Started.
2005-01-06 16:16:31 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:04.27
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 8       WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /?S=A
2005-01-06 16:16:31 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:04.65
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 33      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /icons/
2005-01-06 16:16:31 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:04.76
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 34      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.ca
2005-01-06 16:16:31 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:04.98
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 36      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.de
2005-01-06 16:16:31 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:05.08
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 37      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.dk
2005-01-06 16:16:32 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:05.19
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 38      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.ee
2005-01-06 16:16:32 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:05.30
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 39      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.el
2005-01-06 16:16:32 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:05.42
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 40      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.en

2005-01-06 16:16:33 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:06.76
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 55      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.pt
2005-01-06 16:16:33 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:06.87
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 55      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.pt-br
2005-01-06 16:16:33 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:06.96
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 56      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.pt-br
2005-01-06 16:16:34 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:07.64
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 65      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /index.html.var
2005-01-06 16:16:34 localhost [127.0.0.1] (via 127.0.0.1) TRAP, SNMP v1, community public
       WWWSEC-MIB::wwwsecmibNotifications Enterprise Specific Trap (1) Uptime: 4:18:07.79
       WWWSEC-MIB::attackSignEntryCode.0 = INTEGER: 70      WWWSEC-
MIB::attackSignEntryIP.0 = IpAddress: 127.0.0.1      WWWSEC-MIB::attackSignEntryURI.0 =
STRING: /manual/images/
```

Figura 6-7 - Trecho do arquivo trapd.log

```
//, 200, 3, Apache on Red Hat Linux release 9 reveals the root directory listing by
default if there is no index page.

//, 200, 3, Apache on Red Hat Linux release 9 reveals the root directory listing by
default if there is no index page.

/?D=A, 200, 5, Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.

/?M=A, 200, 6, Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.

/?N=D, 200, 7, Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.

/?S=A, 200, 8, Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.

/icons/, 200, 33, Directory indexing is enabled it should only be enabled for specific
directories (if required). If indexing is not used all the /icons directory should be
removed.

/index.html.ca, 200, 34, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.

/index.html.cz.iso8859-2, 200, 35, Apache default foreign language file found. All
default files should be removed from the web server as they may give an attacker
additional system information.

/index.html.de, 200, 36, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.

/index.html.dk, 200, 37, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.

/index.html.ee, 200, 38, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.


/index.html.pt-br, 200, 55, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.

/index.html.pt-br, 200, 56, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.

/index.html.ru.utf8, 200, 61, Apache default foreign language file found. All default
files should be removed from the web server as they may give an attacker additional
system information.

/index.html.var, 200, 65, Apache default foreign language file found. All default files
should be removed from the web server as they may give an attacker additional system
information.

