

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

**IMPLANTAÇÃO DE SISTEMA DE DETECÇÃO DE INTRUSÃO - IDS EM UM
AMBIENTE CORPORATIVO**

FABIO ANDRÉ BARCELOS

**NATAL/RN
2010**

FABIO ANDRÉ BARCELOS

**IMPLANTAÇÃO DE SISTEMA DE DETECÇÃO DE INTRUSÃO - IDS EM UM
AMBIENTE CORPORATIVO**

Monografia apresentada ao curso de Especialização em Redes de Computadores da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Redes de Computadores.

Orientador:

Aluizio Ferreira da Rocha Neto, M.Sc.

**NATAL/RN
2010**

LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

Fábio André Barcelos

IMPLANTAÇÃO DE SISTEMA DE DETECÇÃO DE INTRUSÃO - IDS EM UM
AMBIENTE CORPORATIVO

Monografia apresentada e aprovada em ____ de _____ de ____, pela banca examinadora
composta pelos seguintes membros

Aluizio Ferreira da Rocha Neto, M.Sc.
Orientador

Membro examinador

Membro examinador

Natal/RN, ____ de _____ de ____.

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Barcelos, Fabio André

Implantação de sistema de detecção de intrusão: IDS em um ambiente corporativo / Fabio André Barcelos. – Natal: FARN, 2010.
36 f.

Orientador: Aluizio Ferreira da Rocha Neto (Msc.)

Monografia (Especialização em Redes de Computadores) –
Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Redes de Computadores – Monografia. 2. Segurança de Redes –
Monografia. 3. Segurança da Informação – Monografia. I. Título.

RN/FARN/BC

CDU 004.72

AGRADECIMENTOS

A Deus, pela sua presença constante em minha vida.

“TUDO POSSO NAQUELE QUE ME FORTALECE.”

À minha esposa, pois sempre me incentivou e apoiou nos estudos.

Aos professores do curso Pós-Graduação em Redes de Computadores – FARN, por compartilharem seus conhecimentos. E a todos aqueles que, de algum modo, me ajudaram.

RESUMO

A importância da segurança da informação progrediu à medida que o número contínuo de ataques e intrusões a sistemas e a redes de computadores aumentaram. Juntamente com as suas numerosas vantagens, a internet também criou inúmeras maneiras de comprometer a segurança e a estabilidade dos sistemas conectados a ela. Operações de gestão de segurança protegem as redes de computadores contra ataques de negação de serviço, contra divulgação não autorizada de informações, e, também, contra a alteração ou destruição de dados. Sistemas de detecção de intrusão ou IDS tornaram-se um importante componente na caixa de ferramentas do Especialista de Segurança, porque diariamente empresas são atacadas por invasores em busca de informações sigilosas.

Palavras-chaves: Segurança; Ataques; Intrusão.

ABSTRACT

The importance of information security has progressed as the continuous number of attacks and intrusion systems and computer networks has increased. Together with its numerous advantages, the Internet has also created numerous ways to compromise security and stability of systems connected to it. Management operations security protect computer networks against denial of service, unauthorized disclosure of information, and also against alteration or destruction of data. Intrusion Detection Systems or IDS became an important component in the toolbox of Specialist Security, because businesses are attacked daily by invaders in search of sensitive information.

Keywords: Security, Attack, Intrusion.

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS.....	10
1.1.1	<i>Objetivos Específicos</i>	<i>10</i>
1.2	MOTIVAÇÃO	10
2	REFERENCIAL TEÓRICO	12
2.1	TROJAN HORSES	12
2.2	BUFFER OVERFLOW	13
2.3	EXPLOITS.....	13
2.4	BACKDOORS	14
2.5	DOS (DENIAL OF SERVICE)	15
2.5.1	<i>Ataque Distribuído</i>	<i>15</i>
2.6	SISTEMA DE DETECÇÃO DE INTRUSÃO	15
2.7	TIPOS DE DETECÇÃO.....	16
2.7.1	<i>Baseados na Estação.....</i>	<i>16</i>
2.7.2	<i>Baseados em Rede.....</i>	<i>18</i>
2.7.3	<i>Baseados em Pilhas.....</i>	<i>18</i>
2.8	MÉTODOS DE DETECÇÃO	19
2.8.1	<i>Detecção de Anomalias.....</i>	<i>19</i>
2.8.2	<i>Detecção de Assinaturas</i>	<i>19</i>
3	ESTUDO DE CASO	21
3.1	A REDE.....	21
3.2	A FERRAMENTA SNORT	21
3.3	OUTROS SOFTWARES REQUERIDOS	22
3.3.1	<i>Apache Web Server</i>	<i>23</i>
3.3.2	<i>MySQL Server</i>	<i>23</i>
3.3.3	<i>Analysis Console for Intrusion Databases (ACID)</i>	<i>23</i>
3.4	INSTALAÇÃO DOS SOFTWARES REQUERIDOS	23
3.5	INSTALAÇÃO DO SNORT	25
3.6	CONFIGURAÇÃO DO SNORT.....	29
3.6.1	<i>Configurações das variáveis de rede</i>	<i>29</i>
3.6.2	<i>Configuração dos pré-processadores</i>	<i>31</i>
3.6.3	<i>Configuração dos plugins de saída.....</i>	<i>31</i>
3.7	CUSTOMIZAÇÃO DAS REGRAS	32
3.8	INSTALAÇÃO E CONFIGURAÇÃO DO ACID	33
4	CONCLUSÃO.....	35
	REFERÊNCIAS BIBLIOGRÁFICAS	36

LISTA DE FIGURAS

Figura 1: Localização do IDS na rede do estudo de caso	21
Figura 2: Instalação do Apache 2	24
Figura 3: Instalação do MySQL server.....	24
Figura 4: Instalação do PHP 5	24
Figura 5: Instalação do suporte a MySQL no PHP	24
Figura 6: Configurar o iptables para liberação do trafego IP	24
Figura 7: Pequeno código PHP para teste do servidor Apache com PHP	25
Figura 8: Instalação do PCRE	25
Figura 9: Instalação do Snort.....	25
Figura 10: Configurações dos parâmetros do Snort	26
Figura 11: Configuração da base de dados MySQL.....	26
Figura 12: Criando as tabelas na base de dados do snort	27
Figura 13: Observando o banco de dados do Snort criado no MySQL.....	27
Figura 14: Observando as tabelas da base de dados do Snort	28
Figura 15: Instalação da biblioteca libpcap	28
Figura 16: Instalação do Snort.....	29
Figura 17: Variáveis de configuração do Snort.....	29
Figura 18: Lista de pré-processadores na configuração do Snort.....	31
Figura 19: Comando para iniciar o Snort	32
Figura 20: Interface principal do ACID	33
Figura 21: Interface do ACID com alertas	34

1 INTRODUÇÃO

Ao mesmo tempo em que ocorreu um enorme avanço tecnológico nas comunicações em rede, possibilitando o compartilhamento de informações a nível mundial, a segurança dessas redes e informações tornou-se crítica. Sendo o meio físico compartilhado, a probabilidade de que pessoas não autorizadas consigam acesso a informações sigilosas através de métodos ilícitos é grande. Desse modo, o uso de especialistas em segurança, aliados a poderosas ferramentas que o auxiliem, tornou-se indispensável na gestão de sistemas computacionais modernos.

Uma dessas ferramentas é o sistema de detecção de intrusão, que, analogamente a um sistema do mesmo nome empregado na segurança domiciliar, é composto por sensores capazes de disparar um alarme caso algum evento determinado ou não esperado venha a acontecer.

