Implementando Firewall com IDS

Série Firewalls

Manuela Lima manuela@cbpf.br
Marita Maestrelli marita@cbpf.br

Dezembro 2006

CAT – CBPF - MCT

PREFÁCIO

Prosseguindo com a Serie Firewalls de Notas Técnicas do CBPF**[7] ,** descrevemos agora o projeto sobre a integração de um firewall(filtros colocados estrategicamente na entrada de uma rede) com um IDS(sistema de detecção de intrusos) desenvolvido com a bolsista de Iniciação Tecnológica Manuela Lima da Escola Politécnica da UFRJ.

O desenvolvimento tecnológico continua a pleno vapor, enquanto a curiosidade e especulação estão no mesmo ritmo, e surgem cada vez mais internautas mau intencionados explorando nossas redes de produção. Portanto medidas de segurança e prevenção de incidentes fazem parte do dia a dia de quem trabalha com informática.

Índice

1. INTRODUÇÃO

Não há como discutir, os números de invasores e incidentes de segurança vem aumentando ano após ano e ao mesmo passo de nossa tecnologia. Infelizmente, as redes de trabalho não possuem um esconderijo ou um lugar em que elas possam existir sem terem nenhum problema com o mundo externo.

Podemos ser achados através de uma variedade de opções como DNS, (*Name Server Lookup*), *Newsgroups*, web sites, emails e por ai vai. Não se sabe ao certo se a motivação é financeira, desafio intelectual, espionagem, política ou simplesmente para arrumar confusões, mas podemos ser expostos a várias ameaças.

Para tentar minimizar esses danos esse projeto tem como objetivo unir Sistemas de Detecção de Intrusos(**1.1**) a um Firewall(**1.2**) e possibilitar um gerenciamento mais fácil e eficiente de uma rede.

No item **2** mostraremos a estrutura do projeto proposto.

1.1. Sistema de Detecção de Intrusos – IDS

Os Sistemas de Detecção de Intrusos (IDS) são *softwares* utilizados para uma melhor monitoração da rede a procura de sinais padrões de comportamento considerados maliciosos. Existem muitas categorias de IDS dependendo do tipo e da localização dos seus sensores, e de sua metodologia de uso com os geradores de alertas.

Ele é composto por vários componentes: sensores, que geram eventos de segurança, um painel de controle para monitorar eventos e alertas, para controlar os sensores e também um programa central para gravar os logs pelos sensores em um banco de dados e tudo isso ajuda e facilita usar um sistema de regras para gerar alertas advindos de anomalias no padrão da rede.

Os IDSs ajudam na administração das redes em geral, sendo ferramentas que possibilitam acompanhar o tráfego da rede, saber quais pacotes consomem mais ou menos banda podemos, inclusive, saber endereçamentos desses pacotes. Através do IDS é possível acompanhar como a rede está sendo utilizada e, se necessário, bloquear certos tipos de pacotes, ou ao invés de pacotes bloquear por endereços de rede ou de máquinas específicas.

1.2 Firewall

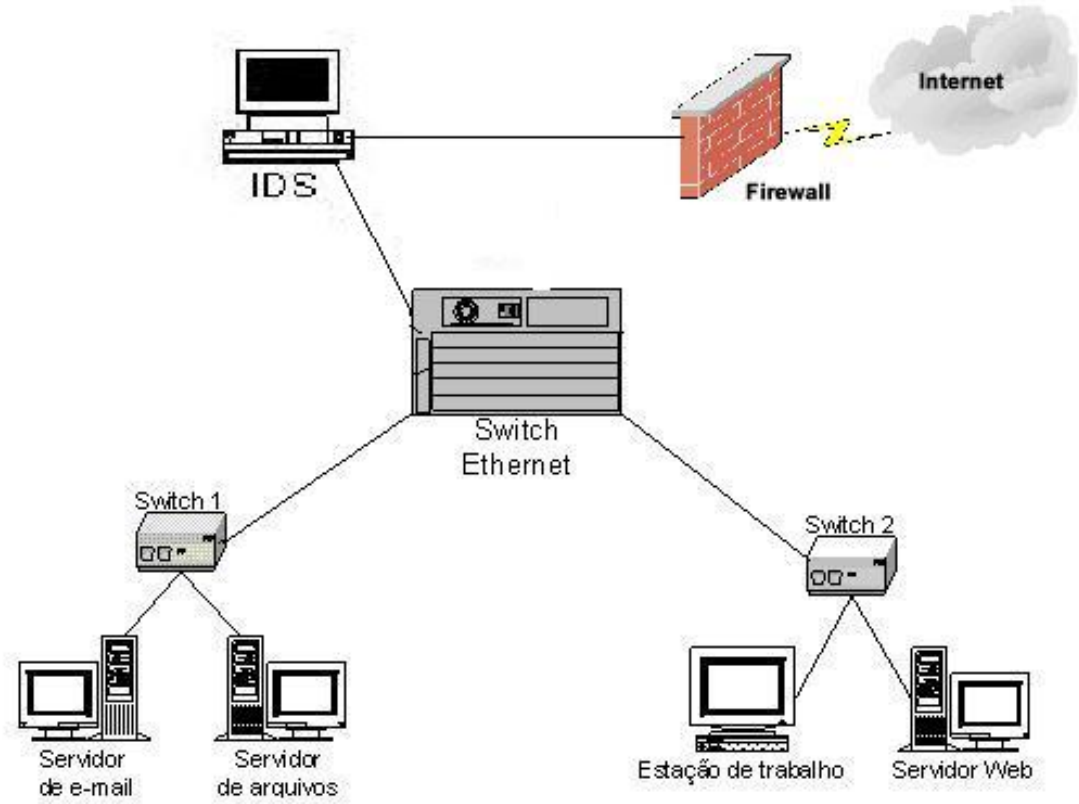
Firewall consiste em combinar *hardware* e *software* para limitar a exposição de uma máquina ou uma rede de um ataque feito por um '*hacker*', ou melhor, é o nome dado ao dispositivo de rede que tem por função regular o tráfego de rede entre redes distintas e impedir a transmissão de dados nocivos ou não autorizados de uma rede a outra.

Existe na forma de *software* e *hardware*, ou na combinação de ambos. A instalação depende do tamanho da rede, da complexidade das regras que autorizam o fluxo de entrada e saída de informações e do grau de segurança desejado.

É utilizado para evitar que o tráfego não autorizado possa fluir de um domínio de rede para o outro, apesar de ser diretamente ligado a proteção de redes, o firewall não possui capacidade de analisar toda a extensão do protocolo, ficando geralmente restrito ao nível 4 da camada OSI (transporte).