/manual/images/, 200, 70, Apache 2.0 directory indexing is enabled it should only be
enabled for specific directories (if required). Apache's manual should be removed and
directory indexing disabled.
```

Figura 6-8 - Trecho do arquivo contataque.log

6.3 Avaliação dos resultados apresentados pelo IDS

Os testes realizados forneceram dados que podem ser utilizados para avaliar dois importantes aspectos técnicos do IDS desenvolvido:

a) Eficiência na identificação de ataques

O relatório apresentado pelo Nikto informa que foram identificadas 35 vulnerabilidades.

O arquivo trapd.log, que registrou os Traps enviados pelo IDS, contabilizou 35 ataques identificados.

Apesar da quantidade de ataques reportados pelo scanner e pelo IDS serem a mesma, existem as seguintes diferenças:

- Ataques não identificados pelo IDS: 3;
- Ataques identificados pelo IDS que não ocorreram (falsos-positivos): 3.

Considerando-se essas deficiências e limitações do IDS desenvolvido neste estudo e a partir dos valores resultantes dos testes, pode-se calcular o índice de eficiência em identificação através da seguinte fórmula:

$I = 1 – F / A$

Onde,

I é o índice de eficiência, em porcentagem;
F é a quantidade de falhas, ou seja, quantidade de ataques não identificados somada à quantidade de falsos-positivos;
A é a quantidade total de ataques.

Desta forma, para os testes realizados,

$I = 1 – 6 / 35 \Leftrightarrow I = 82{,}86\ \%$

b) Intervalo de tempo para notificação ao gerenciador

Pelo fato do IDS analisar cada registro de acesso aos recursos do servidor Web à medida que ocorrem, pode-se dizer que seu funcionamento ocorre em tempo real.

Apesar do algoritmo de identificação ter sido desenvolvido para apresentar menor latência possível, existem diversas otimizações e melhorias a realizar. Pode-se verificar que a diferença entre os tempos do final dos acessos realizados pelo Nikto e do recebimento do último Trap registrado no arquivo trapd.log, considerando-se as limitações do IDS e o volume de acessos do Nikto, que exigiu um processamento bastante rápido, é satisfatoriamente pequeno.

7 Avaliação do Web IDS conforme guia do IEEE

Conforme afirma [MCH2000], para se implementar uma solução de IDS deve-se considerar o ambiente a ser gerenciado, os recursos disponíveis para a operação e manutenção e escolher aquele que seja mais adequado.

Apesar de todo o desenvolvimento no segmento de segurança de ambientes Web, ainda não existe padrão de indústria para avaliação da eficiência de um IDS, mesmo porque não é uma tarefa trivial. Desta forma, pode-se dizer que são bastante subjetivos os resultados apresentados pelos fabricantes e até mesmo por laboratórios independentes que se propõem a avaliá-los.

Edward Amoroso e Richard Kwapniewski elaboraram um guia para auxiliar a seleção de um IDS, publicado em [AMO1998]. Os critérios apresentados no guia serão utilizados para classificar o IDS desenvolvido neste estudo. O conjunto de critérios está dividido em três categorias:

- Detecção. Este requerimento trata de questões funcionais relacionadas à detecção de intrusão;
- Resposta. Este requerimento trata de atividades funcionais e processuais iniciadas em resposta a detecção;
- Implementação. Trata de questões práticas da instalação, plataforma suportada, integração, interoperabilidade e outras áreas.

Os produtos são classificados em três níveis de atendimento ao requerimento em cada categoria. A Classe A representa total atendimento ao requerimento, Classe B representa um atendimento diferenciado e Classe C outra forma. A Figura 7-1 apresenta graficamente os níveis de classificação.

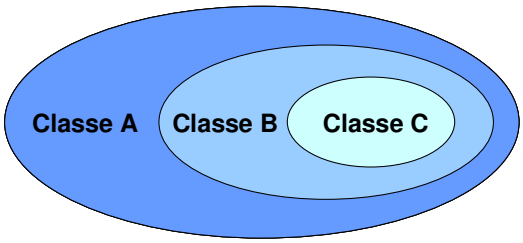


Figura 7-1 - Os níveis de classificação de atendimento aos requerimentos

Pode-se dizer que dada uma categoria, a Classe A inclui todos os requerimentos da Classe B, que por sua vez inclui todos os requerimentos da Classe C. [AMO1998] alerta que os sistemas que atendem a classes menos abrangentes (C, ou mesmo B) não devem ser considerados inferiores aos sistemas que atendem a classes mais completas (A, ou ainda B), pois esta avaliação depende das necessidades de segurança e possibilidades financeiras para implementação da solução.

As categorias serão brevemente descritas, de acordo com o guia, porém adaptadas para as características dos Web IDS, juntamente com a classificação do IDS desenvolvido neste estudo.

7.1 Requerimentos de detecção

Os requerimentos de detecção classificam o grau de eficiência com que o IDS consegue detectar intrusões reais. São quatro tipos de requerimentos nesta categoria:

- D1: Tempo para detectar intrusões;
- D2: Base de conhecimento sobre intrusões;
- D3: Utilização de perfis para usuários e sistemas;
- D4: Correlacionamento de várias fontes de informação.

7.1.1 D1: Tempo para detectar intrusões

Neste requerimento procura-se avaliar o tempo decorrido entre um evento de intrusão e sua detecção automática.

Classe C: o sistema possui capacidade para eventualmente detectar intrusões, provavelmente depois do ataque acontecer. As ferramentas de análise das informações com o sistema protegido fora de funcionamento pertencem a esta classe.

Classe B: os IDS apresentam respostas em tempo próximo ao real (podem ser em questão de minutos ou horas). A técnica típica para este tipo de IDS é a análise das informações à procura de padrões de ataques com o sistema protegido em funcionamento.

O Sistema Desenvolvido neste estudo pertence a esta classe, pois utiliza a técnica descrita.

Classe A: estes sistemas possuem recursos especiais que minimizam o tempo de detecção. Estes recursos são geralmente implementados através de hardware especializado que garantem a detecção em tempo real, na ordem de poucos segundos.

7.1.2 D2: Base de conhecimento sobre intrusões

O nível de detalhes da base de conhecimento de intrusões utilizadas no produto são endereçados por este requerimento.

Classe C: estes sistemas incluem um conjunto de padrões e perfis de ataques fornecidos pelo desenvolvedor que são atualizados periodicamente. Quase todos os IDS do mercado atendem a este requisito.

Classe B: os sistemas desta classe permitem que o administrador customize os padrões (inclusão, alteração).

O IDS desenvolvido neste estudo pertence a esta classe, pois permite a edição dos arquivos que compõem a base de assinaturas.

Classe A: disponibilizam uma API, ou linguagem para que os usuários possam incluir novos padrões de ataques.

7.1.3 D3: Utilização de perfis para usuários e sistemas

O perfil é uma forma de representar um comportamento esperado. A possibilidade de criar perfis de comportamento dos usuários e sistemas para servir de base para detecção de atividades anômalas é um recurso interessante. No entanto não é uma tecnologia madura implementada nos sistemas de detecção de intrusão. Desta forma não existe requerimentos para a Classe C.

O IDS desenvolvido não oferece recursos para manipulação de perfis de usuários e sistemas, portanto pertence à Classe C.

Classe B: o mecanismo de tratamento de perfis existe para especificar e capturar informações estatísticas sobre o sistema em funcionamento.

Não existe Classe A para este requerimento.

7.1.4 D4: Correlacionamento de várias fontes de informação

Este requerimento trata da habilidade de um IDS relacionar indicações de diversas fontes em diferentes momentos.

Classe C: os sistemas desta classe permitem que diversas fontes de informação sirvam de fonte para uma central de processamento.

Classe B: os sistemas desta classe devem incluir mecanismos para permitir a definição e procurar padrões de atividades na fonte de informações.

Classe A: os sistemas classe A incluem mecanismos automáticos para incorporação de informações de várias fontes.

O IDS desenvolvido neste estudo não atende a este requerimento em nenhuma classe, pois utiliza somente o registro de acessos ao servidor Web como fonte de informações.

7.2 Requerimentos de Resposta

Os requerimentos de resposta classificam o nível de resposta de um IDS à atividade hostil. Quatro tipos de requerimento compõem esta categoria:

- R1: Tipos de diretivas de resposta
- R2: Registro de informações
- R3: Identificação do invasor
- R4: Armadilhas para o invasor

7.2.1 R1: Tipos de diretivas de resposta

A maioria das ferramentas de defesa inclui algum tipo de alarme. Considera-se alarme alguma diretiva de resposta do sistema protegido à ferramenta de segurança.

Classe C: os sistemas desta classe implementam mecanismos de alarme estático, isto é, informam sobre um evento suspeito através de mensagens apresentadas em console.

Classe B: os sistemas classe B incrementam os mecanismos da classe C com algum método de envio de mensagem ao administrador, através de e-mail, por exemplo. Mecanismos de alteração da configuração do sistema protegido em resposta ao evento hostil são desejáveis nesta classe. Por exemplo, como resposta a um ataque de negação de serviço, o IDS poderia desabilitar o serviço HTTP.

O IDS desenvolvido neste estudo atende à Classe B, pois utiliza a console de eventos de um sistema de gerenciamento SNMP, que pode estar remota, para enviar alertas (trap SNMP) sobre eventos hostis. No ambiente de homologação, a console de eventos do sistema de gerenciamento foi representado pelo daemon trapd, cujas mensagens recebidas são registradas no arquivo trapd.log.

Classe A: os sistemas desta classe devem incluir um mecanismo automático de resposta, customizável através de uma API. O propósito deste requerimento é a possibilidade de integração do IDS a um sistema mais complexo de recuperação de desastre.

7.2.2 R2: Registro de informações

Este requerimento avalia a forma que a informação é registrada e armazenada e protegida pelo IDS. Não existem requerimentos para a Classe C, uma vez que existem ambientes onde as ferramentas de defesa já fazem as verificações e executam ações em tempo real.

O IDS desenvolvido pertence a esta classe, pois executa as verificações em tempo real, apesar de armazenar as informações referentes aos ataques na tabela da MIB.

Classe B: em sistemas Classe B são gerado registros detalhando a atividade hostil, que podem servir de base para análises futuras.

Classe A: os sistemas Classe A incluem ferramentas para processar os registros dos ataques.

7.2.3 R3: Identificação do invasor

Este requerimento descreve o nível de suporte a investigação da identidade do invasor. As classes C e B não têm requerimento para identificação do invasor.

Classe A: sistemas desta classe utilizam ferramentas apropriadas para investigação. Estes sistemas também devem ter habilidade de correlacionar informações de outros sistemas para identificar o invasor.

O IDS desenvolvido atende à classe A, pois a partir dos registros de invasão, consegue identificar o invasor pelo seu endereço IP e, se disponível, a sua identidade. Considera-se que o invasor não esteja forjando sua identidade com técnicas como IP Spoofing.

7.2.4 R4: Armadilhas para o invasor

Indica a forma que os sistemas disponibilizam de armadilha para atrair invasores. Os sistemas que atendem às classes C e B não têm requerimentos de armadilhas. Isto se deve ao fato de quase não existir sistemas que utilizam armadilhas.

O IDS desenvolvido não apresenta qualquer forma de armadilha, portanto pertence às classes C e B

Em sistemas Classe A, os invasores são, de forma imperceptível, direcionados a armadilhas que extraem o máximo de informação do invasor, ao mesmo tempo que evitam que este consiga cumprir seu objetivo.

7.3 Requerimentos Operacionais

Tratam questões referentes à instalação, uso e suporte do IDS em um ambiente real, em produção. Esta categoria é composta por quatro requerimentos:

- O1: Testes e verificações
- O2: Segurança do sistema
- O3: Plataformas suportadas. Este requerimento foi adaptado do guia para os IDS baseados em host. Especificamente para este estudo, os Web IDS.
- O4: Interface com o usuário

7.3.1 O1: Testes e verificações

Classificam os IDS quanto ao nível de testes a que foram submetidos, muito importantes para a disponibilização de um produto no mercado.

Classe C: estes sistemas não possuem nenhum requerimento de testes. Considerou-se verdadeira a necessidade de rápida disponibilização dos produtos, exigindo o rápido desenvolvimento, impossibilitando que planos extensos de testes sejam aplicados.

Classe B: os produtores destes sistemas fornecem evidências da execução de um plano de testes, que devem ser o mais extenso e exaustivo possível.

O IDS desenvolvido neste estudo pertence a classes B, pois os testes de eficiência, apesar de realizados em ambiente de testes, simularam diversas situações de ataques reais e os resultados obtidos demonstram a capacidade do IDS em detectá-las.

Classe A: os produtores destes sistemas devem fornecer relatórios de testes realizados por instituições bem conceituadas.

7.3.2 O2: Segurança do sistema

Classificam os IDS quanto ao grau de segurança com que os IDS protegem a si próprios de ataques. Os IDS são elementos importantes na arquitetura de segurança e podem ser alvo de ataques, pois uma vez desativados, a tarefa dos invasores torna-se mais fácil.

Classe C: os sistemas desta classe implementam alguma forma de segurança, como controle de acesso, níveis de auditoria e autenticação do usuário.

Classe B: os sistemas que atendem aos requerimentos da classe B devem implementar alguma forma de criptografia da comunicação com o sistema a ser protegido.

O IDS desenvolvido neste estudo é compatível com SNMP v2 e 3. Apesar dos testes realizados terem utilizado SNMP v2, que utiliza autenticação simples entre o agente do sistema protegido e o gerenciador através de nomes de comunidades, o SNMP v3 pode ser utilizado, o que oferece tanto autenticação forte entre os parceiros, quanto confidencialidade através de criptografia das informações.

Classe A: devem implementar segurança através de hardware específico.

7.3.3 O3: Plataformas suportadas

Estes requerimentos classificam o IDS quanto à abrangência da solução.

Classe C: os sistemas desta classe suportam somente uma plataforma. No caso dos Web IDS, somente um tipo de servidor. Por exemplo, pode oferecer suporte ao Apache, ou Microsoft Internet Information Server; não os dois.

Classe B: os sistemas classe B devem suportar várias plataformas, que podem ser detectadas automaticamente pelo IDS, ou especificadas pelo administrador durante sua inicialização.

O IDS desenvolvido neste estudo suporta várias plataformas, conforme mencionado nos capítulos anteriores, portanto pertence à classe B.

Classe A: os sistemas classe A devem permitir que o administrador estenda o suporte do IDS a novas plataformas.

7.3.4 O4: Interface com o usuário

Este requerimento classifica o IDS quanto ao tipo de interface oferecida ao administrador para sua configuração e operação.

Classe C: estes sistemas não oferecem interface formal, contam apenas com recursos oferecidos nativamente pelo sistema operacional em que o IDS encontra-se instalado. Estes recursos podem ser a operação do IDS através da edição de arquivos de controle.

Classe B: os sistemas classe B disponibilizam uma interface dedicada através de linha de comando, ou aplicação GUI.

Classe A: os sistemas classe A disponibilizam uma aplicação GUI centralizada, a partir da qual um administrador pode controlar remotamente e operacionalizar qualquer componente do IDS.

O IDS desenvolvido neste estudo utiliza a estrutura do SNMP para controle e operacionalização. Pode-se controlar, desta forma, o IDS a partir de um Sistema de Gerenciamento de Rede centralizado com interface gráfica. Atende, desta forma, os requerimentos da classe A.

7.4 Sumário da avaliação

A Tabela 7-1 apresenta a classificação do IDS desenvolvido neste estudo para cada uma das categorias descritas.

Tabela 7-1 - Matriz de classificação do IDS desenvolvido no estudo

Requerimento		Classificação
Detecção	D1: Tempo de detecção	Classe B: Tempo próximo ao real
	D2: Base de conhecimento	Classe B: Permite customização
	D3: Utilização de perfis	Classe C: Não utilize
	D4: Correlacionamento	Não utilize outras fontes
Resposta	R1: Diretivas de resposta	Classe B: Envio de Traps SNMP
	R2: Registro de informações	Classe B: Armazena em tabela o histórico de ataques
	R3: Identificação do invasor	Classe A: Através do endereço IP
	R4: Armadilhas	Classes B e C: Recurso não implementado
Implementação	O1: Testes e verificações	Classe B: Relatório com resultados dos testes
	O2: Segurança do sistema	Classe B: Permite utilizar recursos do SNMPv3 de autenticação e criptografia
	O3: Plataformas suportadas	Classe B: Suporta os principais servidores Web e sistemas operacionais
	O4: Interface com o usuário	Classe A: Através da console do Sistema de Gerenciamento da Rede, como NetView e OpenView

O resultado numérico (adotando-se C=1, B=2 e A=3, por exemplo) a partir da classificação através dos critérios apresentados no guia pode ser bastante útil para a escolha do IDS e [AMO1998] apresenta alguns métodos para obtê-lo:

- Soma simples: somando-se o valor de atendimento para cada um dos requerimentos em cada uma das categorias apresentadas é um método que apresenta como grande vantagem a simplicidade. No entanto as desvantagens são definitivamente contrárias ao método, pois atribui equivalência entre todos requerimentos. Ignora-se o fato de que cada requerimento pode contribuir de forma decisiva para atender o orçamento destinado ao IDS e as necessidades de segurança.

- Soma ponderada: atribui pesos diferenciados entre os requerimentos. Apesar de dificultar o processo, resulta em avaliações mais adequadas para cada situação. Um mesmo IDS pode receber notas diferentes para ambientes diferentes.

- Vetor de requerimentos: este método exige consideração não somente do ambiente existente, mas também (e decisivamente) dos planos para o futuro. Por exemplo, caso o ambiente possua somente atualmente servidores Microsoft Internet Information Server, porém têm planos de migrar para servidores Apache, é importante que o Web IDS suporte esta plataforma.

Recomenda-se que a avaliação não seja feita através da simples somatória da pontuação, mas sim considerando a real necessidade em cada um dos requerimentos. Desta forma, o método de Vetor de Requerimentos, apesar de ser o mais complexo e também demorado, apresentará certamente o Web IDS mais adequado ao ambiente.

Não foi possível realizar a comparação com produtos disponíveis no mercado devido a indisponibilidade de tempo. No entanto, pode-se notar que o Web IDS desenvolvido neste estudo atende de alguma forma a todos os requerimentos apresentados no guia.