1.1 OBJETIVOS

O principal objetivo deste trabalho é demonstrar uma de várias formas que existem de detectar intrusos e ataques em uma rede de computadores. Para isso será estudado suas características, seu funcionamento e os tipos de IDs.

1.1.1 Objetivos Específicos

Este trabalho tem como finalidade a instalação de um Sistema de Detecção de Intrusão – SNORT e os softwares requeridos para o seu funcionamento tais como: Debian lenny, MySQL, ACID, PHP, ADODB e o navegador Mozilla. O SNORT demonstrará as anomalias na rede de dados conforme as regras que forem configuradas.

1.2 MOTIVAÇÃO

Os crimes digitais de informática estão se tornando uma constante em nossa sociedade. Seja através de engenharia social, de invasões de sistemas, Trojans (cavalos de tróia), de vírus, através da clonagem de sites, envio de e-mails com código malicioso, ou de crackers (invasores de sistemas que visam o roubo de informações). Estas informações vão

desde a senha e conta de um usuário de bancos (acesso aos serviços bancários a partir da internet) até a espionagem onde grandes projetos de empresas são furtados.

2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados os conceitos e técnicas envolvidas em ataques a redes de computadores, como também os mecanismos usados para detectar tais eventos.

2.1 TROJAN HORSES

Segundo (CARISSIMI, 2009), *Trojan Horse* ou Cavalo de Tróia é um programa que age como a lenda do Cavalo de Tróia, entrando no computador e liberando uma porta para um possível invasão e é fácil de ser enviado, é só clicar no ID do computador e enviar para qualquer outro computador.

Trojan é um programa que é usado em conexão reversa ou inversa que geralmente é usado para invasão para pegar senha ou outras informações. O Cavalo de Tróia é um programa que tem um pacote de vírus que é usado geralmente para destruir um computador. O conceito nasceu de simples programas que se faziam passar por esquemas de autenticação, em que o utilizador era obrigado a inserir as senhas, pensando que estas operações eram legítimas. Por exemplo, na autenticação de uma *shell*, poderia ser um simples programa numa conta já aberta, e o utilizador que chegasse seria forçado a introduzir a sua password. O *trojan* iria então guardar o password e mascarar a conta (que seria do dono do *trojan*) para que parecesse legítima (a conta da vítima). Entretanto, o conceito evoluiu para programas mais completos.

Os *trojans* atuais são disfarçados de programas legítimos, embora, diferentemente de vírus ou de *worms*, não criam réplicas de si (e esse é o motivo pelo qual o Cavalo de Tróia não é considerado um vírus). São instalados diretamente no computador. De fato, alguns *trojan* são programados para se auto-destruir com um comando do cliente ou depois de um determinado tempo.

Os *trojans* ficaram famosos na Internet pela sua facilidade de uso, fazendo qualquer pessoa possuir o controle de um outro computador apenas com o envio de um arquivo. Por isso os trojans têm fama de ser considerados "ferramentas de *script kid*".

Os trojans atuais são divididos em duas partes:

- O servidor e
- O cliente.

O servidor se instala e se oculta no computador da vítima, normalmente dentro de algum outro arquivo. No momento que esse arquivo é executado, o computador pode ser

acessado pelo cliente, que irá enviar instruções para o servidor executar certas operações no computador da vítima.

Dentro do Servidor Existe 2 conexões:

- Conexão Direta
- Conexão Reversa

A conexão direta tende a precisar do IP da vítima para funcionar, já a reversa tem o IP do dono do *trojan*, fazendo assim a conexão.

Geralmente um *trojan* é instalado com o auxílio de um ataque de engenharia social, com apelos para convencer a vítima a executar o arquivo do servidor, o que muitas vezes acaba acontecendo, dada a curiosidade do internauta, como um email atraindo a pessoa a ver fotos de um artista, pedindo a instalação de um *Plugin*, onde o *Trojan* fica "Hospedado".

2.2 BUFFER OVERFLOW

(TANENBAUM, 2003) define como um transbordamento de dados (em inglês: *buffer overflow*) que acontece quando o tamanho de um *buffer* ultrapassa sua capacidade máxima de armazenamento.

Se o programa não foi adequadamente escrito, esse excesso de dados pode acabar sendo armazenado em áreas de memória próximas, corrompendo dados ou travando o programa, ou mesmo ser executado, que é a possibilidade mais perigosa. Por exemplo, se um programa qualquer possuir uma vulnerabilidade no sistema de *login* por exemplo, pode-se criar um outro programa que fornece caracteres de texto até completar o buffer e que depois envie um executável, que acabaria rodando graças à vulnerabilidade.

2.3 EXPLOITS

Segundo (CARISSIMI, 2009), um *exploit*, em segurança da informação, é um programa de computador, uma porção de dados ou uma sequência de comandos que se aproveita das vulnerabilidades de um sistema computacional – como o próprio sistema operativo ou serviços de interação de protocolos (ex: servidores Web). São geralmente elaborados por hackers como programas de demonstração das vulnerabilidades, a fim de que as falhas sejam corrigidas, ou por *crackers* a fim de ganhar acesso não autorizado a sistemas. Por isso muitos *crackers* não publicam seus *exploits*, conhecidos como 0days, e o seu uso massificado deve-se aos *script kiddies*.

Para um *exploit* atacar, o sistema precisa ter uma vulnerabilidade, ou seja, um meio de comunicação com a rede que possa ser usado para entrar, uma porta ou uma consola.

Um *exploit* muito usado é no sistema RPC do Windows:

- o usuário localiza a porta e envia à porta RPC uma sequência de bytes, que são interpretados como dados pelo servidor;
- quando são recebidos, estes dados deixam propositamente o sistema em pane;
- o sistema passa o controle a estes próprios dados que então são uma sequência de ordem para dominar a CPU.

Desta forma esta sequência de informações toma conta do PC e abre-o para o hacker que aguarda na outra ponta.

No sistema Linux, quando existem vulnerabilidades, sempre são publicadas, como já houve no sistema Apache, Samba ou MySQL, que também apresentam vulnerabilidades e possibilita o controle do PC por um hacker remoto. Com isso um hacker pode ter acesso a todos seus arquivos pessoais, Usando o próprio CMD do Windows.

2.4 BACKDOORS

Segundo (TANENBAUM, 2003), *Backdoor*, também conhecido por Porta dos fundos, é uma falha de segurança que pode existir em um programa de computador ou sistema operacional, que pode permitir a invasão do sistema por um *cracker* para que ele possa obter um total controle da máquina. Muitos *crackers* utilizam-se de um *Backdoor* para instalar vírus de computador ou outros programas maliciosos, conhecidos como *malware*.

Em geral, referindo-se a um *Backdoor*, trata-se de um *Backdoor* que possa ser explorado através da Internet, mas o termo pode ser usado de forma mais ampla para designar formas furtivas de se obter informações privilegiadas em sistemas de todo tipo. Por exemplo: o Clipper Chip, dispositivo de criptografia do Governo Estados Unidos, possui um *Backdoor* embutido pelo próprio Governo que permite recuperar as informações codificadas anteriormente com o dispositivo.

Existem casos em que, teoricamente, o programa de computador pode conter um *Backdoor* implementado no momento em que ele foi compilado. Geralmente esse recurso é interessante quando um software deve realizar operações de atualização ou validação, mas essa é uma técnica já defasada, pois os programadores preferem utilizar protocolos de rede do sistema operacional (como o SSH ou o Telnet, embora eles também utilizem os tradicionais protocolos TCP/IP, UDP ou ainda o FTP).