2. ESTRUTURA

Esboço do Projeto proposto:



2.1. Sistema Operacional – Slackware 10.2

O Slackware Linux é um Sistema Operacional livre, ou seja, está disponível na internet e todos têm acesso ao código-fonte, podendo então melhorá-lo ou adaptá-lo às próprias nescessidades, atualmente está na sua versão 10.2.

Slackware Linux, ou somente ‘Slack’, é o nome de uma das mais antigas e conhecidas distribuições do Linux. Tem como objetivo manter-se fiel aos padrões UNIX, deixando de fora ferramentas de configuração que escondam do usuário o real funcionamento do sistema. Além disso o Slack é composto somente de aplicativos estáveis, ou seja, não possui versões beta ou pré-releases.

Simplicidade e estabilidade são duas de suas características marcantes que muitos de seus usuários mais experientes veneram. A tradução de Slackware é "Sistema preguiçoso" no ambito de que não gostar de ferramentas de configuração, assim sendo, as configurações do sistema são feitas a partir da edição de documentos de texto, por isso é a preferida entre os usuários mais experientes.

Possui seu próprio gerenciamento de pacotes, o pkgtool (installpkg, upgradepkg, removekpg, explodepkg, makepkg), sem gerenciamento de dependências (existem programas que adicionam esse gerenciamento, como o slapt-get e swaret). O formato dos pacotes .tgz é bastante simples, similar a um .tar.gz contendo apenas os arquivos a serem instalados em suas respectivas pastas em relação à raiz do sistema, além de um script com comandos complementares para a instalação.

2.2. IDS

Existem hoje vários tipos de IDS, alguns exemplos são o SNORT (usado neste projeto) que utiliza de vários outros programas como MySQL, PHP entre outros para seu completo funcionamento, o PRELUDE que habilita diferentes tipos de sensores para gerar eventos usando linguagens unificadas e o OSSEC HIDS que faz análise de logs, checagem de integridade de arquivos, detecção de rootkits, notificação por e-mail e resposta automatica a incidentes (bloqueios no firewall ou no hosts-deny, etc).

Dentre esses vários tipos de IDS eles se classificam em IDS baseados em redes e IDS baseados em host e dentro desses por assinaturas e anomalias que seguem abaixo devidamente explicadas.

2.2.1. Tipos de IDS

- I**DS baseados em redes**:

São normalmente um conjunto de IDS colocados nos pontos de entrada e saída de cada rede. Cada IDS monitora o tráfego e faz uma análise da rede logando os tipos de ataques em uma pasta específica para melhor acompanhamento do administrator. Se aconselha colocar esses IDS em *bridges* para dificultar a sua localização por *hackers*.

Um pequeno número de IDS instalados em uma organização causa um impacto muito pequeno na rede, pois como seu funcionamento é classificado como passivo, ou seja, ele passa o tempo escutando a rede. O ideal é que se faça um mapeamento da rede para melhor alocar os IDS. Poucos IDS alocados e localizados em pontos chaves da rede podem monitorar uma boa parte da mesma e gerar logs significativos.

Alguns dos problemas de se ter um IDS baseado em rede é que por sofrer uma grande carga de tráfego de dados e ter que processar muitos pacotes, pode ser que em algum momento ele venha a perder alguns ou falhar numa aplicação da regra na hora em que o tráfego é muito intenso. Outro grande problema é que os IDS baseados em rede não podem analisar pacotes criptografados, os quais vêm aumentando a cada dia e, também não podem informar se o ataque foi bem sucedido ou não, apenas geram logs para o administrador. No futuro, devem ser estudados métodos para que não aconteçam de novo. Até mesmo uma rede com um *switch* pode ser um problema, pois eles ramificam muito a rede.

- **IDS baseados em host**:

Ao contrário dos IDS baseados em redes, o baseado em hosts consegue analisar um grande número de informações. Como ele foi instalado para monitorar apenas uma máquina ele tem uma precisão muito maior para processar os pacotes e ataques voltados para ele.

Por ser um IDS local ele pode operar com pacotes criptografados e tem uma maior capacidade de gerenciar e detectar ataques já que está destinado a avaliar todo o tipo de informações que passam somente naquela máquina, e não possuem o problema com o *switch*, pois como são locais, a ramificação da rede não interfere na sua operação.

Porém, IDS baseado em host torna o gerenciamento pelo administrador muito mais difícil, pois não existe nenhum método de centralizar os logs, fazendo com que seja preciso ir em cada um dos IDS para fazer a análise de cada log. Por não fazerem detecção de *scans* de rede, são facilmente desativados (como por exemplo por um ataque de negação), consomem muito da máquina utilizada.

2.2.2 Detecções baseadas em assinaturas ou anomalias

- **Por assinaturas**

Analisam eventos que se assemelham a padrões pré-definidos de ataques e/ou atividades maliciosas. Cada ataque e/ou atividade maliciosa corresponde a uma assinatura específica. Têem como pontos positivos sua eficiência em avaliar os pacotes não gerando assim um grande número de alarmes falsos, e podem detectar qual tipo de técnica ou ferramenta foi utilizada exatamente para o ataque.

Porém, possui uma necessidade extrema de ser atualizado constantemente para poder detectar quais assinaturas estão sendo usadas e para qual tipo de ataque, o que causa também uma restrição, o IDS não consegue 'avaliar' as variações dessas assinaturas por mais atualizadas que estejam.

- **Por anomalias**

Parte do princípio que tudo aquilo que não é atividade normal é ataque ou atividade maliciosa. Esse tipo de monitoração ainda não é muito confiável porque lança um grande número de alarmes falsos, pois cada usuário possui um comportamento diferente com atividades/rotinas diferentes que podem não ser agressivas a rede ou a certo host.

Como não possui um padrão ou pré-definições de ataques é capaz de detectar um comportamento fora do cotidiano e produzir informações que poderão ser usadas posteriormente nos detectores por assinaturas. Contudo, requer um enorme tempo para verificação dos dados registrados.

2.3. Firewall

Os sistemas de firewall podem ser classificados da seguinte forma:

1.Filtro de Pacotes

Estes sistemas analisam individualmente os pacotes à medida que estes são transmitidos da camada de enlace (camada 2 do modelo ISO/OSI) para a camada de rede (camada 3 do modelo ISO/OSI).

As regras podem ser formadas estabelecendo os endereços de rede (origem e destino) e as portas TCP/IP envolvidas na conexão. As principais desvantagens deste tipo de tecnologia é a falta de controle de estado do pacote, o que permite que agentes maliciosos possam produzir pacotes simulados (IP Spoofing) para serem injetados na sessão.