8 Conclusões e o futuro deste trabalho

8.1 Conclusões

A abordagem utilizada para o tema escolhido, Gerenciamento de Segurança de Servidores Web utilizando SNMP foi centralizada no desenvolvimento de um Sistema de Detecção de Intrusão, ou IDS (Intrusion Detection System) de código aberto especializado em servidores Web, baseado na arquitetura de gerenciamento de redes TCP/IP denominada SNMP.

A utilização do SNMP permite a integração com qualquer sistema de gerenciamento de redes de forma nativa, o que o diferencia das demais soluções existentes.

A base para do desenvolvimento foi o poderoso agente Net-SNMP, distribuído gratuitamente, que tem como principais características técnicas a expansibilidade e portabilidade. Através destes recursos foi possível elaborar uma solução que suporte inicialmente os principais servidores Web e sistemas operacionais do mercado.

O desenvolvimento do IDS compreendeu a definição completa da solução, implementação e testes de todos os seus componentes:

- Agente: componente que monitora e alerta em tempo real as atividades do servidor Web. A solução suporta atualmente os principais servidores Web disponíveis no mercado (Apache, Microsoft Internet Information Server, SunOne e Zeus), podendo ser estendida no futuro a outros servidores;

- MIB: base de dados genérica para segurança de servidores Web. Os objetos que a compõem permitem controlar a operação do IDS: o formato e localização da log analisada, o tipo de servidor e o próprio estado do agente;

- Módulo de atualização automática: permite a atualização de sua base de informações de ataques automaticamente, em horários agendados, a partir de um serviço na Internet e sem interromper o funcionamento do agente. Este recurso evita que o servidor Web fique sem proteção durante o período de atualização.

A homologação da ferramenta foi realizada em um ambiente simulado com servidores Apache sobre a plataforma Linux atacado pelo rastreador de vulnerabilidades Nikto. Os resultados mostraram que, apesar de alarmar alguns falsos positivos, é boa a sua confiabilidade.

Os critérios apresentados no guia para auxiliar a escolha de um IDS, publicado em [AMO1998] foram utilizados para classificar o IDS desenvolvido. O conjunto de critérios divide-se em três categorias de requerimentos: detecção, resposta e implementação. Para cada requerimento o produto é classificado em três níveis de atendimento.

Os resultados obtidos mostraram que o IDS desenvolvido atende de alguma forma a todos os requerimentos apresentados no guia, credenciando a ferramenta através critérios aceitos internacionalmente.

No entanto, é importante lembrar que o IDS é apenas um componente de uma estrutura de defesa para ambientes Web. E a qualidade da segurança oferecida ao ambiente está diretamente relacionada à forma como a defesa está estruturada. Recomenda-se que esta estrutura seja baseada na postura de defesa denominada Defesa Intensa.

Para finalizar, acompanhando [STE1999], pode-se afirmar que proteger sistemas de informação, como um ambiente Web, é semelhante a proteger uma residência: se alguém está decidido a invadi-la, é quase que impossível impedir que isto ocorra, porém pode-se torná-lo bastante difícil. E isso pode ser o suficiente para fazer o invasor mudar de idéia e procurar uma vítima mais fácil.

8.2 Contribuições desta dissertação

A contribuição mais significativa é a apresentação de uma eficiente ferramenta para detecção de intrusão em servidores Web, de baixo custo operacional, com suporte a diversas plataformas.

A expansibilidade é uma característica muito importante, pois através da flexibilidade do Net-SNMP, juntamente com o código aberto e estruturado do IDS desenvolvido, permitem com pouca modificação, a inclusão do suporte a outros servidores Web.

Outro aspecto que deve ser mencionado é a pronta integração com qualquer software de gerenciamento de redes, pelo fato de usar a estrutura SNMP.

Existem diversos IDS no mercado, tanto comerciais, quanto gratuitos, no entanto cada um opera de sua própria forma, com suas próprias linguagens de customização e capacidade de integração com consoles de eventos de outros fabricantes bastante restrita, quando isto é possível.

8.3 Futuro para este trabalho

A finalização da dissertação certamente não significará o término do desenvolvimento da ferramenta. Muito pelo contrário. Existem diversas imperfeições e funcionalidades que podem ser implementadas. Os próximos passos para o desenvolvimento da ferramenta podem ser assim relacionados:

- Incluir o projeto no SourceForge.net, que é o maior *site* que abriga projetos de software de filosofia Open Source, para que possa receber o maior número de contribuições para a evolução do projeto;
- Agente: testar a solução nas outras plataformas suportadas. Testar a solução em um ambiente real em produção;
- MIB: inclusão de novos objetos, como por exemplo o controle dos eventos monitorados (considerar somente requisições aceitas pelo servidor Web, ou qualquer tentativa);
- Padronização da MIB: criação de um documento para submeter ao IETF (Internet Engineering Task Force) como candidata a RFC (Request For Comments).

Referências

[AMO1998] AMOROSO, E.; KWAPNIEWSKI, R. **A selection criteria for Intrusion Detection Systems.** 14th Annual Computer Security Applications Conference. IEEE Computer Society Press, Los Alamitos, California, Dezembro 1998.

[APA2003] APACHE SOFTWARE FOUNDATION. **Apache HTTP Server Version 2.0 Documentation.** [online] Disponível na Internet via WWW. URL: http://httpd.apache.org/docs-2.0. Acessado em 10 de Outubro de 2003.

[CER2004] COMPUTER EMERGENCY RESPONSE TEAM / Coordination Center. **Statistics 1988-2004.** Software Engineering Institute, Carnegie Mellon University, Pittsburgh. [online] Disponível na Internet via WWW. URL: http://www.cert.org/stats. Outubro de 2004.

[CIR2004] CIRT.net.**Nikto 1.34**. [online] Disponível na Internet via WWW. URL: http://www.cirt.net/code/nikto.shtml. Outubro de 2004.

[DEN1987] DENNING, D. E. **An Intrusion Detection Model.** IEEE Trans. Software Engineering, Vol. SE-13, No. 2, Fevereiro de 1987, p. 222-232.

[FIE1999] FIELDING, R. **Hypertext Transfer Protocol – HTTP/1.1 – RFC2616.** [online] Disponível na Internet via WWW. URL: http://www.ietf.org/rfc/rfc2616.txt. Junho de 1999.

[FRA1997] FRASER, B. **Site Security Handbook - RFC2196.** Software Engineering Institute Carnegie Mellon University, Pittsburgh. [online] Disponível na Internet via WWW. URL: http://www.ietf.org/rfc/rfc2196.txt. Setembro de 1997.

[FSF1991] FREE SOFTWARE FOUNDATION. **GNU General Public License**. Free Software Foundation, Boston. [online] Disponível na Internet via WWW. URL: http://www.gnu.org/licenses/licenses.html#GPL. Junho de 1991. v. 2.

[FYO2003] FYODOR. **Top 75 Security Tools.** [online] Disponível na Internet via WWW. URL: http://www.insecure.org/tools. Maio de 2003.

[HAL1996] HALLAM-BAKER, P. M.; BEHLENDORF, B. **Extended Log File Format. W3C Working Draft WD-logfile-960323.** World Wide Web Consortium. [online] Disponível na Internet via WWW. URL: http://www.w3.org/pub/WWW/TR/WD-logfile.html. 23 de Março de 1996.

[MCH2000] McHUGH, J.; CHRISTIE, A.; ALLEN, J. **Defending Yourself: The Role of Intrusion Detection Systems.** Software Engineering Institute Carnegie Mellon University, Pittsburgh. IEEE Software, Setembro / Outubro de 2000.

[NET2003] NETCRAFT. **Web Server Survey.** [online] Disponível na Internet via WWW. URL: http://www.netcraft.com. Setembro de 2003.

[NIS2003] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, Computer Security Division. **ICAT Metabase.** [online] Disponível na Internet via WWW. URL: http://icat.nist.gov/icat.cfm.

[RIC2003] RICHARDSON, R. **2003 CSI / FBI Computer Crime and Security Survey.** Disponível em: http://www.gocsi.com. Agosto de 2003.

[SEC2003] SECURITYFOCUS. [online] Disponível na Internet via WWW. URL: http://www.securityfocus.com.

[SCA2002] SCAMBRAY, J.; SHEMA, M. **Hacking Exposed: Web Applications.** Osborne – McGraw-Hill, Junho de 2002.

[STA1999] STALLINGS, W. **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2.** 3ª ed. Upper Saddle River: Addison-Wesley, 1999.

[STE1999] STEINAUER, D.; KATZKE, S.; RADACK, S. **Basic Intrusion Protection: The First Line of Defense.** IEEE IT Pro, Janeiro / Fevereiro de 1999.

[STE2002] STEIN, L. D.; STEWART, J. N. **The World Wide Web Security FAQ, version 3.1.2.** World Wide Web Consortium. [online] Disponível na Internet via WWW. URL: http://www.w3.org/Security/Faq. 4 de Fevereiro de 2002.

[TAN1996] TANENBAUM, A. S. **Computer Networks.** 3ª ed. Upper Saddle River: Prentice-Hall, 1996.

ANEXOS

Anexo A – Padrão SNMP para gerenciamento de redes

O termo Simple Network Management Protocol (SNMP) refere-se a um conjunto de especificações para gerenciamento de redes que inclui o próprio protocolo, uma definição para estruturas de dados (SMI, que é um subconjunto do ASN.1) e conceitos associados.

O SNMP foi elaborado para atender a necessidade crescente de um padrão para gerenciamento dos dispositivos de uma rede IP.

Segundo [STA1999], no início da operação da ARPANET, virtualmente todos os hosts e subredes conectados baseavam-se em ambientes que incluíam programadores de sistemas e desenvolvedores de protocolos, que trabalhavam em algum aspecto das pesquisas para a ARPANET. Portanto, problemas de gerenciamento podiam ser deixados para especialistas em protocolos que analisariam a rede com algumas ferramentas básicas.

Quando a ARPANET se transformou na mundial Internet, o número de hosts conectados cresceu exponencialmente, acompanhado de um aumento de complexidade, tornando a solução corrente de gerenciamento de redes ineficaz.

Na RFC 1028 e 1067 houve duas tentativas de definir novas soluções, mas elas tiveram curta duração. Em maio de 1990, a RFC 1157 foi publicada, definindo a versão 1 do SNMP. Junto com um documento complementar (RFC 1155), que trata de informações de gerenciamento, o SNMP oferecia uma forma sistemática de monitoração e gerenciamento para uma rede heterogênea de computadores. Essa estrutura e o protocolo foram largamente implementados em produtos comerciais e se tornaram os padrões de facto para o gerenciamento de redes [TAN1996].

À medida que o SNMP foi sendo empregado em redes mais complexas, suas deficiências tornaram-se mais evidentes e versões aprimoradas, SNMPv2 definido nas RFCs 1441 e 1452 e SNMPv3, nas RFCs 2271-2275.

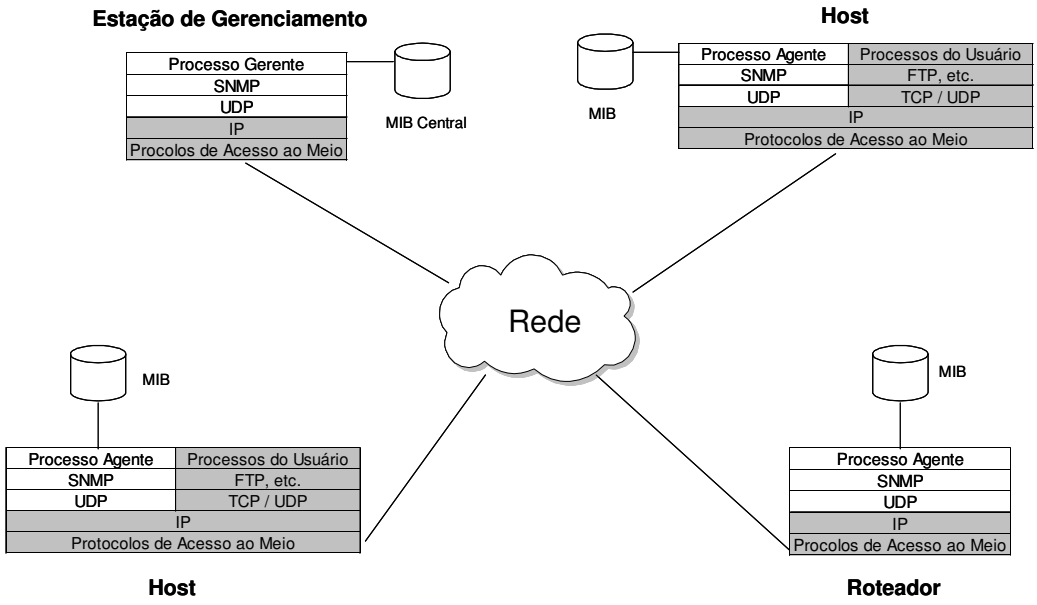
O Modelo SNMP

O modelo SNMP de gerenciamento de redes consiste em quatro componentes, a saber:

1. Nós gerenciados;
2. Estações de gerenciamento;
3. Informações de gerenciamento;

4. Protocolo de gerenciamento.

A interação entre os componentes é ilustrada na figura abaixo:



Os nós gerenciados podem ser hosts, roteadores, bridges, impressoras, ou qualquer outro dispositivo capaz de comunicar informações de estado para o mundo externo. Para ser diretamente gerenciado através do SNMP, um nó deve ser capaz de executar um processo de gerenciamento SNMP, denominado agente SNMP. Cada agente mantém um banco de dados local contendo variáveis que não só descrevem seu estado e histórico como também podem alterar a própria operação do nó. Esta possibilidade gerou uma série de discussões a respeito de segurança.

O gerenciamento de rede é feito a partir de estações de gerenciamento, que na verdade são simples estações que executam uma aplicação especial. As estações de gerenciamento contêm um ou mais processos que se comunicam com os agentes espalhados pela rede, emitindo comandos e obtendo respostas. Muitas estações de gerenciamento têm uma interface gráfica para permitir que o gerente da rede inspecione seu estado e tome as providências necessárias, se for o caso.

Para permitir que uma estação de gerenciamento se comunique com todos os componentes de uma rede heterogênea, a natureza das informações mantidas por todos os dispositivos deve ser rigidamente especificada.

Muito resumidamente, cada dispositivo mantém uma ou mais variáveis que descrevem seu estado. No modelo SNMP, esta variáveis são chamadas objetos, mas o termo é confuso, pois elas não são objetos no sentido de um sistema orientado a objetos. Estas variáveis têm apenas estados e não dispõem de métodos (além da leitura e da escrita de seus valores). O conjunto formado por todos os objetos possíveis para uma categoria de elementos gerenciáveis é fornecido em uma estrutura de dados chamada de MIB (Management Information Base).

A estação de gerenciamento interage com os agentes utilizando o protocolo SNMP. Esse protocolo permite que a estação de gerenciamento consulte o valor dos objetos locais de um agente e altere-os se possível e necessário. A maior parte do SNMP consiste nessa comunicação do tipo consulta-resposta.

No entanto, às vezes acontecem eventos que não estavam planejados. Quando um agente percebe que ocorreu um evento significativo, definido em módulo da MIB, ele imediatamente informa sua ocorrência através de um relatório a todas as estações de gerenciamento de sua lista de configuração. Esse relatório é chamado de trap do SNMP. Geralmente o relatório declara apenas que um evento ocorreu. Cabe à estação de gerenciamento executar as consultas necessárias para obter mais detalhes. Como a comunicação entre os nós gerenciados não é confiável (ou seja, não é confirmada), é uma boa idéia que a estação de gerenciamento consulte cada nó ocasionalmente, verificando se há algum evento anormal.

O modelo SNMP assume de forma geral, que cada nó gerenciado seja capaz de executar um agente SNMP internamente. Talvez dispositivos mais antigos ou que originalmente não tenham sido projetados para uso em uma rede que não disponham desse recurso. Para lidar com esses dispositivos, o SNMP define o que é chamado de agente proxy; na verdade, trata-se de um agente que controla um ou mais dispositivos não-SNMP e que se comunica com a estação de gerenciamento a fim de interceder por esses dispositivos. Possivelmente, o agente se comunica com os dispositivos através de algum tipo de protocolo não padronizado.

Segundo [TAN1997], a segurança e autenticação têm um importante papel no SNMP. Uma estação de gerenciamento tem a capacidade de obter muitas informações sobre todos os nós que estão sob seu controle e também pode desativar todos eles. Portanto, é de grande importância que todos os agentes sejam convencidos de que as consultas que alegam estar vindo da estação de gerenciamento realmente estejam vindo de lá.

Os aspectos de segurança foram ganhando maior destaque na evolução do modelo SNMP, sendo que os documentos da versão 3 definem uma estrutura para incorporar aspectos de segurança ao SNMP.

Estrutura da MIB

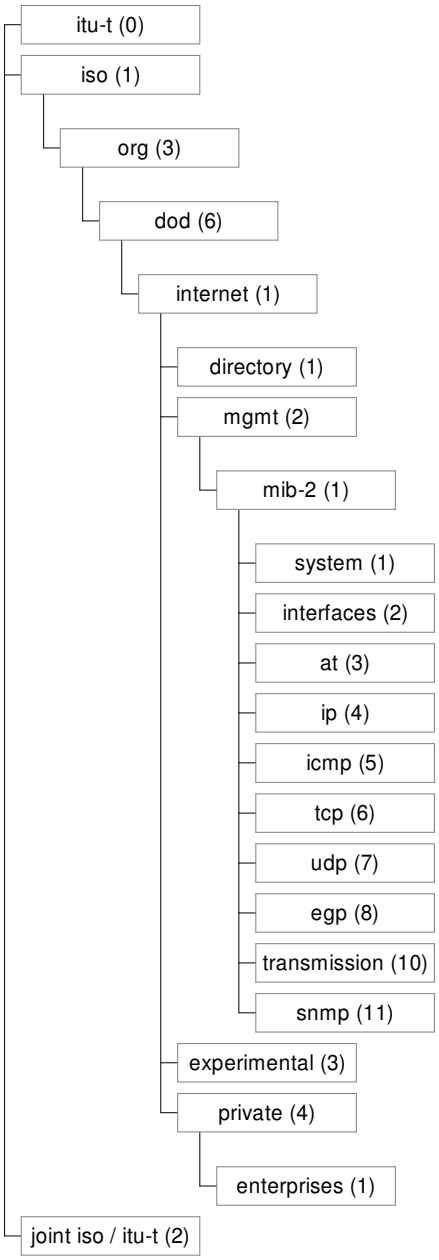
Todos os objetos gerenciados em um ambiente SNMP são organizados em uma estrutura hierárquica, ou de árvore. Os objetos-folha da árvore são os objetos realmente gerenciados, cada um deles representa determinado recurso, atividade, ou informação relacionada que deve ser gerenciado. A estrutura de árvore define um agrupamento de objetos em conjuntos logicamente correlatos.

Associado a cada tipo de objeto em uma MIB está um identificador do tipo OBJECT IDENTIFIER, em ASN.1. O identificador tem a função de nomear o objeto. Além disso, devido ao fato do valor associado ao tipo OBJECT IDENTIFIER ser hierárquico, a convenção de nomenclatura também serve para identificar a estrutura dos tipos de objetos.

O identificador de objeto é um identificador exclusivo para um objeto em particular. Seu valor consiste em uma seqüência de inteiros. O conjunto de objetos tem uma estrutura de árvore, com a raiz da árvore referindo-se ao padrão ASN.1. A partir da raiz da árvore de identificadores de objeto, cada valor (inteiro) componente do identificador identifica um nó da árvore. A partir da raiz, existem três nós no primeiro nível: iso, itu-t e joint-iso-itu-t. Abaixo do nó iso, uma ramificação é destinada ao uso de outras organizações, sendo uma delas o Departamento de Defesa Norte Americano (dod). A RFC 1155 assume que uma ramificação abaixo de dod será alocada à Internet Activities Board (IAB):

internet OBJECT IDENTIFIER ::= { iso (1) org (3) dod (6) 1 }

A figura abaixo ilustra a árvore de objetos. Desta forma, o nó internet tem o identificador de objetos de valor 1.3.6.1. Este valor serve de prefixo para os nós do nível seguinte.



O SMI define 4 ramificações abaixo de internet:

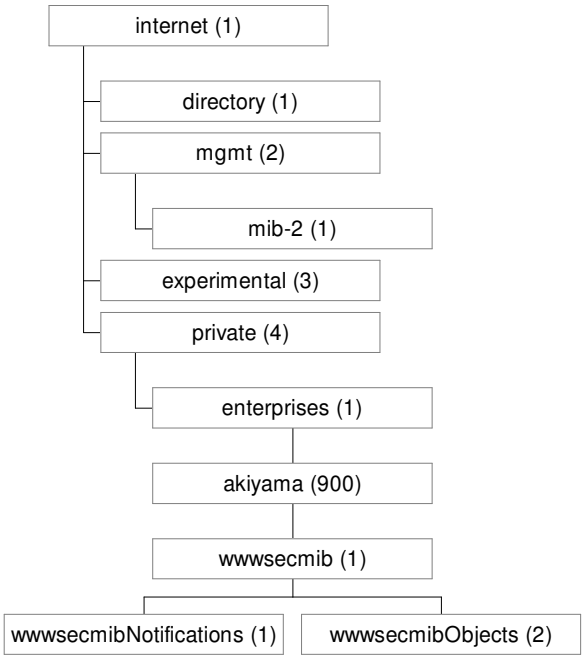
- directory: reservado para utilização futura do diretório OSI (X.500);
- mgmt: usado para objetos definidos em documentos aprovados pelo IAB;
- experimental: usado para identificar objetos usados em experiências da Internet;
- private: usado para identificar objetos definidos unilateralmente.

A árvore mgmt contém definições de MIBs aprovadas pelo IAB, atualmente mib-1 e mib-2. A mib-2 é uma extensão da primeira. Ambos têm o mesmo identificador de objetos (OID), uma vez que somente uma das MIBs pode estar presente em determinada configuração. Novos objetos podem ser definidos para uma MIB em uma das 3 formas:

1. A mib-2 pode ser ser expandida, sendo necessário definir uma nova ramificação;
2. Uma MIB experimental pode ser construída para uma aplicação em particular. Estes objetos podem ser deslocados para a árvore mgmt;
3. Extensões particulares podem ser adicionadas na ramificação private [STA1999].

A ramificação private tem atualmente somente um nó definido, o enterprises. Este segmento da ramificação é utilizado para permitir que fabricantes tanto de hardware como de software aumentem o grau de gerenciamento de seus produtos e para compartilhar estas informações com outros usuários, ou fabricantes que necessitem integrar com seus sistemas. Um segmento na ramificação enterprise é alocado para cada fabricante que obtiver um registro e, consequentemente um OID.

A IANA gerencia todas as atribuições de números da ramificação enterprises, sendo que não estão restritos a empresas. Qualquer instituição pode solicitar seu registro e criar MIBs públicas. No entanto, foi desenvolvida neste trabalho uma ramificação enterprise, denominada “akiyama”, com o número fictício inicialmente. O diagrama abaixo apresenta a localização da ramificação deste trabalho.



A definição da MIB, codificada em SMIv2 é apresentada no Apêndice B.

Anexo B – Código da MIB

Definição da MIB, codificada em SMIv2:

```
WWWSEC-MIB DEFINITIONS ::= BEGIN
    IMPORTS
    IpAddress, Integer32, Opaque, OBJECT-TYPE,
    NOTIFICATION-TYPE, MODULE-IDENTITY, enterprises
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, DisplayString
        FROM SNMPv2-TC;

    wwwsecmib MODULE-IDENTITY
        LAST-UPDATED      "200408172349Z"
        ORGANIZATION      "akiyama"
        CONTACT-INFO        "Vinicius Akiyama
                             IBM Brasil
                             Sao Paulo - SP
                             CEP: 04007-900 Brasil
                             Tel: +55 11 2132 5373
                             E-mail: akiyamav@br.ibm.com"
        DESCRIPTION         "The MIB module for managing security of web servers."

        REVISION          "200408172349Z"
        DESCRIPTION         "initial version of this module"
        ::= { akiyama 1 }

    WwwServerList ::= TEXTUAL-CONVENTION
        STATUS            current
        DESCRIPTION         "Identifica o servidor a ser monitorado.
                             1 - Apache,
                             2 - Microsoft Internet Information Server
                             3 - Zeus
                             4 - Sun SunONE"
        SYNTAX            Integer32 (1..4)

    WwwLogList ::= TEXTUAL-CONVENTION
        STATUS            current
        DESCRIPTION         "Especifica o tipo de log a ser tratado.
                             1 - W3C Extended Log Format,
                             2 - NCSA Log Format / Common Log Format
        SYNTAX            Integer32 (1..2)



    akiyama OBJECT IDENTIFIER ::= { enterprises 900 }

    wwwsecmibNotifications OBJECT IDENTIFIER ::= { wwwsecmib 1 }

    attackEvent NOTIFICATION-TYPE
        OBJECTS       {
                      attackSignEntryCode,
                      attackSignEntryURI,
                      attackSignEntryIP
              }
        STATUS            current
        DESCRIPTION         "Informa a ocorrencia de um ataque."
        ::= { wwwsecmibNotifications 1 }

    wwwsecmibObjects OBJECT IDENTIFIER ::= { wwwsecmib 2 }

    wwwsecServerType OBJECT-TYPE
        SYNTAX              WwwServerList
        MAX-ACCESS          read-write
        STATUS              current
        DESCRIPTION         "Especifica o tipo de servidor gerenciado."
        DEFVAL              { 1 }
        ::= { wwwsecmibObjects 1 }
```

```
    wwwsecLogFormat OBJECT-TYPE
        SYNTAX              WwwLogList
        MAX-ACCESS          read-write
        STATUS              current
        DESCRIPTION         "Especifica o tipo de log a ser tratado."
        DEFVAL              { 2 }
        ::= { wwwsecmibObjects 2 }

    wwwsecLogDir OBJECT-TYPE
        SYNTAX              DisplayString
        MAX-ACCESS          read-write
        STATUS              current
        DESCRIPTION         "Informa a diretorio onde se localiza o arquivo
                            de log do Web server."
        ::= { wwwsecmibObjects 3 }

    wwwsecLogFile OBJECT-TYPE
        SYNTAX              DisplayString
        MAX-ACCESS          read-write
        STATUS              current
        DESCRIPTION         "Informa o nome do arquivo de log do servidor Web."
        ::= { wwwsecmibObjects 4 }

    wwwsecMonState OBJECT-TYPE
        SYNTAX              Integer32 (0..1)
        MAX-ACCESS          read-write
        STATUS              current
        DESCRIPTION         "Descreve o estado do monitor (1) ativo, (0) inativo."
        DEFVAL              { 0 }
        ::= { wwwsecmibObjects 5 }

    wwwsecError OBJECT-TYPE
        SYNTAX              Integer32 (0..15)
        MAX-ACCESS          read-write
        STATUS              current
        DESCRIPTION             "Contem o codigo de erro de 2 bits em decimal. 0
Representa estado OK.
                             Funcao de cada bit:
                             0 - Localizacao do diretorio
                             1 - Localizacao do arquivo
        DEFVAL              { 0 }
        ::= { wwwsecmibObjects 6 }

    wwwsecAttackSignTable OBJECT-TYPE
        SYNTAX              SEQUENCE OF WwwsecAttackSignEntry
        MAX-ACCESS          not-accessible
        STATUS              current
        DESCRIPTION      "Tabela de ataques relacionados no arquivo de assinaturas
                            relacionado no arquivo de configuracao."
        ::= { wwwsecmibObjects 7 }

    wwwsecAttackSignEntry OBJECT-TYPE
        SYNTAX              WwwsecAttackSignEntry
        MAX-ACCESS          not-accessible
        STATUS              current
        DESCRIPTION         "Please replace this text with your description."
        INDEX                               {  attackSignEntryCode,  attackSignEntryIP,
attackSignEntryURI }
        ::= { wwwsecAttackSignTable 1 }

    WwwsecAttackSignEntry ::= SEQUENCE {
        attackSignEntryCode
            Integer32,
        attackSignEntryIP
            IpAddress,
        attackSignEntryURI
            DisplayString,
        attackSignEntryCnt
            Opaque
    }

    attackSignEntryCode OBJECT-TYPE
        SYNTAX              Integer32
```

```
        MAX-ACCESS          read-only
        STATUS              current
        DESCRIPTION         "Codigo do ataque, obtido no arquivo de assinaturas."
        ::= { wwwsecAttackSignEntry 1 }

    attackSignEntryIP OBJECT-TYPE
        SYNTAX              IpAddress
        MAX-ACCESS          read-only
        STATUS              current
        DESCRIPTION         "Endereco IP do atacante."
        ::= { wwwsecAttackSignEntry 2 }

    attackSignEntryURI OBJECT-TYPE
        SYNTAX              DisplayString
        MAX-ACCESS          read-only
        STATUS              current
        DESCRIPTION         "Informa o recurso solicitado que ocasionou o erro."
        ::= { wwwsecAttackSignEntry 3 }

    attackSignEntryCnt OBJECT-TYPE
        SYNTAX              Opaque
        MAX-ACCESS          read-only
        STATUS              current
        DESCRIPTION         "Contador dos erros."
        ::= { wwwsecAttackSignEntry 4 }
END
```

Anexo C – Código do Web IDS

O detector de intrusão é composto por três módulos, com as seguintes funções:

- Extensão do agente Net-SNMP;
- Principal, ou módulo de inteligência;
- Atualização automática da base de assinaturas;
- MIB.

Módulo de extensão do agente Net-SNMP

Arquivos: wwwsecmib.h (arquivo de cabeçalho)
wwwsecmib.c (código)

O módulo ainda possui algumas limitações, que o impedem de armazenar variáveis em formato string e gerar alarmes. Para validação da solução foi utilizado um arquivo texto que registra as informações de cada ataque detectado em substituição aos alarmes e inclusas as informações das variáveis string diretamente no código do agente.

O módulo ainda possui algumas limitações, que o impedem de armazenar variáveis em formato string. Para validação da solução foram atribuídos os valores das variáveis em formato string diretamente no código do agente.

wwwsecmib.h

```
/*
 * Note: this file originally auto-generated by mib2c using
 *        : mib2c.scalar.conf,v 1.7 2003/04/08 14:57:04 dts12 Exp $
 */
#ifndef WWWSECMIB_H
#define WWWSECMIB_H

/*
 * function declarations
 */
void            init_wwwsecmib(void);
Netsnmp_Node_Handler handle_wwwsecLogDir;
Netsnmp_Node_Handler handle_wwwsecMonState;
Netsnmp_Node_Handler handle_wwwsecServerType;
Netsnmp_Node_Handler handle_wwwsecLogFile;
Netsnmp_Node_Handler handle_wwwsecLogFormat;
Netsnmp_Node_Handler handle_wwwsecError;

#endif                          /* WWWSECMIB_H */
```

wwwsecmib.c

```
/*
 * Note: this file originally auto-generated by mib2c using
 *        : mib2c.scalar.conf,v 1.7 2003/04/08 14:57:04 dts12 Exp $
 */

#include <net-snmp/net-snmp-config.h>
#include <net-snmp/net-snmp-includes.h>
#include <net-snmp/agent/net-snmp-agent-includes.h>
#include "wwwsecmib.h"

#include <stdlib.h>

static int mstate  = 0;
static int lformat = 3;
static int stype   = 1;
static int secerr  = 0;

/** Initializes the wwwsecmib module */
void
init_wwwsecmib(void)
{
    static oid      wwwsecMonState_oid[] =
        { 1, 3, 6, 1, 4, 1, 900, 1, 2, 5, 0 };
    static oid      wwwsecServerType_oid[] =
        { 1, 3, 6, 1, 4, 1, 900, 1, 2, 1, 0 };
    static oid      wwwsecLogFormat_oid[] =
        { 1, 3, 6, 1, 4, 1, 900, 1, 2, 2, 0 };
    static oid      wwwsecError_oid[] =
        { 1, 3, 6, 1, 4, 1, 900, 1, 2, 6, 0 };

    DEBUGMSGTL(("wwwsecmib", "Initializing\n"));

    netsnmp_register_int_instance ("wwwsecMonState",
                                   wwwsecMonState_oid,
                                   OID_LENGTH(wwwsecMonState_oid),
                                   &mstate, NULL);
    netsnmp_register_int_instance ("wwwsecServerType",
                                   wwwsecServerType_oid,
                                   OID_LENGTH(wwwsecServerType_oid),
                                   &stype, NULL);
    netsnmp_register_int_instance ("wwwsecLogFormat",
                                   wwwsecLogFormat_oid,
                                   OID_LENGTH(wwwsecLogFormat_oid),
                                   &lformat, NULL);
    netsnmp_register_int_instance ("wwwsecError",
                                   wwwsecError_oid,
                                   OID_LENGTH(wwwsecError_oid),
                                   &secerr, NULL);
}
```

Módulo principal

Nome do arquivo: wids.pl