2.5 DOS (DENIAL OF SERVICE)

Um ataque de negação de serviço (também conhecido como DoS, um acrônimo em inglês para *Denial of Service*), é uma tentativa em tornar os recursos de um sistema indisponíveis para seus utilizadores. Alvos típicos são servidores web, e o ataque tenta tornar as páginas hospedadas indisponíveis na WWW. Não se trata de uma invasão do sistema, mas sim da sua invalidação por sobrecarga. Os ataques de negação de serviço são feitos geralmente de duas formas:

- Forçar o sistema vítima a reinicializar ou consumir todos os recursos (como memória ou processamento por exemplo) de forma que ele não pode mais fornecer seu serviço.
- Obstruir a mídia de comunicação entre os utilizadores e o sistema vítima de forma a não comunicarem-se adequadamente.

2.5.1 Ataque Distribuído

Um ataque distribuído de negação de serviço é conhecido como DDoS, um acrônimo em inglês para *Distributed Denial of Service*. Um computador mestre (denominado "*Master*") pode ter sob seu comando até milhares de computadores ("*Zombies*" - zumbis). Repare que nestes casos, as tarefas de ataque de negação de serviço são distribuídas a um "exército" de máquinas escravizadas.

O ataque consiste em fazer com que os Zumbis (máquinas infectadas e sob comando do Mestre) se preparem para acessar um determinado recurso em um determinado servidor em uma mesma hora de uma mesma data. Passada essa fase, na determinada hora, todos os zumbis (ligados e conectados à rede) acessarão ao mesmo recurso do mesmo servidor. Como servidores web possuem um número limitado de usuários que pode atender simultaneamente ("*slots*"), o grande e repentino número de requisições de acesso esgota esse número de *slot*, fazendo com que o servidor não seja capaz de atender a mais nenhum pedido. Dependendo do recurso atacado, o servidor pode chegar a reiniciar ou até mesmo ficar travado.

2.6 SISTEMA DE DETECÇÃO DE INTRUSÃO

Os sistemas de detecção de intrusão, ou simplesmente IDS (*Intrusion Detection System*) são aplicações que monitoram e correlacionam uma série de eventos que ocorrem em uma rede e em sistemas computacionais, em geral. São exemplos de eventos: horários em que serviços de rede e do próprio sistema operacional foram requisitados, processos e usuários

que os requisitaram, tentativas fracassadas de *login* no sistema, padrões de acesso e de uso de rede, protocolos etc. Esses eventos, à medida que ocorrem, são registrados em arquivos especiais denominados de *logs* do sistema. O IDS examina os *logs* procurando por eventos, ou por uma combinação deles, que possam indicar um comportamento suspeito na rede ou em uma máquina. Se isso ocorrer, os administradores são alertados para verificar se o que está ocorrendo é realmente uma tentativa de invasão ou apenas um evento fora do comum.

Os sistemas IDS, em função dos tipos de eventos que registram e de sua localização em um sistema computacional, são classificados como: baseado em hospedeiro (host) e baseado em rede. Um IDS baseado em hospedeiro executa diretamente em uma máquina monitorando e registrando atividades feitas pelo sistema operacional e por processos em geral. Eles são vinculados às diferentes chamadas de sistema, registrando quando e quem as faz, e o sucesso ou não da sua execução. Nesse tipo de IDS também é comum a verificação de datas de modificação de arquivo. O segundo tipo de IDS, baseado em rede, monitora o tráfego de uma rede em busca de padrões fora do comum que poderiam indicar ataques, como por exemplo, de negação de serviços através do SYN *flooding*. Em ambos tipos de IDS, os administradores podem verificar os arquivos de registros para determinar se houve ou não uma invasão, ou sua tentativa, e, caso isso tenha ocorrido, tentar rastrear sua origem.

Um dos problemas para se utilizar um IDS é configurá-lo adequadamente. É responsabilidade dos administradores de rede determinar os eventos de interesse, o que é considerado atividade suspeita e que eventos podem (ou devem) ser correlacionados para indicar a suspeita de uma invasão. Normalmente, essa tarefa é feita em três fases: montagem de perfis de ameaças, identificação de vulnerabilidades e desenvolvimento de soluções e planos de segurança. Existem softwares que auxiliam os administradores de redes a executar essas fases, porém é de fundamental importância o conhecimento e a experiência dos administradores de rede para alimentar adequadamente esses softwares.

2.7 TIPOS DE DETECÇÃO

2.7.1 Baseados na Estação

Os sistemas baseados em estação foram os primeiros tipos de IDS a serem desenvolvidos e implementados. Eles começaram no início dos anos 80. Os IDS de estação podem ser instalados em diferentes tipos de máquinas, como servidores, estações de trabalho e notebook. Esses sistemas coletam e analisam dados originados nos computadores que hospedam um serviço, como servidor de web. Uma vez que o dado é agregado em um dado

computador, ele pode ser analisado localmente ou enviado para uma máquina central de análise. Um exemplo de sistema baseado em estação são programas que operam em um sistema e recebem aplicações ou logs do sistema operacional. Esses programas são altamente efetivos para detectar abusos de dentro. Residindo em sistemas confiáveis, eles estão perto dos usuários autênticos da rede. Se um desses usuários tentar realizar alguma atividade desautorizada, o IDS de estação geralmente detecta e coleta a maioria das informações possíveis, da maneira mais rápida. Esse tipo de sistema também é efetivo em detectar modificações em arquivos não autorizadas.

2.7.1.1 Vantagens dos IDS baseados em estação

Como vantagens dos IDS baseados em estação, pode-se citar:

- Como podem monitorar eventos localmente, os IDS de estação conseguem detectar ataques que não conseguem ser detectados por IDS de rede;
- Trabalham em ambientes onde o tráfego seja criptografado, desde que os dados sejam encriptados na estação antes do envio ou decryptados nos hosts após o recebimento;
- Não são afetados por switches;

2.7.1.2 Desvantagens dos IDS baseados em estação

Como desvantagens, pode-se citar:

- São de difícil monitoramento, já que em cada estação deve ser instalado e configurado um IDS;
- Podem ser desativados por DoS;
- Recursos computacionais são consumidos nas estações monitoradas, com diminuição do desempenho;
- O IDS pode ser atacado e desativado, escondendo um ataque, se as fontes de informações residirem na estação monitorada;

2.7.2 Baseados em Rede

A maioria dos IDS comerciais é baseada em rede. Os IDS baseados em rede analisam pacotes de dados que trafegam pela rede. Esses pacotes são examinados e algumas vezes comparados com dados empíricos para verificar a sua natureza: maliciosa ou benigna. Pelo fato de serem responsáveis pelo monitoramento da rede, ao invés de uma simples estação, os IDS de rede tem a tendência de serem mais distribuídos do que os IDS baseados em estação. Softwares, ou em alguns casos aparelhos de hardware, residem em um ou mais sistemas conectados numa rede, e são usados para analisar dados, como pacotes de rede. Ao invés de analisar informação que reside e é originada em um computador, IDS de rede usa técnicas como “*packet-sniffing*”, para pegar dados do TCP/IP ou pacotes de outros protocolos passando sozinhos pela rede. Essa vigilância das conexões entre computadores faz com que IDS de rede sejam ótimos em detectar tentativas de acesso por fora da rede confiável. Os IDS baseados em rede normalmente consistem em um conjunto de sensores implantado em diversas partes da rede, monitorando o tráfego, realizando análises e relatando ataques a uma central. Em geral, IDS de rede são melhores em detectar as seguintes atividades:

- Acesso desautorizado de fora: quando um usuário desautorizado loga com sucesso, ou tenta logar, são melhores monitorados por IDS de estação. Entretanto, detectar usuários desautorizados antes da tentativa de log é mais bem realizado por IDS de rede;
- *Denial of Service* (Negação de Serviço): os pacotes que iniciam esses ataques podem melhor ser percebidos com o uso de IDS de rede.