2. Proxy Firewall

Os conhecidos "bastion hosts", os firewalls de proxy trabalham recebendo o fluxo de conexão e originando um novo pedido sob a responsabilidade do firewall (non-transparent proxy). A resposta para o pedido é analisada antes de ser entregue para o solicitante original.

Stateful Firewall

Os firewalls de estado, prometia ter capacidade para identificar o protocolo dos pacotes transitados e "prever" as respostas legítimas. Na verdade, ele inspecionava o tráfego para evitar pacotes ilegítimos, guardando o estado de todas as últimas transações efetuadas (este firewall pode impedir o funcionamento do msn messenger).

Firewall de Aplicação

Com a explosão do comércio eletrônico percebeu-se que mesmo a última tecnologia em filtragem de pacotes TCP/IP poderia não ser tão efetiva quanto se esperava. Com todos os investimentos dispendidos em tecnologia de stateful firewalls, as estatísticas demonstravam que os ataques continuavam a prosperar de forma avassaladora. Percebeu-se que havia a necessidade de desenvolver uma tecnologia que pudesse analisar as particularidades de cada protocolo e tomar decisões que pudessem evitar ataques maliciosos.

A tecnologia vem sendo explorada do começo dos anos 90. A idéia é analisar o protocolo específico da aplicação e tomar decisões dentro das particularidades da aplicação, criando uma complexidade muito maior do que configurar regras de fluxo de tráfego TCP/IP.

- Comandos e Opções de Firewall

MASQUERADE: esta opção em um comando iptables permite a tradução de endereços de rede quando um pacote de dados passa por um servidor firewall.

REDIRECT: esta opção, quando associada aos comandos iptables ou ipchains em um servidor firewall, permite a configuração de um sistema de proxy transparente.

3. INSTALAÇÃO:

Instalando e configurando um IDS

5. 1º Programas necessários:

mysql
zlib
libpcap
snort
apache
php
jpgraph
adodb
acid

6. 2º Começando a instalar:

No nosso caso iremos criar uma pasta chamada 'ids' dentro de /usr/local/ e lá, iremos descompactar todos os programas necessários.

MYSQL
Entrar na pasta onde irá descompactar o mysql e executar o seguinte comando:

```
#cd /usr/local/ids
#tar -xzvf mysql-5.0.15.tar.gz
```

Agora iremos mudar a nomenclatura da pasta mysql-5.0.15 para facilitar o uso:

```
#mv -v mysql-5.0.15 mysql
```

Agora, depois de descompactarmos, iremos compilar o banco de dados. Entre na pasta mysql e execute os comandos:

```
#cd mysql
#./configure
#make
#make install (se você já estiver acostumado a usar o checkinstall pode substituir o make install por ele)
```

Em seguida entre na pasta scripts e execute o arquivo mysql_install_db que é usado para inicializar o banco de dados e suas tabelas de permissão. Depois mudaremos alguns atributos de proprietários:

```
#cd scripts
#./mysql_install_db
#chown -R root /usr/local/ids/mysql
```

```
#chown -R mysql /usr/local/mysql/var
#chgrp -R mysql /usr/local/mysql
```

Feito isso, copiaremos o arquivo "my-medium.cnf" para a pasta /etc/ e vamos adicionar as linhas "/usr/local/mysql/lib/mysql" e "/usr/local/lib" ao final do arquivo /etc/ld.so.conf. Depois disso execute o comando:

```
#cp -v support-files/my-medium.cnf /etc/my.cnf
#ldconfig -v
```

Agora entre no arquivo my.cnf e verifique se o seguinte trecho corresponde com o que está no arquivo, caso contrário altere para que fique assim:

```
# The following options will be passed to all MySQL clients
[client]
#password       = your_password
port            = 3306
socket          = /var/run/mysql/mysql.sock

# Here follows entries for some specific programs

# The MySQL server
[mysqld]
port            = 3306
socket          = /var/run/mysql/mysql.sock
```

A seguir testaremos o BD:

```
#/usr/local/mysql/bin/mysqld_safe -user=mysql (você deve pressionar 'enter' pararetornar ao prompt)
```

Agora vamos configurar uma senha para o root do mysql:

```
#mysql
mysql> SET PASSWORD FOR root@localhost=PASSWORD('passwd');
mysql>quit;
```

Agora para inicializarmos o BD é necessário que se digite:

```
#mysql -p (e a senha que ele pede é a de root que acabamos de configurar)
```

Para colocar o mysql para iniciar automaticamente basta copiar o arquivo mysql.server para a pasta /etc/rc.d mudar sua permissão.

```
#cp -v support-files/mysql.server /etc/rc.d/mysql
#chmod 755 /etc/rc.d/mysql
```

Volte par pasta de origem dos arquivos que é, no nosso caso, "/usr/local/ids".

ZLIB

#tar -zxvf zlib-1.2.3.tar.gz
#cd zlib-1.2.3
#./configure
#make
#make install (ou checkinstall)

Volte par pasta de origem dos arquivos que é, no nosso caso, "/usr/local/ids".

LIBPCAP

#tar -xzvf libpcap-0.9.4.tar.gz
#cd libpcap-0.9.4
#./configure
#make
#make install (ou checkinstall)

Volte par pasta de origem dos arquivos que é, no nosso caso, "/usr/local/ids".

SNORT

#tar -xzvf snort-2.4.3.tar.gz
#cd snort-2.4.3
#./configure --with-mysql=/usr/local/mysql
#make
#make install (ou checkinstall)

Agora para testar o funcionamento do mesmo digite:

#snort -vde (para finalizar o teste é preciso apertar ctrl+C)

Após verificarmos que o snort está funcionando, teremos que configurá-lo, para isso iremos mexer no arquivo snort.conf.

#vi etc/snort.conf

Configure aqui o endereço da sua rede interna (var HOME_NET 10.2.2.0/24), o path das regras (var RULE_PATH ../RULES) e por último descomente a linha "output database: log, mysql, user=root password=test dbname=db host=localhost" para que o snort use o mysql como banco de dados e altere o password para a sua senha e o dbname para 'snort'.

Ajustaremos agora o mysql:

#mysql -p (a senha pedida aqui é a do root do mysql)
mysql> CREATE DATABASE snort;

```
mysql> grant INSERT, SELECT on snort.* to root@localhost;
mysql> quit
```

Criaremos o banco de dados do snort no mysql. Para isso volaremos ao diretório aonde o snort foi instalado (/usr/local/ids/snort-2.4.3).