```
#!/usr/bin/perl
#
# Modulo externo do agente SNMP para
# gerenciamento de seguranca de servidores web
#
# wids.pl
# Autor: Vinicius Akiyama
# em 10/2004
```

```
# Mestrado Profissional do IPT
#

use Net::SNMP;
use IO::Handle;
use Time::localtime;

$mstate = 0;
$naptime = 1;

sub lemib {
   ($session,$error) = Net::SNMP->session(Hostname => $ARGV[0],
                                          Community => $ARGV[1]);

   die "session error: $error" unless ($session);
   $result = $session->get_request("1.3.6.1.4.1.900.1.2.5.0");
   $mstate = $result->{"1.3.6.1.4.1.900.1.2.5.0"};
   die "request error: ".$session->error unless (defined $result);

   $session->close;

   print "Monitor state: ",$mstate,"\n";
}

# Funcao para verificacao do estado do monitor
sub monitor_state {
   while ($mstate == 0) {
      # requires a hostname and a community string as its arguments
      ($session,$error) = Net::SNMP->session(Hostname => $ARGV[0],
                                          Community => $ARGV[1]);

      die "session error: $error" unless ($session);

      # Verifica o estado do monitor
      $result = $session->get_request("1.3.6.1.4.1.900.1.2.5.0");
      $mstate = $result->{"1.3.6.1.4.1.900.1.2.5.0"};

      die "request error: ".$session->error unless (defined $result);

      $session->close;

      print "Monitor state: ",$mstate,"\n";
      sleep 5;
   }
}

sub map_log_fields {
   # Verifica o formato do log
   ($session,$error) = Net::SNMP->session(Hostname => $ARGV[0],
                                     Community => $ARGV[1]);
   die "session error: $error" unless ($session);
   $result = $session->get_request("1.3.6.1.4.1.900.1.2.2.0");
   $lformat = $result->{"1.3.6.1.4.1.900.1.2.2.0"};
   die "request error: ".$session->error unless (defined $result);
   $session->close;
   #

   # Mapeia os campos de acordo com o formato do log
   SWITCH: {
      if ($lformat == 1) {
         # Formato W3C
         ($datacc, $timeacc, $ipaddr, $ident, $serverip, $accmeth, $uri, $uriq,
$scode) = split(" ", $logent);
         last SWITCH;
      }
      # Default = NCSA
      ($ipaddr, $ident, $usrid, $datacc, $tzone, $accmeth, $uri, $prot, $scode) =
split(" ", $logent);
   }
}

sub verif_acesso {
   # Verifica o tipo do servidor
   ($session,$error) = Net::SNMP->session(Hostname => $ARGV[0],
```

```
                                    Community => $ARGV[1]);
   die "session error: $error" unless ($session);
   $result = $session->get_request("1.3.6.1.4.1.900.1.2.1.0");
   $stype = $result->{"1.3.6.1.4.1.900.1.2.1.0"};
   die "request error: ".$session->error unless (defined $result);
   $session->close;
   #

   SWITCH: {
      if ($stype == 2) {
         $attfile = "db/web_ids_iis.db";
         last SWITCH;
      }
      if ($stype == 3) {
         $attfile = "db/web_ids_zeus.db";
         last SWITCH;
      }
      if ($stype == 4) {
         $attfile = "db/web_ids_sun.db";
         last SWITCH;
      }
      $attfile = "db/web_ids_apache.db";
    }

### Somente para debug
    open(CONTAATT,">> contataque.log") or die "Can't open file: $!\n";
    print "Verificando contra o arquivo de assinaturas $attfile \n";
    $attflag = 0;
    open(SIGFILE,$attfile) or die "Can't open file: $!\n";
    while ($line = <SIGFILE>) {
        ($index, $sigpat, $sigcom) = split(",", $line);
        if ($uri =~ $sigpat) {
           # Verifica se o servidor aceitou a requisicao
           # $scode < 400
           if ($scode < 400) {
              # Envia trap
              system "snmptrap -v 1 -c $ARGV[1] $ARGV[0] .1.3.6.1.4.1.900.1.1
$ARGV[0] 6 1 '' .1.3.6.1.4.1.900.1.2.7.1.1.0 i $index .1.3.6.1.4.1.900.1.2.7.1.2.0 a
$ipaddr .1.3.6.1.4.1.900.1.2.7.1.3.0 s $uri";
              print "Matched! $index, $sigcom\n";
              print CONTAATT "$uri, $scode, $index, $sigcom\n";
              $attflag = 1;
           }
        }
    }
    close SIGFILE;
    close CONTAATT;
    if ($attflag == 0) { print "Acesso normal! $ipaddr, $uri"; }
}

sub tail_log {
   # Abre arquivo de acordo com os valores dos objetos
   # wwwsecLogDir e wwwsecLogFile
   open (LOGFILE, "/usr/local/apache2/logs/access_log") or die "can't open logfile:
$!";
   seek (LOGFILE,0,2); # Posiciona o handle no final do arquivo
   while ($mstate == 1) {
      # Tratamento de novos acessos logados
      while ($logent = <LOGFILE>) {
         # Mapeia os campos
         map_log_fields();
         # Verifica se e ataque
         verif_acesso();
         print $mstate,"\n";
      }
      sleep $naptime;
      print 'Aguardando nova entrada ';
      lemib();
      LOGFILE->clearerr();
   }
}

# Rotina principal em loop infinito
for (;;) {
```

```
   # Verifica o estado do monitor
   monitor_state();
   # Vai para a rotina de tratamento da log
   # somente se o monitor for ativo (objeto wwwsecMonState)
   tail_log();
   # Sai do tratamento da log se o monitor for desativado
   # e reinicia o loop
}
```

Módulo de atualização automática da base de assinaturas

Nome do arquivo: autoupdate.pl

```
#!/usr/bin/perl
# autoupdate.pl - 10/10/2004
# versao 0.9
# release 4
#
# Detector de Intrusoes em Servidores Web
# Modulo de atualizacao da base de assinaturas
#

use File::Compare;
use IO::Handle;
use HTTP::Lite;

sub savetofile {
   my ($self,$dataref,$cbargs) = @_;
      print $cbargs $$dataref;
      return undef;
}

sub separa_campos {
   open(SIGSOURCE,"db/nikto.db") or die "Can't open file: $!\n";
   open SRCAPACHE, ">db/web_ids_apache.src";
   open SRCIIS, ">db/web_ids_iis.src";
   open SRCZEUS,">db/web_ids_zeus.src";
   open SRCSUN,">db/web_ids_sun.src";
   $cnt_apache = 1;
   $cnt_iis = 1;
   $cnt_zeus = 1;
   $cnt_sun = 1;
   $inicio = 0;
   while ($line = <SIGSOURCE>) {
      if ($inicio == 0) {
         if ($line =~ "These are normal tests") {
            $inicio = 1;
         }
      } else {
         ($stype, $atpatt, $var1, $meth, $attdesc) = split(",", $line);
         if ($stype =~ "apache") {
            print SRCAPACHE "$cnt_apache, $atpatt, $attdesc";
            ++$cnt_apache;
         } elsif ($stype =~ "iis") {
              print SRCIIS "$cnt_iis,$atpatt,$attdesc";
              ++$cnt_iis;
         } elsif ($stype =~ "iplanet") {
              print SRCSUN "$cnt_sun,$atpatt,$attdesc";
              ++$cnt_sun;
         } elsif ($stype =~ "zeus") {
              print SRCZEUS "$cnt_zeus,$atpatt,$attdesc";
              ++$cnt_zeus;
         }
      }
   }
}
```

```
####################
# Estrutura principal
#
#rename ("nikto.db","nikto.db.old");
#$url = "http://www.cirt.net/nikto/UPDATES/1.34/scan_database.db";
#$http = new HTTP::Lite;
#open OUT, ">nikto.db";
#$res = $http->request($url, \&savetofile, OUT);
#close OUT;

if (compare("db/nikto.db","db/nikto.db.old") == 0) {
    print "They're equal\n";
} else {
      print "They're not equal\n";
      separa_campos();
}
                    #
####################
```

Anexo D – Lista de assinaturas de ataques

A estrutura dos registros de ataques está detalhada no capítulo 5. O conteúdo dos arquivos de assinaturas inicialmente integrantes do detector de intrusão desenvolvido neste trabalho está relacionado a seguir.