2.7.3 Baseados em Pilhas

Essa é a tecnologia de IDS mais nova e varia bastante de vendedor para vendedor. Este tipo de IDS funciona integrando próximo à pilha TCP/IP, permitindo que pacotes sejam vistos quando eles fazem seu caminho através das camadas OSI. Vendo os pacotes desse jeito permite ao IDS puxar pacotes da pilha antes que o sistema operacional ou a aplicação tenham a chance de processar os pacotes. Esse IDS deve olhar o tráfego entrando e saindo do sistema. Monitorando apenas pacotes de rede destinados apenas para uma simples estação, o princípio é fazer com que o IDS tenha baixo overhead suficiente para que todos os sistemas na rede possam rodar esse IDS.

2.8 MÉTODOS DE DETECÇÃO

Para cada um dos tipos de detecção, existem duas técnicas bastante utilizadas para detectar intrusos: detecção de anomalias e detecção de assinaturas.

2.8.1 Detecção de Anomalias

O IDS que usa essa técnica estabelece uma base com padrões de uso normal, e tudo o que desviar bastante disso será marcado como uma possível intrusão. O que é considerado anomalia pode variar, mas normalmente, qualquer incidente que ocorra com frequência de mais ou menos dois desvios padrões da norma estatística gera uma preocupação. Esses detectores constroem um padrão de comportamento para os usuários, hosts e conexões de rede.

2.8.1.1 Vantagens da detecção por anomalias

Como vantagens deste método de detecção, pode-se citar:

- Como detecta comportamentos não usuáios podem detectar um ataque sem conhecimento prévio do mesmo;
- Gera informações que podem ser usadas para definir assinaturas.

2.8.1.2 Desvantagens da detecção por anomalias

Como desvantagens deste método de detecção, pode-se citar:

- É comum produzir grande numero de alarmes falsos, devido a comportamentos imprevisíveis dos usuáios e de sistemas;
- Devem ser realizadas muitas sessões para coletar dados para o sistema, com o intuito de caracterizar os padrões normais de comportamento;

2.8.2 Detecção de Assinaturas

Detecção de assinaturas envolve a procura em tráfego de rede de bytes ou seqüências de pacotes conhecidos como maliciosos. Uma vantagem chave desse método é que assinaturas são fáceis de serem desenvolvidas e entendidas se sabida o comportamento da rede que se esta tentando identificar. Os eventos gerados por um IDS baseado em assinaturas podem comunicar o que causou o alerta. Também, testes de correlacionamento podem ser executados bem rápidos em modernos sistemas, então a energia necessária para executar essas checagens

pode ser mínima para um conjunto de regras. Como exemplo, se o sistema só se comunicar via DNS, ICMP e SMTP, todas as outras assinaturas podem ser removidas.

Para sistemas baseados em estação, um exemplo de assinatura é três logins falhos. Já para IDS de rede, uma assinatura pode ser simplesmente teste padrão que combina com uma porção de pacotes de redes, como por exemplo, assinaturas de pacotes e/ou assinaturas de cabeçalho que possam indicar ações não autorizadas.

2.8.2.1 Vantagem da detecção baseada em assinaturas

Como vantagem deste método de detecção, pode-se citar:

- Comparadas às detecções por anomalia, são muito mais eficientes, gerando um numero bem menor de alarmes falsos.

2.8.2.2 Desvantagens da detecção baseada em assinaturas

Como desvantagens deste método de detecção, pode-se citar:

- Somente identificam ataques conhecidos, por isso, necessita de uma constante atualização no conjunto de assinaturas;
- Como grande parte das assinaturas são muito específicas, ataques variantes não são detectados.

3 ESTUDO DE CASO

Neste capítulo será feito um estudo de caso de implantação de uma ferramenta de IDS em uma rede fictícia, a fim de aplicar os conceitos envolvidos na detecção de intrusões e avaliar a eficiência de tais ferramentas.

3.1 A REDE

Na Figura 1 é apresentada a topologia de uma rede corporativa na qual será feito o estudo de caso.

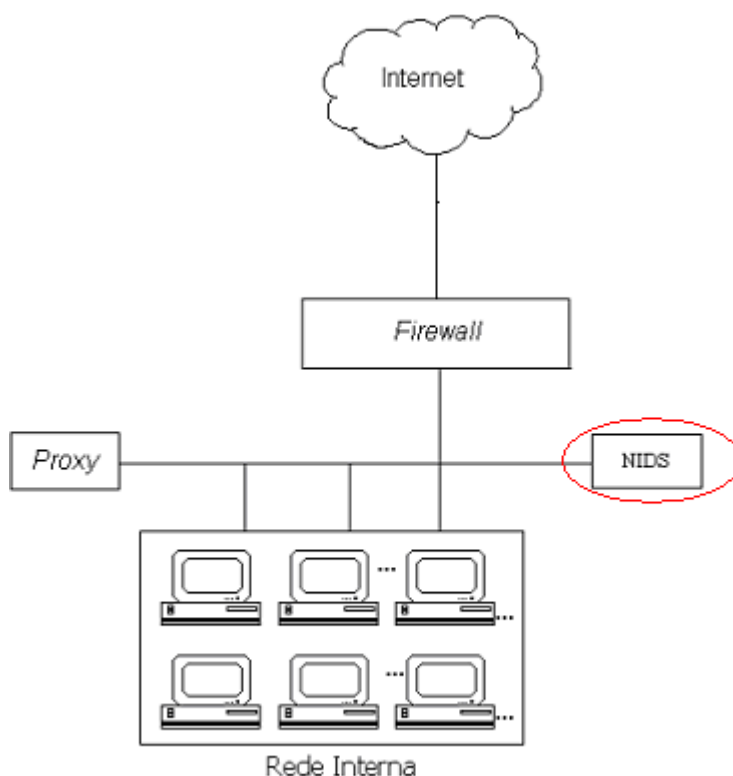


Figura 1: Localização do IDS na rede do estudo de caso

3.2 A FERRAMENTA SNORT

O Snort é um software livre de detecção de intrusão para rede (NIDS) desenvolvido inicialmente por Martin Roesch, capaz de desenvolver análise de tráfego em tempo real e registro de pacote em redes IP. Executa análise de protocolo, busca/associa padrões de conteúdo e pode ser usado para detectar uma variedade de ataques, tais como *buffer overflows*, *stealth port scans*, ataques CGI, SMB probes, OS fingerprinting, entre outras. Esta

ferramenta é suportada em arquiteturas RISC e CISC e em plataformas das mais diversas, como os vários sabores de Linux (RedHat, Debian, Slackware, Mandrake, etc.), OpenBSD, FreeBSD, NetBSD, Solaris, SunOS, HP-UX, AIX, IRIX, Tru64, MacOS X.