#/usr/local/mysql/bin/mysql -p <./schemas/create_mysql snort (a senha pedida aqui é a do root do mysql)

Vamos checar agora se as tabelas foram corretamente criadas.

```
#mysql -p
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| snort              |
| test               |
+--------------------+
4 rows in set (0.10 sec)

mysql> use snort;
Database changed

mysql> show tables;
+------------------+
| Tables_in_snort  |
+------------------+
| data             |
| detail           |
| encoding         |
| event            |
| icmphdr          |
| iphdr            |
| opt              |
| reference        |
| reference_system |
| schema           |
| sensor           |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+------------------+
16 rows in set (0.01 sec)
```

mysql> quit

Que tal fazermos um teste antes de instalarmos uma interface gráfica para verificarse o snort está funcionando com o banco de dados.

Pode ser que seu mysql dê um erro dizendo que não está achando o arquivo mysql.sock, /tmp/mysql.sock. Para resolvê-lo, basta criar um link simbólico dentro de /tmp apontando para a real localização do arquivo, que no meu caso é “/var/run/mysql/mysql.sock”, desse modo.

#cd /tmp
#ln –s /var/run/mysql/mysql.sock

Dê um ‘ls –l’ e verifique se foi criado o link.

#snort -vde -c /usr/local/ids/snort-2.4.3/etc/snort.conf

Agora voltaremos a pasta /usr/local/ids para continuar a instalar os outros programas.

APACHE

#tar -xzvf httpd-2.2.0.tar.gz
#cd httpd-2.2.0
#./confiugre --prefix=/usr/local/apache --enable-so
#make
#make install (ou checkinstall)

Para testar se o apache está funcionando direito digite o seguinte comando:

#/usr/local/apache/bin/apachectl start

Depois abra o seu navegar e digite o endereço http://localhost.

PHP

#tar -xzvf php-5.1.2.tar.gz
#cd php-5.1.2
#./configure --prefix=/usr/local/php --with-apxs2=/usr/local/apache/bin/apxs --with-config-file-path=/usr/local/php --with-zlib-dir=/usr/local/ids/zlib-1.2.3/ --with-mysql=/usr/local/mysql/ --with-gd
#make
#make install (ou checkinstall)

Precisaremos mudar o arquivo de conifugração do Apache para ele invoque o parser do PHP. No aruiqvo httpd.conf procure por 'AddType' e adicione a seguinte linha abaixo de AddType application/x-gzip .gz .tgz:

```
#vi /usr/local/ids/httpd-2.2.0/docs/conf/httpd.conf
AddType application/x-httpd-php .php
```

Após a instalação vamos testar.
Abra um editor de textos e escreva as linhas abaixo e salve com o nome de test.php no diretório /usr/local/apache/htdocs.

```
#vi /usr/local/apache/htdocs/test.php
<?php
phpinfo();
?>

#cp /usr/local/ids/php-5.1.2/php.ini-dist /usr/local/php/php.ini
#/usr/local/apache/bin/apachectl restart
```

Agora no navegador digite 'http://localhost/test.php'. Se estiver tudo OK aparecerá uma página com informações sobre o ambiente Apache.

JPGRAPH

É a biblioteca gráfica orientada a objeto para a linguagem PHP. Ela nos ajudará a ver os resultados apurados pelo SNORT e armazenados pelo MYSQL.
Começaramos então a descompactar igual aos outros.
Agora iremos para a pasta /usr/local/apache/htdocs para continuar a instalar os outros programas.

```
#tar -xzvf jpgraph-1.20.3
```

ADODB

É uma biblioteca de classes utilizada por banco de dados para dar mais flexibilidade.
Ainda dentro da pasta /usr/local/apache/htdocs dê o seguintes comandos:

```
#tar -xzvf adodb390.tgz
```

ACID - Analysis Console Intrusion Database

Agora iremos descompactá-lo e configurar o seu arquivo acid_conf.php

```
#tar -xzvf /acid-0.9.6b23.tar.gz
#vi acid/acid_conf.php
```

Faça com que as seguintes linhas fiquem desses jeito:

```
$DBlib_path = "";
```

```
$DBlib_path = "/usr/local/apache/htdocs/adodb";

$DBtype = "";
$DBtype = "mysql";

 *  This information can be gleaned from the Snort database
 *  output plugin configuration.
 */
$alert_dbname   = "snort_log";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "root";
$alert_password = "password";

/* Archive DB connection parameters */
$archive_dbname   = "snort_archive";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "root";
$archive_password = "password";

 *  This information can be gleaned from the Snort database
 *  output plugin configuration.
 */
$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "root";
$alert_password = "password";

/* Archive DB connection parameters */
$archive_dbname   = "snort";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "root";
$archive_password = "password";


$ChartLib_path = "";
$ChartLib_path = "/usr/local/apache/htdocs/jpgraph-1.20.3/src/";
```

Feito isso abra o navegador e digite http://localhost/acid/acid_main.php.
Como essa será a primeira vez que rodamos o acid, ele mostrará um aviso dizendo que não encontra a estruturas de tabelas do ACID. Para dar seqüência na criação dessa estrutura clique em "Page Setup" e depois aparecerá outro botão "Create ACID AG" que você também deverá clicar.

Depois volte a "main page" e pronto seu IDS está completamente instalado.

Exemplos de utilização do Snort podem ser vistos no Anexo 1.

4. RESULTADOS:

Após certo tempo em prática conseguimos identificar os tipos de tráfegos de pacotes, testar o funcionamento pleno do SNORT como IDS e melhorar a funcionalidade do FIREWALL para uma melhor proteção para as portas do servidor da rede.

É possível também utilizar os logs salvos pelo SNORT para analisar e gerar repostas a determinados ataques.

5. REFERÊNCIAS:

[1] Snort 2 – Sistema de Detecção de Intruso – Open Source
Brian Caswell, Jay Beale, James C. Foster, Jeffrey Posluns
Editora Alta Books

[2] Snort User Manual
Martin Roesch, Chris Green
- http://www.snort.org/docs/snort_htmanuals/htmanual_2.4/rc1/

[3] Snort, MySQL, Apache e ACID
Paulo Augusto Moda Lari, Dino Macedo Amaral
Editora Brasport

[4] Trabalhando com PHP e MySQL: uma introdução
Apostilando.com – O portal do conhecimento
- http://apostilando.com

[5] Internet Security System – How to Guide – Implementing a network based Intrusion Detection Systems
Brian Laing

[6] Wikipedia
3. http://www.wikipedia.org

[7] CBPF-NT-007/03 - IPFILTER, Série Firewalls – A. A. de Albuquerque, F. Spencer, J. Gracia Neto, M. Maestrelli – outubro/2003

6. GLOSSÁRIO

Hackers – indivíduos que elaboram e modificam software e hardware de computadores, desenvolvendo funcionalidades novas, ou adaptando as antigas. São programadores maliciosos e ciberpiratas que agem com o intuito de violar ilegal ou imoralmente sistemas cibernéticos;