```
Tipo de servidor:   Apache
Nome do arquivo:    db/web_ids_apache.db

1,/.DS_Store,Apache on Mac OSX will serve the .DS_Store file which contains sensitive
information. Configure Apache to ignore this file or upgrade to a newer version.
2,/.FBCIndex,This file son OSX contains the source of the files in the directory.
http://www.securiteam.com/securitynews/5LP0O005FS.html
3,//,Apache on Red Hat Linux release 9 reveals the root directory listing by default
if there is no index page.
4,/666%0a%0a<script>alert('Vulnerable');</script>666.jsp,Apache Tomcat 4.1 / Linux is
vulnerable to Cross Site Scripting (XSS). CA-2000-02.
5,/?D=A,Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.
6,/?M=A,Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.
7,/?N=D,Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.
8,/?S=A,Apache allows directory listings by requesting. Upgrade Apache or disable
directory indexing.
9,/admin.cgi,InterScan VirusWall administration is accessible without authentication.
10,/blah-whatever.jsp,The Apache Tomcat 3.1 server reveals the web root path when
requesting a non-existent JSP file. CAN-2000-0759.
11,/cgi-bin/main_menu.pl,The NetDetector allows unauthenticated users to perform
database queries.
12,/cgi-bin/printenv,Apache 2.0 default script is executable and gives server
environment variables. All default scripts should be removed. It may also allow XSS
types of attacks. BID-4431.
13,/cgi-bin/printenv,Apache 2.0 printenv default script does not have execute
permissions but leaks file system paths. It may also allow XSS types of attacks. BID-
4431.
14,/cgi-bin/search,Apache Stronghold 3.0 may reveal the web root in the source of
this CGI ('sourcedir' value).
15,/cgi-bin/test-cgi,Apache 2.0 default script is executable and reveals system
information. All default scripts should be removed.
16,/cgi-bin/test-cgi,Apache 2.0 printenv default script does not have execute
permissions but leaks file system paths.
17,/content/base/build/explorer/none.php?..:..:..:..:..:..:..:etc:passwd:,SunPS
iRunbook Version 2.5.2 allows files to be read remotely.
18,/content/base/build/explorer/none.php?/etc/passwd,SunPS iRunbook Version 2.5.2
allows files to be read remotely.
19,/doc/rt/overview-summary.html,Oracle Business Components for Java 3.1 docs is
running.
20,/doc/webmin.config.notes,Webmin config file found may contain Webmin ID/Password.
Typically runs on port 10000.
21,/docs/,May give list of installed software
22,/docs/sdb/en/html/index.html,This may be a default SuSe Apache install. This is
the support page.
23,/error/%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5cwindows%5cwin.ini,Apache allows
files to be retrieved outside of the web root. Apache should be upgraded to 2.0.40 or
above. CAN-2002-0661.
24,/error/%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5cwinnt%5cwin.ini,Apache allows files
to be retrieved outside of the web root. Apache should be upgraded to 2.0.40 or
above. CAN-2002-0661.
25,/error/HTTP_NOT_FOUND.html.var,Apache reveals file system paths when invalid error
documents are requested.
26,/examples/,Directory indexing enabled also default JSP examples.
27,/examples/jsp/index.html,Apache Tomcat default JSP pages present.
28,/examples/jsp/snp/snoop.jsp,Displays information about page retrievals including
other users.
```

29,/examples/jsp/source.jsp??,Tomcat 3.23/3.24 allows directory listings by performing a malformed request to a default jsp. Default pages should be removed.
30,/examples/servlet/AUX,Apache Tomcat versions below 4.1 may be vulnerable to DoS by repeatedly requesting this file.
31,/examples/servlet/TroubleShooter,Tomcat default jsp page reveals system information and may be vulnerable to XSS.
32,/examples/servlets/index.html,Apache Tomcat default JSP pages present.
33,/icons/,Directory indexing is enabled it should only be enabled for specific directories (if required). If indexing is not used all the /icons directory should be removed.
34,/index.html.ca,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
35,/index.html.cz.iso8859-2,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
36,/index.html.de,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
37,/index.html.dk,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
38,/index.html.ee,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
39,/index.html.el,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
40,/index.html.en,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
41,/index.html.es,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
42,/index.html.et,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
43,/index.html.fr,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
44,/index.html.he.iso8859-8,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
45,/index.html.hr.iso8859-2,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
46,/index.html.it,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
47,/index.html.ja.iso2022-jp,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
48,/index.html.kr.iso2022-kr,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
49,/index.html.ltz.utf8,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
50,/index.html.lu.utf8,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
51,/index.html.nl,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
52,/index.html.nn,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
53,/index.html.no,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
54,/index.html.po.iso8859-2,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.

55,/index.html.pt,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
56,/index.html.pt-br,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
57,/index.html.ru.cp-1251,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
58,/index.html.ru.cp866,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
59,/index.html.ru.iso-ru,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
60,/index.html.ru.koi8-r,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
61,/index.html.ru.utf8,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
62,/index.html.se,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
63,/index.html.tw,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
64,/index.html.tw.Big5,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
65,/index.html.var,Apache default foreign language file found. All default files should be removed from the web server as they may give an attacker additional system information.
66,/interscan/,InterScan VirusWall administration is accessible without authentication.
67,/jservdocs/,Default Apache JServ docs should be removed.
68,/lpt9,Apache Tomcat 4.0.3 reveals the web root when requesting a non-existent DOS device. Upgrade to version 4.1.3beta or higher.
69,/main_page.php,Mazu Networks Profiler or Sensor is running.
70,/manual/images/,Apache 2.0 directory indexing is enabled it should only be enabled for specific directories (if required). Apache's manual should be removed and directory indexing disabled.
71,/NetDetector/middle_help_intro.htm,The system appears to be a Niksun NetDetector (network monitoring). ÊThe help files should be available at /NetDetector/quick_help_index.html
72,/oem_webstage/cgi-bin/oemapp_cgi,Oracle reveals the CGI source by prepending /oem_webstage to CGI urls.
73,/oem_webstage/oem.conf,Oracle reveals a portion of the Apache httpd.conf file.
74,/php/php.exe?c:boot.ini,The Apache config allows php.exe to be called directly.
75,/pls/admin,Oracle Apache+WebDB gives a lot of system information via the pls/admin script
76,/server-info,This gives a lot of Apache information. Comment out appropriate line in httpd.conf or restrict access to allowed hosts.
77,/server-status,This gives a lot of Apache information. Comment out appropriate line in httpd.conf or restrict access to allowed hosts.
78,/servlet/MsgPage?action=test&msg=<script>alert('Vulnerable')</script>,NetDetector 3.0 and below are vulnerable to Cross Site Scripting (XSS). CA-2000-02.
79,/servlet/org.apache.catalina.ContainerServlet/<script>alert('Vulnerable')</script>,Apache-Tomcat is vulnerable to Cross Site Scripting (XSS) by invoking java classes. CA-2000-02.
80,/servlet/org.apache.catalina.Context/<script>alert('Vulnerable')</script>,Apache-Tomcat is vulnerable to Cross Site Scripting (XSS) by invoking java classes. CA-2000-02.
81,/servlet/org.apache.catalina.Globals/<script>alert('Vulnerable')</script>,Apache-Tomcat is vulnerable to Cross Site Scripting (XSS) by invoking java classes. CA-2000-02.
82,/servlet/org.apache.catalina.servlets.WebdavStatus/<script>alert('Vulnerable')</script>,Apache-Tomcat is vulnerable to Cross Site Scripting (XSS) by invoking java classes. CA-2000-02.
83,/servlets/MsgPage?action=badlogin&msg=<script>alert('Vulnerable')</script>,The NetDetector install is vulnerable to Cross Site Scripting (XSS) in it's invalid login message. CA-2000-02.
84,/site/eg/source.asp,This asp (installed with Apache::ASP) allows attackers to upload files to the server. Upgrade to 1.95 or higher. CAN-2000-0628.

85,/soap/servlet/soaprouter,Oracle 9iAS SOAP components allow anonymous users to deploy applications by default.
86,/soapConfig.xml,Oracle 9iAS configuration file found - see bugrtraq #4290.
87,/stronghold-info,Redhat Stronghold from versions 2.3 up to 3.0 disclose sensitive information. This gives information on configuration. CAN-2001-0868.
88,/stronghold-status,Redhat Stronghold from versions 2.3 up to 3.0 disclose sensitive information. CAN-2001-0868.
89,/test,Apache Tomcat default file found. All default files should be removed.
90,/test/jsp/buffer1.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
91,/test/jsp/buffer2.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
92,/test/jsp/buffer3.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
93,/test/jsp/buffer4.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
94,/test/jsp/declaration/IntegerOverflow.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
95,/test/jsp/extends1.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
96,/test/jsp/extends2.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
97,/test/jsp/Language.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
98,/test/jsp/pageAutoFlush.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
99,/test/jsp/pageDouble.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
100,/test/jsp/pageExtends.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
101,/test/jsp/pageImport2.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
102,/test/jsp/pageInfo.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
103,/test/jsp/pageInvalid.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
104,/test/jsp/pageIsErrorPage.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
105,/test/jsp/pageIsThreadSafe.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
106,/test/jsp/pageSession.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
107,/test/realPath.jsp,Apache Tomcat default file found which reveals the web root. The /test directory should be removed.
108,/tomcat-docs/index.html,Default Apache Tomcat documentation found.
109,/XSQLConfig.xml,Oracle 9iAS configuration file found - see bugrtraq #4290.
110,/~nobody/etc/passwd,Apache is misconfigured to view files by accessing ~nobody/filename. Change UserDir from './' to something else in httpd.conf.
111,@CGIDIRS.htaccess,Contains authorization information
112,@CGIDIRStest-cgi.bat,This is an Apache for Win default. If Apache is lower than 1.3.23 this can be exploited as in test-cgi.bat?|dir+c:+>..htdocslisting.txt but may not allow data sent back to the browser.

Tipo de servidor: Microsoft Internet Information Server
Nome do arquivo: db/web_ids_iis.db

1,/%NETHOOD%/,The machine may be infected with the Bugbear.B virus. http://www.f-secure.com/v-descs/bugbear_b.shtml
2,/.....config.sys,PWS allows files to be read by prepending multiple '.' characters. At worst IIS not PWS should be used.
3,/../../../../../../../../../boot.ini,The remote server allows any system file to be retrieved remotely.
4,/../../../../winnt/repair/sam._,Sam backup successfully retrieved.
5,/..\..\..\..\..\..\..\..\boot.ini,It is possible to read files on the server by adding /../ in front of file name.
6,/<script>alert('Vulnerable')</script>.shtm,Server is vulnerable to Cross Site Scripting (XSS). CA-2000-02.
7,/<script>alert('Vulnerable')</script>.stm,Server is vulnerable to Cross Site Scripting (XSS). CA-2000-02.
8,/?><script>alert('Vulnerable');</script>,IIS is vulnerable to Cross Site Scripting (XSS). See MS02-018 CVE-2002-0075 SNS-49 CA-2002-09
9,/admcgi/contents.htm,Default FrontPage CGI found.

10,/admcgi/scripts/Fpadmcgi.exe,Default FrontPage CGI found.

11,/admentor/adminadmin.asp,Version 2.11 of AdMentor is vulnerable to SQL injection during login in the style of: ’ or =

12,/admin/adminproc.asp,Xpede administration page may be available. The /admin directory should be protected.

13,/admin/datasource.asp,Xpede administration page may be available (this page reveals SQL account name). The /admin directory should be protected.

14,/admin/sh_taskframes.asp?Title=Configuraci%C3%B3n%20de%20registro%20Web&URL=Master
Settings/Web_LogSettings.asp?tab1=TabsWebServer%26tab2=TabsWebLogSettings%26__SAPageK
ey=5742D5874845934A134CD05F39C63240&ReturnURL=><script>alert(document.cookie)</script
>?tab1=TabsWebServer%26__SAPageKey=5742D5874845934A134CD05F39C63
240%26R=0.6756681557204625&R=0.9895845379540951&__SAPageKey=5742D5874845934A134CD05F3
9C63240,IIS 6 on Windows 2003 is vulnerable to Cross Site Scripting (XSS) in certain error messages. CA-2000-02.

15,/admisapi/fpadmin.htm,Default FrontPage file found.

16,/asp/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.

17,/asp/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.

18,/bin/admin.pl,Default FrontPage CGI found.

19,/bin/cfgwiz.exe,Default FrontPage CGI found.

20,/bin/CGImail.exe,Default FrontPage CGI found.

21,/bin/contents.htm,Default FrontPage CGI found.

22,/bin/fpadmin.htm,Default FrontPage CGI found.

23,/bin/fpremadm.exe,Default FrontPage CGI found.

24,/bin/fpsrvadm.exe,Default FrontPage CGI found.

25,/blahb.ida,Reveals physical path. To fix: Preferences -> Home directory -> Application & check ’Check if file exists’ for the ISAPI mappings. MS01-033.

26,/blahb.idq,Reveals physical path. To fix: Preferences -> Home directory -> Application & check ’Check if file exists’ for the ISAPI mappings. MS01-033.

27,/c/winnt/system32/cmd.exe?/c+dir+/OG,This machine is infected with Code Red or has Code Red leftovers.

28,/certsrv/..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.

29,/certsrv/..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884

30,/cgi-bin/..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.

31,/cgi-bin/..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884

32,/cgi-bin/admin.pl,Default FrontPage CGI found.

33,/cgi-bin/cfgwiz.exe,Default FrontPage CGI found.

34,/cgi-bin/CGImail.exe,Default FrontPage CGI found.

35,/cgi-bin/contents.htm,Default FrontPage CGI found.

36,/cgi-bin/fpadmin.htm,Default FrontPage CGI found.

37,/cgi-bin/fpremadm.exe,Default FrontPage CGI found.

38,/cgi-bin/fpsrvadm.exe,Default FrontPage CGI found.