O Snort monitora o tráfego de pacotes em redes IP, realizando análises em tempo real sobre diversos protocolos (nível de rede e aplicação) e sobre o conteúdo (hexa e ASCII). Outro ponto positivo desse software é o grande número de possibilidades de tratamento dos alertas gerados. O subsistema de registro e alerta é selecionado em tempo de execução através de argumentos na linha de comando, são três opções de registro e cinco de alerta. O registro pode ser configurado para armazenar pacotes decodificados e legíveis em uma estrutura de diretório baseada em IP, ou no formato binário do tcpdump em um único arquivo. Para um incremento de desempenho, o registro pode ser desligado completamente, permanecendo os alertas. Já os alertas podem, ser enviados ao syslog, registrados num arquivo de texto puro em dois formatos diferentes, ou ser enviados como mensagens WinPopup usando o smbclient.

Os alertas podem ser enviados para arquivo texto de forma completa e o alerta rápido. O alerta completo escreve a mensagem de alerta associada à regra e a informação do cabeçalho do pacote até o protocolo de camada de transporte. A opção de alerta rápido escreve um subconjunto condensado de informação do cabeçalho alerta.

O Snort poderá assumir três modalidades a seguir:

- *Sniffer*: Está modalidade simplesmente captura os pacotes e imprime continuamente no console;
- *Packet logger*: Registra os pacotes capturados no disco rígido;
- *Network Intrusion Detection System*: Esta modalidade é a mais complexa e versátil, permitindo que o Snort analise o tráfego da rede de encontro a regras definidas pelo usuário, executando diversas ações baseadas em suas regras.

3.3 OUTROS SOFTWARES REQUERIDOS

Há cinco pacotes de software preliminares que produzem a topologia do NIDS. O servidor web Apache, o servidor de banco de dados MySQL, ACID e o próprio Snort. Esta topologia supõe que seus sensores estarão funcionando em um hardware dedicado separado da base de dados e do console ACID. Nas seções a seguir estão descrições breves de cada um dos pacotes e de sua finalidade na topologia.

A seguir uma lista dos softwares livres requeridos e as versões que foram utilizadas no estudo de caso:

- Debian lenny (<ftp://ftp.debian.org>)
- Snort v2.8.5.3 (<http://www.snort.org/dl/>)
- MySQL v3.23.52 (<http://www.mysql.com/downloads/mysql-3.23.html>)
- ACID 0.9.6B21 (<http://acidlab.sourceforge.net/>)
- PHP v5.* (<http://www.php.com>)
- ADODB v2.31 (<http://php.weblogs.com/adodb>)
- Apache Webserver 2.0 (<http://www.apache.org/>)

3.3.1 Apache Web Server

Este é o servidor de páginas web, o qual é escolha padrão para a maioria dos *websites* que são acessados na Internet. A única finalidade da Apache é de hospedar o console ACID baseado na Web.

3.3.2 MySQL Server

MySQL é um servidor de base de dados baseado em SQL para uma variedade de plataformas e é a plataforma mais suportada para armazenar alertas do Snort. Todos os alertas do IDS que são provocados por nossos sensores e são armazenados na base de dados do MySQL.

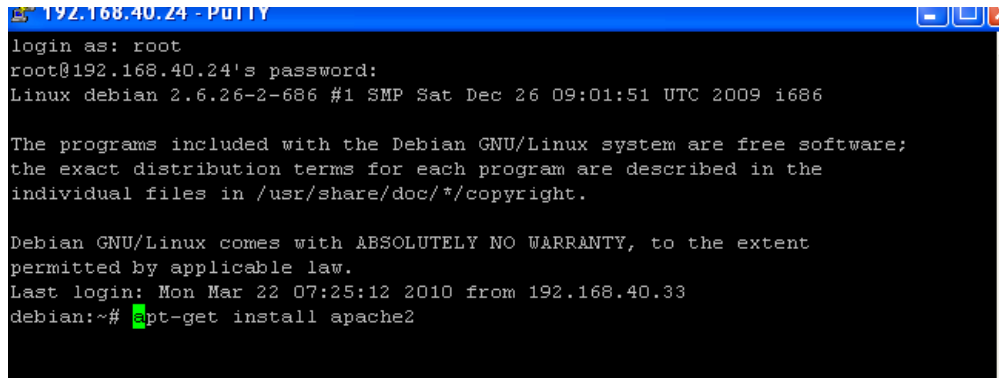
3.3.3 Analysis Console for Intrusion Databases (ACID)

O ACID (console de análise para base de dados do intruso) é uma aplicação baseada na web para que os *logs* (registros) do Firewall que possam ser vistos. Este é o lugar onde toda a informação do sensor é consolidada para a visão.

3.4 INSTALAÇÃO DOS SOFTWARES REQUERIDOS

Antes de iniciar a instalação e configuração do Snort e de alguns dos seus utilitários, é necessário alguns passos anteriores.

1. Primeiro, deve-se instalar o Apache2 (Figura 2), o MySQL (Figura 3), o PHP5 (Figura 4) e a extensão do PHP para bancos de dados MySQL (Figura 5). Todas as instalações foram feitas via comando `apt-get` do Linux Debian.



```

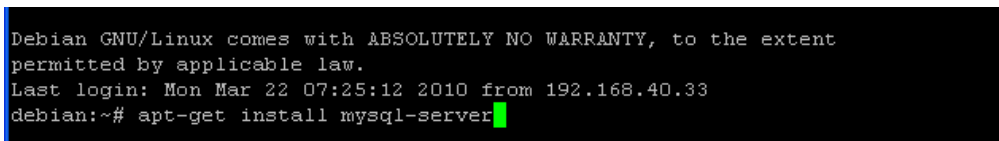
192.168.40.24 - PuTTY
login as: root
root@192.168.40.24's password:
Linux debian 2.6.26-2-686 #1 SMP Sat Dec 26 09:01:51 UTC 2009 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 22 07:25:12 2010 from 192.168.40.33
debian:~# apt-get install apache2

```

Figura 2: Instalação do Apache 2



```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 22 07:25:12 2010 from 192.168.40.33
debian:~# apt-get install mysql-server

```

Figura 3: Instalação do MySQL server



```

192.168.40.24 - PuTTY
debian:~# apt-get install php5

```

Figura 4: Instalação do PHP 5



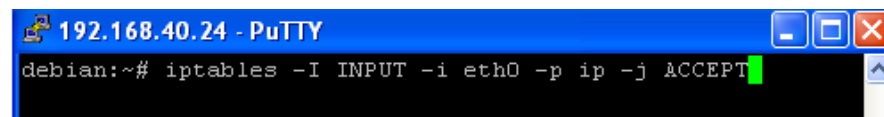
```

192.168.40.24 - PuTTY
debian:~# apt-get install php5-mysql

```

Figura 5: Instalação do suporte a MySQL no PHP

2. Configurar o Firewall do Linux (através do comando iptables), conforme Figura 6, para que, especialmente na fase de testes, não haja bloqueio ao tráfego IP.



```

192.168.40.24 - PuTTY
debian:~# iptables -I INPUT -i eth0 -p ip -j ACCEPT

```

Figura 6: Configurar o iptables para liberação do tráfego IP

3. Testar o Apache com o PHP, criando o arquivo /var/www/index.php com o código PHP da Figura 7. Após isso, deverá ser possível acessar através de um browser no próprio servidor a URL <http://localhost/index.php> e visualizar as informações da configuração do PHP neste.

```
<?php
```

```
?> phpinfo();
```

Figura7: Pequeno código PHP para teste do servidor Apache com PHP

4. Instalar o ADODB, que é uma biblioteca de classes utilizada para interação da linguagem PHP com o banco de dados MySQL e está disponível em <http://prdownloads.sourceforge.net/adodb/>