Bridge – máquina utlizada para fazer a ligação entre duas ou mais redes;

IDS – Sistema de Detecção de Intrusos, programa utilizado para proteger determinada rede;

Firewall – dispositivo de rede que tem por função regular o tráfego de rede entre redes distintas e impedir a transmissão de dados nocivos ou não autorizados de uma rede a outraou de um pc para outro

Log - termo utilizado para descrever o registro de eventos relevantes num sistema computacional;

Criptografia - técnica utilizada para que a informação possa ser transformada da sua forma original para outra ilegível, a menos que seja conhecida uma "chave secreta", o que a torna difícil de ser lida por alguém não autorizado. Assim sendo, só o receptor da mensagem pode ler a informação com facilidade;

Switch – ou **comutador**, é um dispositivo utilizado em redes de computadores para reencaminhar pacotes entre os diversos nós da rede;

Scan – varredura feita por anti-vírus ou programas utilizados por hackers para investigar uma determinada rede ou computador;

DOS – ataque também conhecido como Denial of Service, consiste em tentativas de impedir usuários legítimos de utilizarem um determinado serviço de um computador. Para isso, são usadas técnicas que podem: sobrecarregar uma rede a tal ponto em que os verdadeiros usuários dela não consigam usá-la; derrubar uma conexão entre dois ou mais computadores; fazer tantas

requisições a um site até que este não consiga mais ser acessado; negar acesso a um sistema ou a determinados usuários;

DDOS – Distributed Denial of Service – é um ataque *DoS* ampliado, ou seja, pela instalação de vários agentes remotos em muitos computadores localizadas em várias partes da internet , o invasor consegue coordenar esses agentes em massa para amplificar o volume da flood (inundação) , podendo utilizar até milhares de computadores para atacar uma determinada máquina ou rede;

Proxy - é um software que armazena dados em forma de cache em redes de computadores. São máquinas com ligações tipicamente superiores às dos clientes e com poder de armazenamento elevado;

Bastion Hosts - é um sistema identificado por administradores de firewall como um ponto crítico na segurança de uma rede. Geralmente, bastion hosts necessitam de um cuidado extra, com auditorias regulares e podem ter software modificado;

Enlace (de rede) – a camada de enlace não é realmente parte do modelo TCP/IP, mas é o método usado para passar pacotes da camada de rede de um dispositivo para a camada de internet de outro;

Filtros de Pacotes – é um conjunto de regras que analisam e filtram pacotes enviados por redes distintas de comunicação.

ANEXO 1

Para usar no modo "Sniffer" basta utilizar o comando que mostra informações do IP como os headers TCP/UDP/ICMP:

```
./snort -v
```

Aparecerá algo parecido com o exemplo abaixo.

```
# ./snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Verifying Preprocessor Configurations!
***
*** interface device lookup found: eth0
***

Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.6.0 (Build 59)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2006 Sourcefire Inc., et al.

Not Using PCAP_FRAMES
12/12-15:30:35.714229 217.165.119.248:6548 -> 192.168.0.25:1327
UDP TTL:41 TOS:0x0 ID:0 IpLen:20 DgmLen:70 DF
Len: 42
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-15:30:35.714385 192.168.0.25 -> 217.165.119.248
ICMP TTL:64 TOS:0xC0 ID:42375 IpLen:20 DgmLen:98
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
217.165.119.248:6542 -> 192.168.0.25:6546
UDP TTL:41 TOS:0x0 ID:0 IpLen:20 DgmLen:70 DF
Len: 42
** END OF DUMP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-15:30:46.583160 ARP who-has 192.168.0.105 (FF:FF:FF:FF:FF:FF) tell
192.168.0.215

12/12-15:30:46.655500 86.122.174.225:55555 -> 192.168.0.25:5432
UDP TTL:106 TOS:0x0 ID:55726 IpLen:20 DgmLen:70
Len: 42
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-15:30:46.655707 192.168.0.25 -> 86.122.174.225
ICMP TTL:64 TOS:0xC0 ID:6303 IpLen:20 DgmLen:98
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
```

```
86.122.174.225:55555 -> 192.168.0.25:6548
UDP TTL:106 TOS:0x0 ID:55726 IpLen:20 DgmLen:70
Len: 42
** END OF DUMP
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

*** Caught Int-Signal


===============================================================================

Snort received 17 packets
    Analyzed: 7(41.176%)
    Dropped: 0(0.000%)
    Outstanding: 10(58.824%)
===============================================================================
Breakdown by protocol:
    TCP: 0          (0.000%)
    UDP: 3          (42.857%)
   ICMP: 3          (42.857%)
    ARP: 1          (14.286%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
ETHLOOP: 0          (0.000%)
    IPX: 0          (0.000%)
   FRAG: 0          (0.000%)
  OTHER: 0          (0.000%)
DISCARD: 0          (0.000%)
===============================================================================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================================================================
Snort exiting
```

Para poder avaliar também os dados trafegando utiliza-se a opção -d e para ver informações sobre os headers da layer de link basta adicionar -e:

```
./snort -vde
```