39,/cgi-bin/snorkerz.bat,Arguments passed to DOS CGI without checking

40,/cgi-bin/snorkerz.cmd,Arguments passed to DOS CGI without checking

41,/cgi-dos/args.bat,Default FrontPage CGI found.

42,/cgi-win/uploader.exe,This CGI allows attackers to upload files to the server and then execute them.

43,/contents/extensions/asp/1,The IIS system may be vulnerable to a DOS see MS02-018 for details.

44,/database/db2000.mdb,Max Web Portal database is available remotely. It should be moved from the default location to a directory outside the web root.

45,/default.htm%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20.htw?CiWebHitsFile=../../../../..
/../../../../win.ini&CiRestriction=none&CiHiliteType=Full,Server may be vulnerable to a Webhits.dll arbitrary file retrieval. Ensure Q252463i Q252463a or Q251170 is installed. MS00-006.

46,/default.htm%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2

0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%
20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%2
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20.htw?CiWebHitsFile=../../../../..
/../../../../win.ini&CiRestriction=none&CiHiliteType=Full,Server may be vulnerable to a Webhits.dll arbitrary file retrieval. Ensure Q252463i Q252463a or Q251170 is installed. MS00-006.
47,/exchange/,This may be interesting (Outlook exchange OWA server?)...
48,/exchange/lib/AMPROPS.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
49,/exchange/lib/ATTACH.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
50,/exchange/lib/DELETE.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
51,/exchange/lib/GETREND.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
52,/exchange/lib/GETWHEN.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
53,/exchange/lib/JSATTACH.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
54,/exchange/lib/JSROOT.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
55,/exchange/lib/JSUTIL.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
56,/exchange/lib/LANG.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
57,/exchange/lib/logon.inc,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
58,/exchange/lib/PAGEUTIL.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
59,/exchange/lib/PUBFLD.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
60,/exchange/lib/RENDER.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
61,/exchange/lib/SESSION.INC,Outlook Web Access server allows source code to be viewed by requesting the file directly from /exchange/lib/
62,/exchange/root.asp?acs=anon,This allows anonymous access to portions of the OWA server.
http://support.microsoft.com/support/exchange/content/whitepapers/owaguide.doc
63,/global.asa,The global.asa file was retrieved which may contain sensitive information. Map the .asa extension to the proper dll.
64,/iisadmin/,Access to /iisadmin should be restricted to localhost or allowed hosts only.
65,/iisadmpwd/..%255c..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.
66,/iisadmpwd/..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
67,/iisadmpwd/aexp2.htr,Gives domain and system name may allow an attacker to brute force for access. Also will allow an NT4 user to change his password regardless of the 'user cannot change password' security policy. CAN-2002-0421. BID-4236. BID-2110.
68,/iisadmpwd/aexp2b.htr,Gives domain and system name may allow an attacker to brute force for access. Also will allow an NT4 user to change his password regardless of the 'user cannot change password' security policy. CAN-2002-0421. BID-4236. BID-2110.
69,/iisadmpwd/aexp3.htr,Gives domain and system name may allow an attacker to brute force for access. Also will allow an NT4 user to change his password regardless of the 'user cannot change password' security policy. CAN-2002-0421. BID-4236. BID-2110.
70,/iisadmpwd/aexp4.htr,Gives domain and system name may allow an attacker to brute force for access. Also will allow an NT4 user to change his password regardless of the 'user cannot change password' security policy. CAN-2002-0421. BID-4236. BID-2110.
71,/iisadmpwd/aexp4b.htr,Gives domain and system name may allow an attacker to brute force for access. Also will allow an NT4 user to change his password regardless of the 'user cannot change password' security policy. CAN-2002-0421. BID-4236. BID-2110.
72,/iissamples/exair/howitworks/Codebrw1.asp,This is a default IIS script/file which should be removed it may allow a DoS against the server. CAN-1999-0738. MS99-013. CVE-1999-0449. BID-193.
73,/iissamples/exair/howitworks/codebrws.asp,This is a default IIS script/file which should be removed it may allow a DoS against the server or a DoS. CAN-1999-0738. MS99-013. CVE-1999-0449. BID-193.
74,/iissamples/exair/howitworks/Winmsdp.exe,This is a default IIS script/file which should be removed it may allow a DoS against the server. CAN-1999-0738. MS99-013. CVE-1999-0449. BID-193.

75,/iissamples/exair/search/advsearch.asp,Scripts within the Exair package on IIS 4 can be used for a DoS against the server. CVE-1999-0449. BID-193.
76,/iissamples/exair/search/query.asp,Scripts within the Exair package on IIS 4 can be used for a DoS against the server. CVE-1999-0449. BID-193.
77,/iissamples/exair/search/query.idq?CiTemplate=../../../../../../../../../../winnt/win.ini,This allows arbitrary files to be retrieved from the server it may allow a DoS against the server. CVE-1999-0449. BID-193. MS01-033.
78,/iissamples/exair/search/search.asp,Scripts within the Exair package on IIS 4 can be used for a DoS against the server. CVE-1999-0449. BID-193.
79,/iissamples/exair/search/search.idq?CiTemplate=../../../../../../../../../../winnt/win.ini,This allows arbitrary files to be retrieved from the server it may allow a DoS against the server. CVE-1999-0449. BID-193. MS01-033.
80,/iissamples/issamples/codebrws.asp,This is a default IIS script/file which should be removed. CAN-1999-0738. MS99-013.
81,/iissamples/issamples/fastq.idq?CiTemplate=../../../../../../../../../../winnt/win.ini,This allows arbitrary files to be retrieved from the server. MS01-033.
82,/iissamples/issamples/ixqlang.htm,IIS default file found. All default files should be removed.
83,/iissamples/issamples/oop/qfullhit.htw?CiWebHitsFile=/iissamples/issamples/oop/qfullhit.htw&CiRestriction=none&CiHiliteType=Full,It is possible to retrieve the source of .asp files. Install Webhits patch at http://www.microsoft.com/technet/security/bulletin/ms00-006.asp. MS00-006 CVE-2000-0097.
84,/iissamples/issamples/oop/qsumrhit.htw?CiWebHitsFile=/iissamples/issamples/oop/qsumrhit.htw&CiRestriction=none&CiHiliteType=Full,It is possible to retrieve the source of .asp files. Install Webhits patch at http://www.microsoft.com/technet/security/bulletin/ms00-006.asp. MS00-006 CVE-2000-0097.
85,/iissamples/issamples/query.idq?CiTemplate=../../../../../../../../../../winnt/win.ini,This allows arbitrary files to be retrieved from the server. MS01-033.
86,/iissamples/issamples/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.
87,/iissamples/issamples/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.
88,/iissamples/issamples/Winmsdp.exe,This is a default IIS script/file which should be removed. CAN-1999-0738. MS99-013.
89,/iissamples/sdk/asp/docs/codebrw2.asp,This is a default IIS script/file which should be removed. CAN-1999-0738. MS99-013.
90,/iissamples/sdk/asp/docs/codebrws.asp,This is a default IIS script/file which should be removed. CAN-1999-0738. MS99-013.
91,/iissamples/sdk/asp/docs/codebrws.asp,IIS 5 comes with an ASP that allows remote code to viewed. All default files in /IISSamples should be removed. CAN-1999-0738. MS99-013.
92,/iissamples/sdk/asp/docs/CodeBrws.asp?Source=/IISSAMPLES/%c0%ae%c0%ae/%c0%ae%c0%ae/bogus_directory/nonexistent.asp,CodeBrws.asp can be used to determine if a file system path exists or not. CAN-1999-0738. MS99-013.
93,/iissamples/sdk/asp/docs/CodeBrws.asp?Source=/IISSAMPLES/%c0%ae%c0%ae/default.asp, IIS may be vulnerable to source code viewing via the example CodeBrws.asp file. Remove all default files from the web root. CAN-1999-0738. MS99-013.
94,/iissamples/sdk/asp/docs/Winmsdp.exe,This is a default IIS script/file which should be removed. CAN-1999-0738. MS99-013.
95,/iissamples/sdk/asp/docs/Winmsdp.exe,IIS 5 comes with an ASP that allows remote code to viewed. All default files in /IISSamples should be removed. CAN-1999-0738. MS99-013.
96,/iissamples/sdk/asp/docs/Winmsdp.exe?Source=/IISSAMPLES/%c0%ae%c0%ae/%c0%ae%c0%ae/bogus_directory/nonexistent.asp,Winmsdp.exe can be used to determine if a file system path exists or not. CAN-1999-0738. MS99-013.
97,/iissamples/sdk/asp/docs/Winmsdp.exe?Source=/IISSAMPLES/%c0%ae%c0%ae/default.asp,IIS may be vulnerable to source code viewing via the example Winmsdp.exe file. Remove all default files from the web root. CAN-1999-0738. MS99-013.
98,/isapi/tstisapi.dll,The test tstisapi.dll is available and can allow attackers to execute commands remotely.
99,/ISSamples/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.
100,/ISSamples/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.
101,/JUNK(10),Server appears to be running eEye's SecureIIS application http://www.eeye.com/.
102,/JUNK(10)abcd.html,The IIS server may be vulnerable to Cross Site Scripting (XSS) in error messages ensure Q319733 is installed see MS02-018 CVE-2002-0075
103,/JUNK(5).htw,Server may be vulnerable to a Webhits.dll arbitrary file retrieval. Ensure Q252463i Q252463a or Q251170 is installed. MS00-006.
104,/junk.aspx,ASP.net reveals its version in invalid .aspx error messages. http://www.tconsult.com/aspnet/exceptions/globalexception.aspx

105,/junk.aspx,ASP.net reveals file system paths in invalid .aspx requests. http://www.tconsult.com/aspnet/exceptions/globalexception.aspx
106,/msadc/..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.
107,/msadc/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir+c:%5c,May be able to issue arbitrary commands to host.
108,/msadc/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir+c:%5c,May be able to issue arbitrary commands to host.
109,/msadc/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir+c:%5c,Can issue arbitrary commands to host.
110,/msadc/..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
111,/msadc/msadcs.dll,See RDS advisory RFP9902 CVE-1999-1011 MS98-004 MS99-025 RFP-9902 BID-29 (http://www.wiretrip.net/rfp/p/doc.asp/i2/d1.htm)
112,/msadc/samples/adctest.asp,The IIS sample application adctest.asp may be used to remotely execute commands on the server. RFP9901 (http://www.wiretrip.net/rfp/p/doc.asp/i2/d3.htm)
113,/msadc/samples/adctest.asp,This may allow remote code execution on the server.
114,/msadc/Samples/selector/showcode.asp?source=/msadc/Samples/../../../../../../../../../winnt/win.ini,This allows attackers to read arbitrary files on the host. CAN-1999-0738. MS99-013.
115,/nikto.ida,The IIS server is running UrlScan
116,/nul..cfm,ColdFusion 5.0 and below 4.0-5.0 reveal file system paths of .cfm or .dbm files when the request contains invalid DOS devices. Macromedia MPSB02-01. CVE-2002-0576. KPMG-2002013. BID-4542. http://www.macromedia.com/v1/handlers/index.cfm?ID=22906
117,/nul..dbm,ColdFusion 5.0 and below 4.0-5.0 reveal file system paths of .cfm or .dbm files when the request contains invalid DOS devices. Macromedia MPSB02-01. CVE-2002-0576. KPMG-2002013. BID-4542. http://www.macromedia.com/v1/handlers/index.cfm?ID=22906
118,/nul.cfm,ColdFusion 5.0 and below 4.0-5.0 reveal file system paths of .cfm or .dbm files when the request contains invalid DOS devices. CVE-2002-0576. KPMG-2002013. BID-4542. http://www.macromedia.com/v1/handlers/index.cfm?ID=22906
119,/nul.dbm,ColdFusion 5.0 and below 4.0-5.0 reveal file system paths of .cfm or .dbm files when the request contains invalid DOS devices. CVE-2002-0576. KPMG-2002013. BID-4542. http://www.macromedia.com/v1/handlers/index.cfm?ID=22906
120,/null.htw?CiWebHitsFile=/default.asp%20&CiRestriction=none&CiHiliteType=Full,It is possible to retrieve the source of .asp files or view any file on the system. Install Webhits patch at http://www.microsoft.com/technet/security/bulletin/ms00-006.asp. MS00-006 CVE-2000-0097.
121,/NULL.printer,Internet Printing (IPP) is enabled. Some versions have a buffer overflow/DoS in Windows 2000 which allows remote attackers to gain admin privileges via a long print request that is passed to the extension through IIS 5.0. Disabling the .printer mapping is recommended. EEYE-AD20010501 CVE-2001-0241 MS01-023 CA-2001-10
122,/oc/Search/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.
123,/oc/Search/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.
124,/officescan/cgi/cgiChkMasterPwd.exe,Trend Officescan allows you to skip the login page and access soem CGI programs directly.
125,/OpenFile.aspx?file=../../../../../../../../../../../boot.ini,HTTP Commander 4.0 allows directory traversal and reading of arbitrary files.
126,/pbserver/..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.
127,/pbserver/..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
128,/pbserver/pbserver.dll,This may contain a buffer overflow. http://www.microsoft.com/technet/security/bulletin/ms00-094.asp
129,/prd.i/pgen/,has MS Merchant Server 1.0
130,/prxdocs/misc/prxrch.idq?CiTemplate=../../../../../../../../../../../winnt/win.ini,This allows arbitrary files to be retrieved from the server. MS01-033.
131,/query.idq?CiTemplate=../../../../../../../../../../../winnt/win.ini,This allows arbitrary files to be retrieved from the server. MS01-033.
132,/readme.eml,Remote server may be infected with the Nimda virus.
133,/rpc/..%255c..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.

134,/rpc/..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
135,/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.
136,/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+ver,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.
137,/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
138,/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
139,/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
140,/scripts/admin.pl,Default FrontPage CGI found.
141,/scripts/Carello/Carello.dll,Carello 1.3 may allow commands to be executed on the server by replacing hidden form elements. This could not be tested by Nikto.
142,/scripts/cfgwiz.exe,Default FrontPage CGI found.
143,/scripts/CGImail.exe,Default FrontPage CGI found.
144,/scripts/contents.htm,Default FrontPage CGI found.
145,/scripts/cpshost.dll,posting acceptor...possibly allows you to upload files
146,/scripts/fpadmin.htm,Default FrontPage CGI found.
147,/scripts/fpcount.exe,Default FrontPage CGI found.
148,/scripts/fpremadm.exe,Default FrontPage CGI found.
149,/scripts/fpsrvadm.exe,Default FrontPage CGI found.
150,/scripts/httpodbc.dll,Possible IIS backdoor found.
151,/scripts/iisadmin/bdir.htr,This default script shows host info may allow file browsing and buffer a overrun in the Chunked Encoding data transfer mechanism request /scripts/iisadmin/bdir.htr??c:<dirs> . MS02-028. CA-2002-09.
152,/scripts/iisadmin/ism.dll,allows you to mount a brute force attack on passwords
153,/scripts/no-such-file.pl,Using perl.exe allows attacker to view host info. Use perlis.dll instead.