3.5 INSTALAÇÃO DO SNORT

Para instalação do Snort, os seguintes passos são necessários:

1. Executar o download do Snort e do PCRE nos sites:
 - <http://www.snort.org>
 - <http://prdownloads.sourceforge.net/pcre/>
2. Através do utilitário “Adept” verificar se estão instalados os seguintes componentes. Caso necessário proceder a sua instalação via Adept.
 - libpcap0.8
 - libpcap0.8-dev
 - libpcre3
 - libpcre3-dev
3. Instalar o PCRE, que é um conjunto de funções que implementam a correspondência de padrão de expressões regulares usando a mesma sintaxe e semântica do Perl 5, através dos comandos da Figura 8.

```
# sudo tar -xvzf pcre-6.3.tar.gz
# cd pcre-6.3
# sudo ./configure
# sudo make
# sudo make install
```

Figura 8: Instalação do PCRE

4. Instalar o Snort, através dos comandos da Figura 9.

```
# sudo tar -xvzf snort-2.4.4.tar.gz
# cd snort-2.4.4
# sudo ./configure --with-mysql=<localização_do_mysql>
# sudo make
# sudo make install
```

Figura 91: Instalação do Snort

Se ocorrer a seguinte mensagem de erro: “*snort: error while loading shared libraries: libpcrc.so.0: cannot open shared object file: No such file or directory*”, deverá criar-se um link simbólico, através do seguinte comando:

```
# sudo ln -s /usr/local/lib/libpcrc.so.0 <localização do arquivo>
```

Por exemplo:

```
# sudo ln -s /usr/local/lib/libpcrc.so.0 /usr/lib/libpcrc.so.0
```

5. Efetuar o download das regras e proceder a sua extração para o diretório “/etc/snort/rules”.
6. Configurar os seguintes parâmetros no arquivo “/etc/snort/snort.conf”, vistos na Figura 10.

```
var HOME_NET any      # Para capturar todos as redes
var EXTERNAL_NET !$HOME_NET # Tudo o que não for HOME_NET é externo
var RULE_PATH /etc/snort/rules      # caminho correto para as regras)
--preprocessor
output database: log, mysql, user=snort password=<pwd_escolhida>
dbname=snort
host=localhost
```

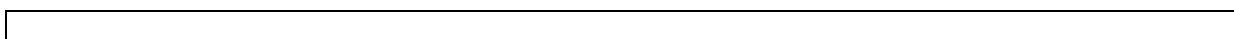
Figura 20: Configurações dos parâmetros do Snort

7. Configurar a base de dados do Snort no MySQL, com os comandos da Figura 11.

```
# mysql
mysql> SET PASSWORD FOR root@localhost=PASSWORD('password');
>Query OK, 0 rows affected (0.25 sec)
mysql> create database snort;
>Query OK, 1 row affected (0.01 sec)
mysql> grant INSERT,SELECT on root.* to snort@localhost;
>Query OK, 0 rows affected (0.02 sec)
mysql> SET PASSWORD FOR
snort@localhost=PASSWORD('password_do_snort.conf');
>Query OK, 0 rows affected (0.25 sec)
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort@localhost;
>Query OK, 0 rows affected (0.02 sec)
mysql> grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort;
>Query OK, 0 rows affected (0.02 sec)
mysql> exit
>Bye
```

Figura 31: Configuração da base de dados MySQL

8. Executar os comandos da Figura 12 para criar as tabelas no banco de dados.

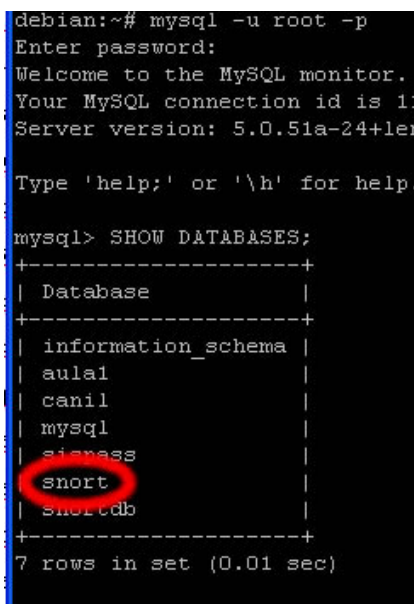


```
# mysql -u root -p < ~/mnt/snort-2.8.5.3/schemas/create_mysql snort
Enter password: <mysql_root_password>
```

Figura 42: Criando as tabelas na base de dados do snort

9. Verificar se a base de dados do Snort foi criada corretamente, conforme Figura 13.

Na Figura 14 são listadas as tabelas do banco de dados do snort.

A terminal window showing a MySQL command prompt. The user has entered 'mysql -u root -p' and provided a password. The prompt shows 'Welcome to the MySQL monitor. Your MySQL connection id is 1. Server version: 5.0.51a-24+le'. The user has entered 'SHOW DATABASES;' and the output is a table with 7 rows. The row for 'snort' is circled in red.

```
debian:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.
Your MySQL connection id is 1
Server version: 5.0.51a-24+le
Type 'help;' or '\h' for help

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| aula1 |
| canil |
| mysql |
| siavass |
| snort |
| snortdb |
+-----+
7 rows in set (0.01 sec)
```

Figura 53: Observando o banco de dados do Snort criado no MySQL



```

192.168.40.24 - PuTTY
mysql> use snort
Reading table information for your current table
You can turn off this behavior by setting 'table_info' to 0 in the
'edit' command output

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_snort |
+-----+
| acid_ag          |
| acid_ag_alert    |
| acid_event       |
| acid_ip_cache    |
| data             |
| detail           |
| encoding         |
| event            |
| icmphdr          |
| iphdr            |
| opt              |
| reference        |
| reference_system |
| schema           |
| sensor           |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+-----+
20 rows in set (0.00 sec)

mysql>

```

Figura 64: Observando as tabelas da base de dados do Snort

Antes de instalar o Snort, é necessário instalar a biblioteca libpcap, que é necessária para a captura dos pacotes que passam pelo IDS. A libpcap é componente essencial para o Snort. Por questões de desempenho e segurança é interessante compilá-lo, pois qualquer vulnerabilidade encontrada pode ser atualizada de maneira fácil, baixando o fonte da nova versão e compilando, sem esperar pelo pacote da distribuição Linux. A Figura 15 mostra os comandos para compilação deste componente.

```

# apt-get install build-essential
# apt-get -y install libpcap0.8-dev

```

Figura 75: Instalação da biblioteca libpcap

Após a instalação e configuração do MySQL e do Snort, deve-se remover o arquivo `/etc/snort/db-pending-config`.

10. Finalmente, instalar o snort, conforme comando da Figura 16.

```
# apt-get install snort-mysql
```

Figura 86: Instalação do Snort

3.6 CONFIGURAÇÃO DO SNORT

O arquivo de configuração do snort é o arquivo `snort.conf`. A configuração deste arquivo é dividido em 4 partes:

1. Configuração das variáveis de rede
2. Configuração dos pré-processadores
3. Configuração dos *plugins* de saídas
4. Customização das regras

3.6.1 Configurações das variáveis de rede

A sintaxe das definições de variáveis podem ser entendidas da seguinte forma:

```
var <NOME-DA-VARIÁVEL> <valor>
```

O nome da variável sempre é definido com letras maiúsculas. O valor que a variável irá guardar pode ser uma palavra chave ou qualquer outra variável definida anteriormente. No último caso deve-se utilizar o nome da variável com o caractere \$.