Agora utilizando essa opção as informações serão similares a estas:

```
# ./snort -vde
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Verifying Preprocessor Configurations!
***
*** interface device lookup found: eth0
***

Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--
```

```
   ,,_     -*> Snort! <*-
  o"  )~   Version 2.6.0 (Build 59)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2006 Sourcefire Inc., et al.

Not Using PCAP_FRAMES
12/12-15:33:11.890745 ARP who-has 192.168.0.25 tell 192.168.0.1

12/12-15:33:11.890799 ARP who-has 192.168.0.123 tell 192.168.0.1

12/12-15:33:11.890831 ARP who-has 192.168.0.5 tell 192.168.0.1

12/12-15:33:11.890860 ARP who-has 192.168.0.220 tell 192.168.0.1

12/12-15:33:12.013021 0:1:3:2D:C9:2 -> 0:4:96:1A:3F:70 type:0x800 len:0x85
192.168.0.25 -> 89.129.180.183 ICMP TTL:64 TOS:0xC0 ID:25354 IpLen:20 DgmLen:119
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
89.129.180.183:16563 -> 192.168.0.25:4525 UDP TTL:106 TOS:0x0 ID:44748 IpLen:20
DgmLen:91
Len: 63
** END OF DUMP
00 00 00 00 45 00 00 5B AE CC 00 00 6A 11 FD E6  ....E..[....j...
59 81 B4 B7 98 54 FD 51 40 B3 77 79 00 47 5A AF  Y....T.Q@.wy.GZ.
E5 F6 41 53 67 68 94 04 00 00 04 04 80 1C 56 B7  ..ASgh........V.
0F 00 00 00 00 00 0F 04 59 81 B4 B7 40 B3 15 C9  ........Y...@...
2D 67 00 00 01 0F 77 B4 24 CC 14 52 88 18 4E D7  -g....w.$..R..N.
11 DA 8E D0 27 CF EC 65 E5 D6 D3 1D EC 4E B8     ....'..e.....N.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-15:33:12.157069 ARP who-has 192.168.0.102 (FF:FF:FF:FF:FF:FF) tell
192.168.0.188

*** Caught Int-Signal


===============================================================================

Snort received 32 packets
    Analyzed: 15(46.875%)
    Dropped: 0(0.000%)
    Outstanding: 17(53.125%)
===============================================================================
Breakdown by protocol:
    TCP: 0          (0.000%)
    UDP: 1          (6.667%)
   ICMP: 1          (6.667%)
    ARP: 12         (80.000%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
ETHLOOP: 0          (0.000%)
    IPX: 0          (0.000%)
   FRAG: 0          (0.000%)
  OTHER: 1          (6.667%)
DISCARD: 0          (0.000%)
===============================================================================
Action Stats:
```

```
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================================================================
Snort exiting


Para executarmos o Snort em modo "Packet Logger" bastar executar, onde
<<arquivo>> pode ser alterado para o lugar aonde irão ficar guardados seus logs
e o -b significa que os logs estarão na forma binária:

./snort -dev -l -b ./<<arquivo>> -h 192.168.1.0/24

Exemplo:

# ./snort -dev -l ./log/ -h 192.168.1.0/24
Running in packet logging mode
Log directory = ./log/

        --== Initializing Snort ==--
Initializing Output Plugins!
Verifying Preprocessor Configurations!
***
*** interface device lookup found: eth0
***

Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.6.0 (Build 59)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
           (C) Copyright 1998-2006 Sourcefire Inc., et al.

Not Using PCAP_FRAMES
12/12-15:47:46.803444 0:E0:18:78:F2:45 -> FF:FF:FF:FF:FF:FF type:0x800 len:0xA9
192.168.0.246:631 -> 192.168.0.255:631 UDP TTL:64 TOS:0x0 ID:5410 IpLen:20
DgmLen:155 DF
Len: 127
39 30 30 65 20 34 20 69 70 70 3A 2F 2F 31 35 32  900e 4 ipp://152
2E 38 34 2E 32 35 33 2E 32 34 36 2F 70 72 69 6E  .84.253.246/prin
74 65 72 73 2F 48 50 20 22 4C 6F 63 61 6C 22 20  ters/HP "Local"
22 48 50 20 4F 46 46 49 43 45 4A 45 54 20 50 52  "HP OFFICEJET PR
4F 20 31 31 35 30 43 22 20 22 48 50 20 4F 66 66  O 1150C" "HP Off
69 63 65 4A 65 74 20 50 72 6F 20 31 31 35 30 43  iceJet Pro 1150C
20 46 6F 6F 6D 61 74 69 63 2F 68 70 69 6A 73 20   Foomatic/hpijs
28 72 65 63 6F 6D 6D 65 6E 64 65 64 29 22 0A     (recommended)".

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

*** Caught Int-Signal


===============================================================================

Snort received 4 packets
    Analyzed: 1(25.000%)
```

```
    Dropped: 0(0.000%)
    Outstanding: 3(75.000%)
===============================================================================
Breakdown by protocol:
    TCP: 0          (0.000%)
    UDP: 1          (100.000%)
   ICMP: 0          (0.000%)
    ARP: 0          (0.000%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
ETHLOOP: 0          (0.000%)
    IPX: 0          (0.000%)
   FRAG: 0          (0.000%)
  OTHER: 0          (0.000%)
DISCARD: 0          (0.000%)
===============================================================================
Action Stats:
ALERTS: 0
LOGGED: 1
PASSED: 0
===============================================================================
Snort exiting
```

E para usar-lo como um Sistema de Detecção de Intrusos em uma rede basta configurar o arquivo snort.conf e executar o comando:

```
./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf
```