154,/scripts/proxy/w3proxy.dll,MSProxy v1.0 installed
155,/scripts/repost.asp,This allows uploads to /users. Create /users and give web user read only access.
156,/scripts/root.exe?/c+dir+c:+/OG,This machine is infected with Code Red or has Code Red leftovers.
157,/scripts/samples/details.idc,See RFP 9901; www.wiretrip.net
158,/scripts/samples/search/author.idq,This is a default IIS script/file which should be removed. MS01-033.
159,/scripts/samples/search/filesize.idq,This is a default IIS script/file which should be removed. MS01-033.
160,/scripts/samples/search/filetime.idq,This is a default IIS script/file which should be removed. MS01-033.
161,/scripts/samples/search/qfullhit.htw,Server may be vulnerable to a Webhits.dll arbitrary file retrieval. MS00-006.
162,/scripts/samples/search/qsumrhit.htw,Server may be vulnerable to a Webhits.dll arbitrary file retrieval. MS00-006.
163,/scripts/samples/search/queryhit.idq,This is a default IIS script/file which should be removed. MS01-033.
164,/scripts/samples/search/simple.idq,This is a default IIS script/file which should be removed. MS01-033.
165,/scripts/tools/ctss.idc,This CGI allows remote users to view and modify SQL DB contents server paths docroot and more.
166,/scripts/tools/dsnform,An oldie but goodie... allows creation of ODBC Data Source
167,/scripts/tools/dsnform.exe,An oldie but goodie... allows creation of ODBC Data Source
168,/scripts/tools/getdrvrs.exe,MS Jet database engine can be used to make DSNs useful with an ODBC exploit and the RDS exploit (with msadcs.dll) which mail allow command execution. RFP9901 (http://www.wiretrip.net/rfp/p/doc.asp/i2/d3.htm).
169,/scripts/tools/newdsn.exe,This can be used to make DSNs useful in use with an ODBC exploit and the RDS exploit (with msadcs.dll). Also may allow files to be created on the server. BID-1818. CVE-1999-0191. RFP9901 (http://www.wiretrip.net/rfp/p/doc.asp/i2/d3.htm)
170,/search.asp?Search=><script>alert()</script>,Max Web Portal is vulnerable to Cross Site Scripting (XSS). CA-2000-02.
171,/search/htx/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.
172,/search/htx/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.

173,/search/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.
174,/search/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.
175,/servlet/com.newatlanta.servletexec.JSP10Servlet/,ServletExec 4.1 ISAPI Java Servlet/JSP Engine for IIS discloses the web root. The server may also be vulnerable to a DoS attack by requesting a long file name ending in .jsp
176,/servlet/com.newatlanta.servletexec.JSP10Servlet/..%5c..%5cglobal.asa,ServletExec 4.1 ISAPI Java Servlet/JSP Engine for IIS can reveal source code. The server may also be vulnerable to a DoS attack by requesting a long file name ending in .jsp
177,/Sites/Knowledge/Membership/Inspired/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
178,/Sites/Knowledge/Membership/Inspiredtutorial/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
179,/Sites/Samples/Knowledge/Membership/Inspired/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
180,/Sites/Samples/Knowledge/Membership/Inspiredtutorial/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
181,/Sites/Samples/Knowledge/Push/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
182,/Sites/Samples/Knowledge/Search/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
183,/siteseed/,Siteseed pre 1.4.2 has 'major' security problems.
184,/SiteServer/admin/,SiteServer components admin. Default account may be 'LDAP_Anonymous' pass is 'LdapPassword_1'. see http://www.wiretrip.net/rfp/p/doc.asp/i1/d69.htm
185,/SiteServer/Admin/commerce/foundation/domain.asp,Displays known domains of which that server is involved.
186,/SiteServer/Admin/commerce/foundation/driver.asp,Displays a list of installed ODBC drivers.
187,/SiteServer/Admin/commerce/foundation/DSN.asp,Displays all DSNs configured for selected ODBC drivers.
188,/SiteServer/admin/findvserver.asp,Gives a list of installed Site Server components.
189,/SiteServer/Admin/knowledge/dsmgr/default.asp,USed to view current search catalog configurations
190,/SiteServer/Admin/knowledge/dsmgr/users/GroupManager.asp,Used to create modify and potentially delete LDAP users and groups.
191,/SiteServer/Admin/knowledge/dsmgr/users/UserManager.asp,Used to create modify and potentially delete LDAP users and groups.
192,/SiteServer/Admin/knowledge/persmbr/vs.asp,Expose various LDAP service and backend configuration parameters
193,/SiteServer/Admin/knowledge/persmbr/VsLsLpRd.asp,Expose various LDAP service and backend configuration parameters
194,/SiteServer/Admin/knowledge/persmbr/VsPrAuoEd.asp,Expose various LDAP service and backend configuration parameters
195,/SiteServer/Admin/knowledge/persmbr/VsTmPr.asp,Expose various LDAP service and backend configuration parameters
196,/SiteServer/Knowledge/Default.asp?ctr=><script>alert('Vulnerable')</script>,Site Server is vulnerable to Cross Site Scripting
197,/SiteServer/Publishing/ViewCode.asp,The default ViewCode.asp can allow an attacker to read any file on the machine. CAN-1999-0738. MS99-013.
198,/siteserver/publishing/viewcode.asp?source=/default.asp,May be able to view source code using Site Server vulnerability. CAN-1999-0738. MS99-013.
199,/smg_Smxcfg30.exe?vcc=3560121183d3,This may be a Trend Officesan 'backdoor'.
200,/SQLQHit.asp,This sample ASP allows anyone to retrieve directory listings.
201,/sqlqhit.asp,This sample ASP allows anyone to retrieve directory listings.
202,/ssi/envout.bat?|dir%20..\..\..\..\..\..\..\,This CGI allows attackers to read files from the server.
203,/trace.axd,The .NET IIS server has application tracing enabled. This could allow an attacker to view the last 50 web requests.
204,/tvcs/getservers.exe?action=selects1,Following steps 2-4 of this page may reveal a zip file which contains passwords and system details.
205,/upload.asp,An ASPpage that allows attackers to upload files to server
206,/uploadn.asp,An ASPpage that allows attackers to upload files to server
207,/uploadx.asp,An ASPpage that allows attackers to upload files to server
208,/wa.exe,An ASPpage that allows attackers to upload files to server
209,/WebAdmin.dll?View=Logon,Some versions of WebAdmin are vulnerable to a remote DoS (not tested). See http://www.ngssoftware.com.
210,/whatever.htr,Reveals physical path. htr files may also be vulnerable to an off-by-one overflow that allows remote command execution (see MS02-018)
211,/whatever.htr,Reveals physical path. htr files may also be vulnerable to an off-by-one overflow that allows remote command execution (see MS02-018)

212,/WS_FTP.LOG,WS_FTP.LOG file was found. It may contain sensitive information.
213,/_cti_pvt/,FrontPage directory found.
214,/_mem_bin/auoconfig.asp,Displays the default AUO (LDAP) schema including host and port.
215,/_mem_bin/auoconfig.asp,LDAP information revealed via asp. See http://www.wiretrip.net/rfp/p/doc.asp/i1/d69.htm
216,/_mem_bin/formslogin.asp?><script>alert('Vulnerable')</script>,Site Server is vulnerable to Cross Site Scripting
217,/_mem_bin/remind.asp,Page will give the password reminder for any user requested (username must be known).
218,/_private/,FrontPage directory found.
219,/_private/form_results.html,This file may contain information submitted by other web users via forms. CAN-1999-1052.
220,/_private/form_results.txt,This file may contain information submitted by other web users via forms. CAN-1999-1052.
221,/_private/orders.htm,Default FrontPage file found.
222,/_private/orders.txt,Default FrontPage file found.
223,/_private/register.htm,Default FrontPage file found.
224,/_private/register.txt,Default FrontPage file found.
225,/_private/registrations.htm,Default FrontPage file found.
226,/_private/registrations.txt,Default FrontPage file found.
227,/_private/_vti_cnf/,FrontPage directory found.
228,/_vti_bin/,FrontPage directory found.
229,/_vti_bin/,shtml.exe is available remotely. Some versions of the Front Page ISAPI filter are vulnerable to a DOS (not attempted) see MS-018 for details. CVE-2000-0709. BID-1608
230,/_vti_bin/..%255c..%255c..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir ,IIS is vulnerable to a double-decode bug which allows commands to be executed on the system. CAN-2001-0333. BID-2708.
231,/_vti_bin/..%c0%af../..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir,IIS Unicode command exec problem see http://www.wiretrip.net/rfp/p/doc.asp?id=57&face=2 and http://www.securitybugware.org/NT/1422.html. CVE-2000-0884
232,/_vti_bin/admin.pl,Default FrontPage CGI found.
233,/_vti_bin/cfgwiz.exe,Default FrontPage CGI found.
234,/_vti_bin/CGImail.exe,Default FrontPage CGI found.
235,/_vti_bin/contents.htm,Default FrontPage CGI found.
236,/_vti_bin/fpadmin.htm,Default FrontPage CGI found.
237,/_vti_bin/fpcount.exe,Frontpage counter CGI has been found. FP Server version 97 allows remote users to execute arbitrary system commands though a vulnerability in this version could not be confirmed. CAN-1999-1376. BID-2252.
238,/_vti_bin/fpremadm.exe,Default FrontPage CGI found.
239,/_vti_bin/fpsrvadm.exe,Default FrontPage CGI found.
240,/_vti_bin/shtml.dll/_vti_rpc?method=server+version%3a4%2e0%2e2%2e2611,Gives info about server settings. CAN-2000-0413 CAN-2000-0709 CAN-2000-0710 BID-1608
241,/_vti_bin/shtml.exe,Attackers may be able to crash FrontPage by requesting a DOS device like shtml.exe/aux.htm -- a DoS was not attempted. CAN-2000-0413 CAN-2000-0709 CAN-2000-0710
242,/_vti_bin/shtml.exe/junk_nonexistant.exe,This exe shows the full web path when a non-existent file is requested. CAN-2000-0413 CAN-2000-0709 CAN-2000-0710 BID-1608
243,/_vti_bin/shtml.exe/_vti_rpc,FrontPage may be installed.
244,/_vti_bin/shtml.exe/_vti_rpc?method=server+version%3a4%2e0%2e2%2e2611,Gives info about server settings. CAN-2000-0413 CAN-2000-0709 CAN-2000-0710 BID-1608
245,/_vti_bin/_vti_aut/author.dll?method=list+documents%3a3%2e0%2e2%2e1706&service%5fname=&listHiddenDocs=true&listExplorerDocs=true&listRecurse=false&listFiles=true&listFolders=true&listLinkInfo=true&listIncludeParent=true&listDerivedT=false&listBorders=false,We seem to have authoring access to the FrontPage web.
246,/_vti_bin/_vti_aut/author.exe?method=list+documents%3a3%2e0%2e2%2e1706&service%5fname=&listHiddenDocs=true&listExplorerDocs=true&listRecurse=false&listFiles=true&listFolders=true&listLinkInfo=true&listIncludeParent=true&listDerivedT=false&listBorders=false,We seem to have authoring access to the FrontPage web.
247,/_vti_bin/_vti_aut/dvwssr.dll,This dll allows anyone with authoring privs to change other users file and may contain a buffer overflow for unauthenticated users. See also : http://www.wiretrip.net/rfp/p/doc.asp?id=45&iface=1. MS00-025.
248,/_vti_bin/_vti_aut/fp30reg.dll,Some versions of the FrontPage fp30reg.dll are vulnerable to a buffer overflow. See http://www.microsoft.com/technet/security/bulletin/ms03-051.asp for details.
249,/_vti_bin/_vti_aut/fp30reg.dll?xxx,This dll is vulnerable to a remote overflow which can allow attackers to execute remote commands.
250,/_vti_bin/_vti_cnf/,FrontPage directory found.
251,/_vti_cnf/_vti_cnf/,FrontPage directory found.
252,/_vti_inf.html,FrontPage may be installed.

253,/_vti_log/_vti_cnf/,FrontPage directory found.
254,/_vti_pvt/access.cnf,Contains HTTP server-specific access control information remove or ACL if FrontPage is not being used.
255,/_vti_pvt/administrators.pwd,Default FrontPage file found may be a password file.
256,/_vti_pvt/authors.pwd,Default FrontPage file found may be a password file.
257,/_vti_pvt/linkinfo.cnf,IIS file shows http links on and off site. Might show host trust relationships and other machines on network.
258,/_vti_pvt/service.cnf,Contains meta-information about the web server remove or ACL if FrontPage is not being used.
259,/_vti_pvt/service.pwd,Default FrontPage file found may be a password file.
260,/_vti_pvt/services.cnf,Contains the list of subwebs remove or ACL if FrontPage is not being used. May reveal server version if Admin has changed it.
261,/_vti_pvt/svacl.cnf,File used to store whether subwebs have unique permissions settings and any IP address restrictions. Can be used to discover information about subwebs remove or ACL if FrontPage is not being used.
262,/_vti_pvt/users.pwd,Default FrontPage file found may be a password file.
263,/_vti_pvt/writeto.cnf,Contains information about form handler result files remove or ACL if FrontPage is not being used.
264,/_vti_txt/,FrontPage directory found.
265,/_vti_txt/_vti_cnf/,FrontPage directory found.
266,@CGIDIRScmd.exe?/c+dir,cmd.exe can execute arbitrary commands
267,@CGIDIRScmd1.exe?/c+dir,cmd1.exe can execute arbitrary commands
268,@CGIDIRSfpsrvadm.exe,Potentially vulnerable CGI program.
269,@CGIDIRShello.bat?&dir+c:,This batch file may allow attackers to execute remote commands.
270,@CGIDIRSinput.bat?|dir%20..\..\..\..\..\..\..\..\..\,This CGI allows attackers to read files from the server.
271,@CGIDIRSinput2.bat?|dir%20..\..\..\..\..\..\..\..\..\,This CGI allows attackers to read files from the server.
272,@CGIDIRSMsmMask.exe,MondoSearch 4.4 may allow source code viewing by requesting MsmMask.exe?mask=/filename.asp where 'filename.asp' is a real asp file.
273,@CGIDIRSpost32.exe|dir%20c:,post32 can execute arbitrary commands
274,@CGIDIRSsensepost.exe?/c+dir,The presence of sensepost.exe indicates the system is/was vulnerable to a Unicode flaw and was compromised with a test script from SensePost. The sensepost.exe allows command execution (it is a copy of cmd.exe) as did the original unicode exploit (see http://www.securitybugware.org/NT/1422.html). CVE-2000-0884.
275,@CGIDIRSshtml.dll,This may allow attackers to retrieve document source.
276,@CGIDIRSSQLServ/sqlbrowse.asp?filepath=c:&Opt=3,Hosting Controller versions 1.4.1 and lower can allow arbitrary files/directories to be read. Upgrade.
277,@CGIDIRSstats/statsbrowse.asp?filepath=c:&Opt=3,Hosting Controller versions 1.4.1 and lower can allow arbitrary files/directories to be read. Upgrade.
278,@CGIDIRStest.bat?|dir%20..\..\..\..\..\..\..\..\..\,This CGI allows attackers to read files from the server.
279,@CGIDIRStst.bat|dir%20..\..\..\..\..\..\..\..\,This CGI allows attackers to execute arbitrary commands on the server.

Tipo de servidor: SunONE
Nome do arquivo: db/web_ids_iplanet.db

1,/admin-serv/tasks/configuration/ViewLog?file=passwd&num=5000&str=&directories=admin-serv%2Flogs%2f..%2f..%2f..%2f..%2f..%2fetc&id=admin-serv,iPlanet Administration Server 5.1 allows remote users to download any file from the server. Upgrade to SunOne DS5.2 and in iDS5.1 SP2 Hotfix 2.
2,/manual/servlets/scripts/servlet1/servform.htm,iPlanet default servlet found. All default code should be removed.
3,/manual/servlets/scripts/shoes/shoeform.htm,iPlanet default servlet found. All default code should be removed.

Tipo de servidor: Zeus
Nome do arquivo: db/web_ids_zeus.db

1,/apps/web/vs_diag.cgi?server=<script>alert('Vulnerable')</script>,Zeus 4.2r2 (webadmin-4.2r2) is vulnerable to Cross Site Scripting (XSS). CA-2000-02.
2,/apps/web/index.fcgi?servers=&section=<script>alert(document.cookie)</script>,Zeus Admin server 4.1r2 is vulnerable to Cross Site Scripting (XSS). CA-2000-02.