Na Figura 17 está apresentada a configuração padrão de instalação do snort.

```
var HOME_NET any
var EXTERNAL_NET any
var DNS_SERVERS $HOME_NET
var SMTP_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var TELNET_SERVERS $HOME_NET
var HTTP_PORTS 80
var SHELLCODE_PORTS !80
var ORACLE_PORTS 1521
var RULE_PATH ../rules
```

Figura 97: Variáveis de configuração do Snort

A configuração padrão do snort gera muito log dificultando a análise, pois se trata de uma configuração genérica. É sempre importante personalizar as variáveis para diminuir a quantidade de informações gravadas em logs. O snort mal-configurado pode gravar muitas

informações ociosas. Abaixo está explicada a finalidade de cada variável, configure-as de acordo com sua rede.

- **var HOME_NET any**: A variável HOME_NET determina a rede privada que o snort irá monitorar. O valor any determina que qualquer rede será monitorada, mais precisamente, determina que todos os dados acessíveis pela máquina onde o snort está instalado será monitorado. Lembre-se: o snort é um tipo de farejador – sniffer – e, portanto captura todos os dados que trafegar no barramento local. Você poderá personalizar esta variável da seguinte forma: var HOME_NET 10.1.0.0/16, onde 10.1.0.0/16 é o endereço de sua rede local.
- **var EXTERNAL_NET any**: Uma boa configuração para esta variável é o valor any. Assim, o snort irá monitorar qualquer pacote externo que tiver destino sua rede.
- **var DNS_SERVERS \$HOME_NET**: Esta variável determina os servidores DNS de sua rede a serem analisados pelo snort. Observe que a configuração padrão utiliza como valor a variável \$HOME_NET. Assim, o valor definido para a variável DNS_SERVERS será o mesmo da variável HOME_NET. É aconselhável alterar o valor desta variável e definir a lista de servidores de sua rede conforme o exemplo: var DNS_SERVERS [10.1.0.2,10.1.0.3].
- **var SMTP_SERVERS \$HOME_NET**
- **var HTTP_SERVERS \$HOME_NET**
- **var SQL_SERVERS \$HOME_NET**
- **var TELNET_SERVERS \$HOME_NET**: A configuração destas quatro últimas variáveis é similar ao exemplo anterior, modificando apenas o tipo de servidor. Neste caso deve ser definida a lista de servidores de email – SMTP_SERVERS, servidores Web - HTTP_SERVERS, servidores de banco de dados SQL – SQL_SERVER e servidores Telnet – TELNET_SERVERS.
- **var HTTP_PORTS 80**: Determina a porta do servidor Web a ser utilizado para análise do snort. Somente será necessário alterar esta opção, se o seu servidor Web estiver sendo executado em uma porta diferente da 80.
- **var ORACLE_PORTS 1521**: Determina a porta do servidor de banco de dados Oracle para análise do snort.
- **var SHELLCODE_PORTS 80**: Portas que deseja-se procurar por códigos em Shell, esta variável permite o snort procurar por pacotes que contenham códigos em shell scripts. O valor 80 determina que será monitorado todas as portas exceto a porta 80.

Páginas Web podem ser utilizadas em conjunto com script Shell para criar estruturas de condições e todos os recursos que a linguagem shell permite. Este é o motivo pelo qual não há a necessidade de monitorar a porta 80.

- **var RULE_PATH ./rules:** Define o diretório que contém os arquivos de regras do snort. Certifique-se de que o caminho destas regras esteja configurada corretamente. Pode-se passar o caminho completo conforme apresentado: `var RULE_PATH /usr/local/snort/rules.`

3.6.2 Configuração dos pré-processadores

A função dos pré-processadores é analisar determinado tráfego na rede. Através dos pré-processadores é possível analisar as integridades dos pacotes assim como detectar vários tipos de varreduras. A sintaxe dos preprocessadores está descrita abaixo:

```
preprocessor <nome_do_processador>: <opções_de_configuração>
```

Na Figura 18 estão listados os pré-processadores que estão habilitados por padrão:

```
preprocessor frag2
preprocessor stream4: detect_scans, disable_evasion_alerts
preprocessor http_decode: 80 unicode iis_alt_unicode
```

Figura 108: Lista de pré-processadores na configuração do Snort

- **preprocessor frag2** : este pré-processador irá detectar pacotes desfragmentados geralmente utilizados em ataques DoS. Ataques como “ping da morte” - Ping of Death – são detectados por este pré-processador.
- **preprocessor stream4: detect_scans, disable evasion alerts** : Faz análise e remontagem das cadeias TCP. A opção `detect_scans` permite que seja identificada varredura silenciosas – stealth scan.
- **preprocessor http_decode: 80 unicode iis alt unicode** : Normaliza requisições HTTP de uma máquina remota. A opção `unicode` normaliza unicode e permite que o Snort monitore ataques baseado no unicode.

3.6.3 Configuração dos plugins de saída

Esta configuração permite definir como serão tratados os logs do snort. Todos os plugins de saídas são comentados por padrão. A configuração dos plugins não é obrigatória, o snort por padrão irá gravar seu logs através de arquivo texto. A sintaxe dos plugins é a seguinte:

```
output <name_of_plugin>: <configuration_options>
```

A seguir dois exemplos de configuração que pode ser utilizado:

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

```
output database: log, mysql, user=root password=test dbname=db
                  host=localhost
```

O primeiro exemplo configura o Snort para utilizar o syslog para gravar e formatar seus logs. Já o segundo exemplo permite o Snort enviar seus logs para um banco de dados, que neste caso é o MySQL.

3.7 CUSTOMIZAÇÃO DAS REGRAS

Vários arquivos de regras acompanham o pacote do Snort. Entretanto, nem todos são necessários utilizar. Para verificar quais regras existem, no diretório rules há os arquivos contendo as regras utilizadas pelo Snort. As regras a serem utilizadas irão depender de quais serviços é disponibilizado em sua rede. O Snort pode analisar quatro protocolos: IP, TCP, UDP e ICMP.

Alguns exemplos demonstrando as regras abaixo:

Esta regra cria um alerta para todo o tráfego UDP proveniente das portas 1024 até 2048 de qualquer terminal. A palavra “any” é utilizada para aplicar a regra a todos os pacotes. A opção “msg” é usada para adicionar uma mensagem

```
Alert udp any 1024:2048 -> any any (msg: “TRÁFEGO UDP 1024 - 2048”);)
```

O exemplo abaixo Registra todo o tráfego UDP exceto aquele originado na porta 53.

```
log udp any !53 -> any any log udp
```

O outro exemplo é negar várias portas:

```
log udp any ![53:554] > any any log udp
```

O comando da Figura 19 é para iniciar o serviço Snort.



Figura 19: Comando para iniciar o Snort

3.8 INSTALAÇÃO E CONFIGURAÇÃO DO ACID

Na Figura 20, tela principal do ACID, é analisado o tráfego TCP, UDP e ICMP e o total de alertas e sensores que foram detectados e armazenados no banco de dados.