Exemplo:

```
# ./snort -dev -l ./log/ -h 192.168.1.0/24 -c snort.conf

Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file snort.conf

++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
Var 'EXTERNAL_NET' defined, value len = 3 chars, value = any
Var 'DNS_SERVERS' defined, value len = 14 chars, value = 192.168.0.0/24
Var 'SMTP_SERVERS' defined, value len = 14 chars, value = 192.168.0.0/24
Var 'HTTP_SERVERS' defined, value len = 14 chars, value = 192.168.0.0/24
Var 'SQL_SERVERS' defined, value len = 14 chars, value = 192.168.0.0/24
Var 'TELNET_SERVERS' defined, value len = 14 chars, value = 192.168.0.0/24
Var 'SNMP_SERVERS' defined, value len = 14 chars, value = 192.168.0.0/24
Var 'HTTP_PORTS' defined, value len = 2 chars, value = 80
Var 'SHELLCODE_PORTS' defined, value len = 3 chars, value = !80
Var 'ORACLE_PORTS' defined, value len = 4 chars, value = 1521
Var 'AIM_SERVERS' defined, value len = 185 chars

[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.18
8.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9
   .0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]
Var 'RULE_PATH' defined, value len = 8 chars, value = ../rules
```

```
,-----------[Flow Config]----------------------
| Stats Interval:  0
| Hash Method:     2
| Memcap:          10485760
| Rows  :          4099
| Overhead Bytes:  16400(%0.16)
`-------------------------------------------
Frag3 global config:
    Max frags: 65536
    Fragment memory cap: 4194304 bytes
Frag3 engine config:
    Target-based policy: FIRST
    Fragment timeout: 60 seconds
    Fragment min_ttl:   1
    Fragment ttl_limit: 5
    Fragment Problems: 1
  Bound Addresses: 0.0.0.0/0.0.0.0
Stream4 config:
    Stateful inspection: ACTIVE
    Session statistics: INACTIVE
    Session timeout: 30 seconds
    Session memory cap: 8388608 bytes
    Session count max: 8192 sessions
    Session cleanup count: 5
    State alerts: INACTIVE
    Evasion alerts: INACTIVE
    Scan alerts: INACTIVE
    Log Flushed Streams: INACTIVE
    MinTTL: 1
    TTL Limit: 5
    Async Link: 0
    State Protection: 0
    Self preservation threshold: 50
    Self preservation period: 90
    Suspend threshold: 200
    Suspend period: 30
    Enforce TCP State: INACTIVE
    Midstream Drop Alerts: INACTIVE
    Server Data Inspection Limit: -1
WARNING snort.conf(372) => flush_behavior set in config file, using old static
flushpoints (0)
Stream4_reassemble config:
    Server reassembly: INACTIVE
    Client reassembly: ACTIVE
    Reassembler alerts: ACTIVE
    Zero out flushed packets: INACTIVE
    Flush stream on alert: INACTIVE
    flush_data_diff_size: 500
    Reassembler Packet Preferance : Favor Old
    Packet Sequence Overlap Limit: -1
    Flush behavior: Small (<255 bytes)
    Ports: 21 23 25 42 53 80 110 111 135 136 137 139 143 445 513 1433 1521 3306
    Emergency Ports: 21 23 25 42 53 80 110 111 135 136 137 139 143 445 513 1433
1521 3306
HttpInspect Config:
    GLOBAL CONFIG
      Max Pipeline Requests:    0
      Inspection Type:          STATELESS
      Detect Proxy Usage:       NO
```

```
      IIS Unicode Map Filename: ./unicode.map
      IIS Unicode Map Codepage: 1252
    DEFAULT SERVER CONFIG:
      Ports: 80 8080 8180
      Flow Depth: 300
 Max Chunk Length: 500000
      Inspect Pipeline Requests: YES
      URI Discovery Strict Mode: NO
      Allow Proxy Usage: NO
      Disable Alerting: NO
      Oversize Dir Length: 500
      Only inspect URI: NO
      Ascii: YES alert: NO
      Double Decoding: YES alert: YES
      %U Encoding: YES alert: YES
      Bare Byte: YES alert: YES
      Base36: OFF
      UTF 8: OFF
      IIS Unicode: YES alert: YES
      Multiple Slash: YES alert: NO
      IIS Backslash: YES alert: NO
      Directory Traversal: YES alert: NO
      Web Root Traversal: YES alert: YES
      Apache WhiteSpace: YES alert: NO
      IIS Delimiter: YES alert: NO
      IIS Unicode Map: GLOBAL IIS UNICODE MAP CONFIG
      Non-RFC Compliant Characters: NONE
rpc_decode arguments:
    Ports to decode RPC on: 111 32771
    alert_fragments: INACTIVE
    alert_large_fragments: ACTIVE
    alert_incomplete: ACTIVE
    alert_multiple_requests: ACTIVE
WARNING: the telnet preprocessor will be deprecated in the next release of
snort.  Please switch to using ftptelnet.
telnet_decode arguments:
    Ports to decode telnet on: 21 23 25 119
Portscan Detection Config:
    Detect Protocols:  TCP UDP ICMP IP
    Detect Scan Type:  portscan portsweep decoy_portscan distributed_portscan
    Sensitivity Level: Low
    Memcap (in bytes): 10000000
    Number of Nodes:   36900

5474 Snort rules read...
5474 Option Chains linked into 230 Chain Headers
0 Dynamic rules
+++++++++++++++++++++++++++++++++++++++++++++++++

Tagged Packet Limit: 256

+----------------------[thresholding-config]----------------------------------
| memory-cap : 1048576 bytes
+----------------------[thresholding-global]----------------------------------
| none
+----------------------[thresholding-local]-----------------------------------
| gen-id=1      sig-id=2924       type=Threshold tracking=dst count=10
seconds=60
```

```
| gen-id=1      sig-id=3527       type=Limit     tracking=dst count=5
seconds=60
| gen-id=1      sig-id=3543       type=Threshold tracking=src count=5
seconds=2
| gen-id=1      sig-id=5323       type=Limit     tracking=src count=1
seconds=60
| gen-id=1      sig-id=3542       type=Threshold tracking=src count=5
seconds=2
| gen-id=1      sig-id=3152       type=Threshold tracking=src count=5
seconds=2
| gen-id=1      sig-id=4984       type=Threshold tracking=src count=5
seconds=2
| gen-id=1      sig-id=2523       type=Both      tracking=dst count=10
seconds=10
| gen-id=1      sig-id=5321       type=Limit     tracking=src count=1
seconds=60
| gen-id=1      sig-id=2923       type=Threshold tracking=dst count=10
seconds=60
| gen-id=1      sig-id=5322       type=Limit     tracking=src count=1
seconds=60
| gen-id=1      sig-id=2275       type=Threshold tracking=dst count=5
seconds=60
| gen-id=1      sig-id=3273       type=Threshold tracking=src count=5
seconds=2
+----------------------[suppression]-----------------------------------------
| none
--------------------------------------------------------------------------------
Rule application order: ->activation->dynamic->pass->drop->alert->log
Log directory = /var/log/snort/
Verifying Preprocessor Configurations!
Warning: flowbits key 'dce.isystemactivator.bind.call.attempt' is set but not
ever checked.
Warning: flowbits key 'dce.bind.veritas' is set but not ever checked.
Warning: flowbits key 'ms_sql_seen_dns' is checked but not ever set.
Warning: flowbits key 'realplayer.playlist' is checked but not ever set.
***
*** interface device lookup found: eth0
***

Initializing Network Interface eth0
Var 'eth0_ADDRESS' defined, value len = 26 chars, value =
152.84.253.0/255.255.255.0
Decoding Ethernet on interface eth0
database: compiled support for ( mysql )
database: configured to use mysql
database: database name = snort
database:          user = snort
database:          host = localhost
database: password is set
Node unique name is: 192.168.0.25
database:   sensor name = 192.168.0.25
database: schema version = 107
database: using the "log" facility

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.6.0 (Build 59)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html
```

```
           (C) Copyright 1998-2006 Sourcefire Inc., et al.

Not Using PCAP_FRAMES
*** Caught Int-Signal
Frag3 statistics:
        Total Fragments: 0
      Frags Reassembled: 0
               Discards: 0
          Memory Faults: 0
               Timeouts: 0
               Overlaps: 0
              Anomalies: 0
                 Alerts: 0
     FragTrackers Added: 0
    FragTrackers Dumped: 0
FragTrackers Auto Freed: 0
    Frag Nodes Inserted: 0
     Frag Nodes Deleted: 0
===============================================================================
Final Flow Statistics
,----[ FLOWCACHE STATS ]----------
Memcap: 10485760 Overhead Bytes 16400 used(%0.156403)/blocks (16400/1)
Overhead blocks: 1 Could Hold: (0)
IPV4 count: 0 frees: 0
low_time: 0, high_time: 0, diff: 0h:00:00s
    finds: 0 reversed: 0(%0.000000)
    find_success: 0 find_fail: 0
percent_success: (%0.000000) new_flows: 0


===============================================================================
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-16:16:53.087517 0:4:96:1A:3F:70 -> 0:1:3:2D:C9:2 type:0x800 len:0x6B
88.149.178.51:58455 -> 192.168.0.25:6548 UDP TTL:107 TOS:0x0 ID:40443 IpLen:20
DgmLen:93
Len: 65
C5 BA 29 95 31 D3 1D E5 00 00 04 06 48 9B 7E 0B  ..).1.......H.~.
0F 00 00 00 00 00 0F 04 58 95 B2 33 ED FD 91 79  ........X..3...y
9D DA 00 00 01 0F 77 DC 1A 14 14 52 C8 83 A6 0B  ......w....R....
AF 5C 65 36 6D 50 7A 0C 74 2D 0B DA 65 E2 68 00  .\e6mPz.t-..e.h.
01                                               .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-16:16:53.087731 0:1:3:2D:C9:2 -> 0:4:96:1A:3F:70 type:0x800 len:0x87
192.168.0.25 -> 88.149.178.51 ICMP TTL:64 TOS:0xC0 ID:4982 IpLen:20 DgmLen:121
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
88.149.178.51:58455 -> 152.84.253.81:30585 UDP TTL:107 TOS:0x0 ID:40443 IpLen:20
DgmLen:93
Len: 65
** END OF DUMP
00 00 00 00 45 00 00 5D 9D FB 00 00 6B 11 11 26  ....E..]....k..&
58 95 B2 33 98 54 FD 51 E4 57 77 79 00 49 52 8C  X..3.T.Q.Wwy.IR.
C5 BA 29 95 31 D3 1D E5 00 00 04 06 48 9B 7E 0B  ..).1.......H.~.
0F 00 00 00 00 00 0F 04 58 95 B2 33 ED FD 91 79  ........X..3...y
9D DA 00 00 01 0F 77 DC 1A 14 14 52 C8 83 A6 0B  ......w....R....
AF 5C 65 36 6D 50 7A 0C 74 2D 0B DA 65 E2 68 00  .\e6mPz.t-..e.h.
```

```
01                                                  .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-16:16:53.195005 ARP who-has 192.168.0.73 (FF:FF:FF:FF:FF:FF) tell
152.84.253.215

12/12-16:16:53.602802 ARP who-has 192.168.0.4 (FF:FF:FF:FF:FF:FF) tell
152.84.253.2

12/12-16:16:53.612549 ARP who-has 192.168.0.64 tell 152.84.253.1

12/12-16:16:54.195206 ARP who-has 192.168.0.73 (FF:FF:FF:FF:FF:FF) tell
152.84.253.215

12/12-16:16:54.602599 ARP who-has 192.168.0.4 (FF:FF:FF:FF:FF:FF) tell
152.84.253.2
12/12-16:16:54.602888 0:4:96:1A:3F:70 -> 0:1:3:2D:C9:2 type:0x800 len:0x54
81.13.89.118:39373 -> 152.84.253.81:30585 UDP TTL:108 TOS:0x0 ID:45185 IpLen:20
DgmLen:70
Len: 42
FD 0A 58 B9 92 F0 19 BF 00 00 04 00 FF 70 69 64  ..X.........pid
0F 00 00 00 00 00 0F 04 51 0D 59 76 99 CD FF 86  ........Q.Yv....
0C E8 00 00 01 0F 77 DB E4 AC                    ......w...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

12/12-16:16:54.603052 0:1:3:2D:C9:2 -> 0:4:96:1A:3F:70 type:0x800 len:0x70
152.84.253.81 -> 81.13.89.118 ICMP TTL:64 TOS:0xC0 ID:49112 IpLen:20 DgmLen:98
Type:3  Code:3  DESTINATION UNREACHABLE: PORT UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
81.13.89.118:39373 -> 192.168.0.25:6548 UDP TTL:108 TOS:0x0 ID:45185 IpLen:20
DgmLen:70
Len: 42
** END OF DUMP
00 00 00 00 45 00 00 46 B0 81 00 00 6C 11 5D FC  ....E..F....l.].
51 0D 59 76 98 54 FD 51 99 CD 77 79 00 32 72 74  Q.Yv.T.Q..wy.2rt
FD 0A 58 B9 92 F0 19 BF 00 00 04 00 FF 70 69 64  ..X.........pid
0F 00 00 00 00 00 0F 04 51 0D 59 76 99 CD FF 86  ........Q.Yv....
0C E8 00 00 01 0F 77 DB E4 AC                    ......w...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

*** Caught Int-Signal
Frag3 statistics:
        Total Fragments: 0
      Frags Reassembled: 0
               Discards: 0
          Memory Faults: 0
               Timeouts: 0
               Overlaps: 0
              Anomalies: 0
                 Alerts: 0
     FragTrackers Added: 0
    FragTrackers Dumped: 0
FragTrackers Auto Freed: 0
    Frag Nodes Inserted: 0
     Frag Nodes Deleted: 0
===============================================================================
```

```
Final Flow Statistics
,----[ FLOWCACHE STATS ]----------
Memcap: 10485760 Overhead Bytes 16400 used(%0.234928)/blocks (24634/47)
Overhead blocks: 1 Could Hold: (58579)
IPV4 count: 46 frees: 0
low_time: 1165947399, high_time: 1165947415, diff: 0h:00:16s
    finds: 54 reversed: 2(%3.703704)
 find_success: 8 find_fail: 46
percent_success: (%14.814815) new_flows: 46
 Protocol: 1 (%35.185185)
   finds: 19
   reversed: 0(%0.000000)
   find_success: 0
   find_fail: 19
   percent_success: (%0.000000)
   new_flows: 19
 Protocol: 2 (%3.703704)
   finds: 2
   reversed: 0(%0.000000)
   find_success: 0
   find_fail: 2
   percent_success: (%0.000000)
   new_flows: 2
 Protocol: 6 (%11.111111)
   finds: 6
   reversed: 2(%33.333333)
   find_success: 4
   find_fail: 2
   percent_success: (%66.666667)
   new_flows: 2
 Protocol: 17 (%50.000000)
   finds: 27
   reversed: 0(%0.000000)
   find_success: 4
   find_fail: 23
   percent_success: (%14.814815)
   new_flows: 23


===============================================================================

Snort received 189 packets
    Analyzed: 0(0.000%)
    Dropped: 0(0.000%)
    Outstanding: 56(100.000%)
===============================================================================
Breakdown by protocol:
    TCP: 0          (0.000%)
    UDP: 0          (0.000%)
   ICMP: 0          (0.000%)
    ARP: 0          (0.000%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
ETHLOOP: 0          (0.000%)
    IPX: 0          (0.000%)
   FRAG: 0          (0.000%)
  OTHER: 0          (0.000%)
DISCARD: 0          (0.000%)
===============================================================================
```

```
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================================================================
database: Closing connection to database "snort"
Snort exiting
```