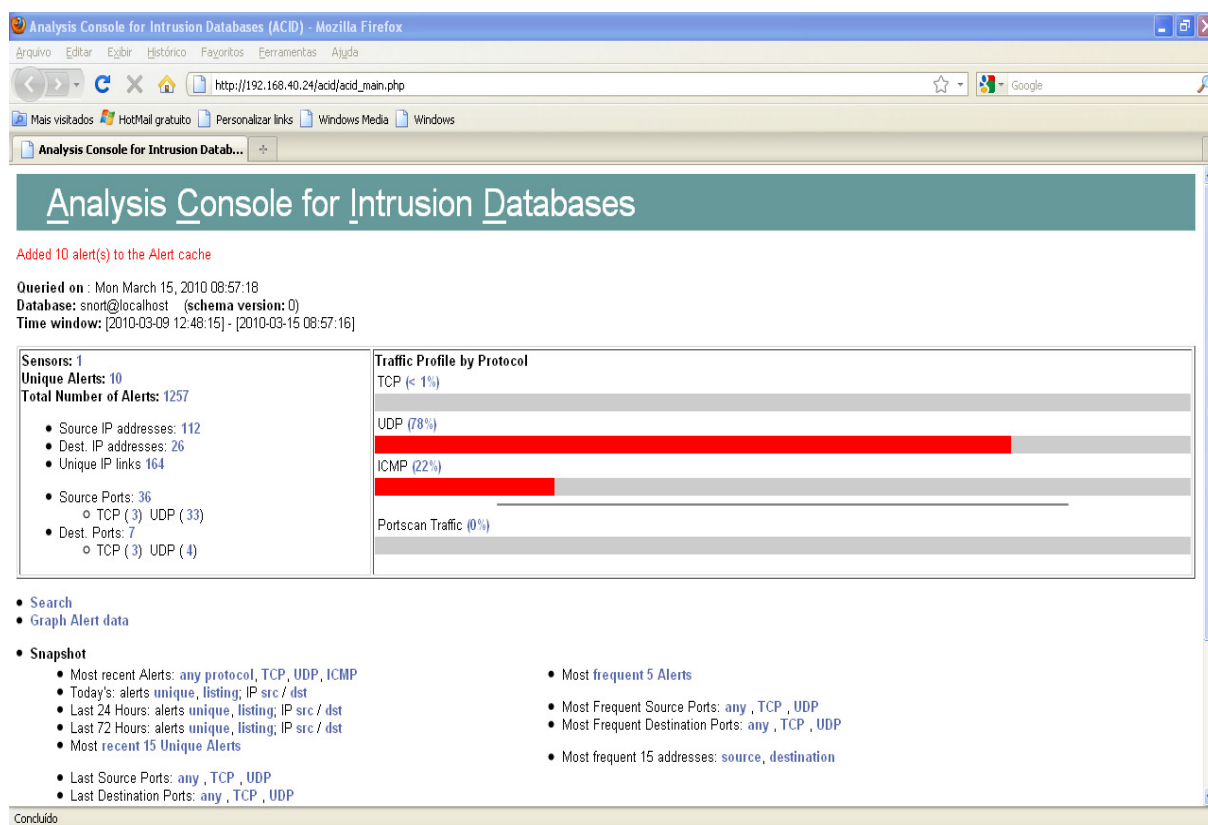


Figura 110: Interface principal do ACID

Na Figura 21 é mostrado um relatório do que já foi capturado.

ACID: Alert Listing - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://192.168.40.24/acid/acid_stat_alerts.php

Mais visitados Guia rápido Últimas notícias

ACID: Alert Listing [Dicas-L] Snort + ACID + MySQL no Slac...

ACID

Alert Listing

Home Search | AG Maintenance

[Back]

Added 0 alert(s) to the Alert cache

Queried DB on : Mon April 19, 2010 11:40:30

Meta Criteria any

IP Criteria any

Layer 4 Criteria none

Payload Criteria any

Displaying alerts 1-10 of 10 total

	< Signature >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
☐	10	2 (1%)	1	2	1	2010-03-31 13:43:18	2010-04-15 13:36:56
☐	12	50 (13%)	1	4	2	2010-04-05 06:58:35	2010-04-07 13:09:39
☐	14	11 (3%)	1	6	3	2010-03-23 11:17:24	2010-04-19 09:46:02
☐	16	225 (59%)	1	12	13	2010-03-22 11:05:10	2010-04-19 11:25:01
☐	17	50 (13%)	1	11	14	2010-03-22 11:04:55	2010-04-19 11:24:45
☐	19	15 (4%)	1	3	3	2010-03-23 10:01:24	2010-04-06 14:00:45
☐	20	23 (6%)	1	6	2	2010-03-23 10:21:22	2010-04-19 09:00:19
☐	21	4 (1%)	1	1	1	2010-04-05 13:04:11	2010-04-05 13:04:25
☐	7	1 (0%)	1	1	1	2010-03-22 13:14:50	2010-03-22 13:14:50
☐	9	2 (1%)	1	1	1	2010-03-23 07:00:27	2010-03-30 07:18:54

{ action } Action Selected ALL on Screen

[Loaded in 0 seconds]

ACID v0.9.6b23 (by Roman Danyliw as part of the AircERT project)

Concluido

Iniciar

ACID: Alert Listing ...

Intrusion detection ...

Fam

PROJETO_DE_MON...

livro snort_10.pdf - ...

Microsoft PowerPol...

192.168.40.24 - Pu...

PT

14:44

Figura 121: Interface do ACID com alertas

4 CONCLUSÃO

Como incidentes de segurança se tornam mais numerosos, ferramentas de IDS são cada vez mais necessários. Eles completam o arsenal de segurança, trabalhando em conjunto com outras ferramentas de informação de segurança, como firewalls, e permitir a supervisão completa de todas as atividades da rede. Esta informação pode, por sua vez, ajudar a determinar o uso indevido da rede, sua natureza e sua origem. Estas ferramentas de detecção de intrusão usam várias técnicas para ajudar a determinar o que qualifica como uma intrusão versus tráfego normal. Usa-se um sistema de detecção de anomalias, detecção de desvio, o acompanhamento de alvo que geralmente caem em uma das duas categorias: baseados em rede ou baseados em host. Cada categoria tem pontos fortes e fracos que devem ser medidos contra os requisitos para cada ambiente alvo distinto. Idealmente, as melhores ferramentas IDS combinam ambas as abordagens em um console de gerenciamento. Dessa forma, o usuário recebe uma cobertura global, certificando de se proteger contra as ameaças possíveis. Seja qual for a escolha, se é baseado em host, baseados em rede, ou um híbrido dos dois, é claro que o uso de sistemas de detecção de intrusão é uma ferramenta importante e indispensável no arsenal do gerente de segurança.

O *Snort* mostrou-se ser uma ferramenta bastante poderosa para prover segurança em uma rede, constituindo uma ótima alternativa para implementação de um sensor SDI. Como toda ferramenta ela não é infalível, da mesma forma que não existe rede 100% segura. Deve-se manter atento com os assuntos relacionados à segurança da informação.

REFERÊNCIAS BIBLIOGRÁFICAS

CERT.Br. **Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil.** <http://www.cert.br>. Acesso em: 22 mar 2010.

Harper, P., **Snort, Apache, SSL, PHP, MySQL, Acid, Install on Fedora Core 2.** Internet Security Guru, Outubro 2004. Disponível na Internet via url http://www.internetsecurityguru.com/documents/Snort_SSL_FC2.pdf. Acessado em mar 2010.

Wikipédia. Disponível em < <http://www.wikipedia.org> >. Acessado em 24 de Março de 2010.

TANENBAUM, A. S. **Redes de Computadores.** 4ª ed. Rio de Janeiro: Campus, 2003.

CARISSIMI, A. da S.; ROCHOL, J.; GRANVILLE, L.Z. **Redes de Computadores.** Porto Alegre: Bookman, 2009

Instalação do Snort e configuração <http://www.vivaolinux.com.br>.

Intrusion Detection Systems, em <http://www.windowsecurity.com/articles/Intrusion.html>