



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DENIS DA SILVA SERRANO

**ANÁLISE E IMPLEMENTAÇÃO DE IDS (INTRUSION DETECTION SYSTEM)
EM REDES DE COMPUTADORES**

ASSIS
2012

DENIS DA SILVA SERRANO

**ANÁLISE E IMPLEMENTAÇÃO DE IDS (Intrusion Detection System) EM
REDES DE COMPUTADORES**

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal de
Ensino Superior de Assis, como
requisito de Curso de Graduação,
analisado pela seguinte comissão
examinadora:

Orientador: Prof. Me. Fábio Eder Cardoso

ASSIS
2012

DENIS DA SILVA SERRANO

**ANÁLISE E IMPLEMENTAÇÃO DE IDS (Intrusion Detection System) EM
REDES DE COMPUTADORES**

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal de
Ensino Superior de Assis, como
requisito do Curso de Graduação,
analisado pela seguinte comissão
examinadora:

Orientador: Prof. Me. Fábio Eder Cardoso
Área de Concentração: Informática

Assis
2012

FICHA CATALOGRÁFICA

Serrano, Denis da Silva

ANÁLISE E IMPLEMENTAÇÃO DE IDS (Intrusion Detection System) EM REDES DE COMPUTADORES / Denis da Silva Serrano. Fundação Educacional do Município de Assis – FEMA – Assis, 2012.

36 Páginas

Orientador: Me. Fábio Eder Cardoso

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1.Segurança 2.IDS 3. *Snort*.

CDD: 001.61
Biblioteca da FEMA

DEDICATÓRIA

Dedico este trabalho aos meus pais, ao orientador Fabio Eder pela sua compreensão, auxílio e sabedoria e a todos meus amigos que me ajudaram e apoiaram na jornada.

AGRADECIMENTOS

Agradeço a Deus por me ajudar nessa caminhada, que meu deu sabedoria para adquirir conhecimentos necessários para enfrentar todas as dificuldades e obstáculos encontrados no caminho.

Ao meu orientador, Fábio Eder Cardoso, que me orientou com determinação, perseverança e sabedoria no desenvolvimento de todo trabalho.

Aos meus familiares e amigos que me apoiaram em momentos difíceis.

RESUMO

O estudo apresentado é voltado para a segurança da informação demonstrando claramente suas vulnerabilidades, apresenta-se à Internet, onde o índice de ataques ocorrem com maior frequência. O *Snort* é uma excelente ferramenta de detecção de intrusão, sendo que, após o monitoramento de tráfego de pacotes na rede, foi gerado um *log*, mostrando que ataques fazem parte do cotidiano e é necessário o uso de sistemas específicos para garantir nossa segurança, sendo que o IDS - Sistema de Detecção de Intrusão vai favorecer a integridade em redes de computadores e nos nós que estão conectados à mesma. Assim, nesse trabalho será realizado monitoramento e varreduras constantes nos pacotes de rede de computadores com a ferramenta *Snort* identificando um possível ataque ou anomalias.

Palavras-chaves: Segurança, IDS, Snort.

ABSTRACT

The present study is focused on information security and is a clear example of the vulnerabilities, we present the Internet, where the rate of attacks occur more often. Snort is an excellent tool for intrusion detection, after monitoration of packet traffic on the network, a log was generated, showing that attacks are part of everyday life and we must use the tools necessary to ensure our security, and the IDS - Intrusion Detection System will increase integrity in computer networks and nodes that are connected to the network. Thus, this work will be performed in constant monitoring and scans packet computer network with Snort tool identifying a possible attack or anomalies.

Keywords: Security, IDS, Snort.

LISTA DE ILUSTRAÇÕES

Figura 1. Demonstrativo estratégico, processos, tecnologias e pessoas	15
Figura 2. Princípio da Segurança da Informação	16
Figura 3. Arquitetura do Snort	26
Figura 4. Oracle VM Virtual Box Gerenciador	27
Figura 5. Tela de Configurações dos Sistema Operacional Virtual	27
Figura 6. Arquivo de Configurações do Snort.....	28
Figura 7. Continuação Arquivo de Configuração do Snort	29
Figura 8. Tela de Rules ou regras	29
Figura 9. Arquivo de Rules ou Regras	30
Figura 10. Tela de inicialização captura de pacotes do Snort	31
Figura 11. Log de Monitoramento Snort 1	32
Figura 12. Log de Monitoramento Snort 2	33
Figura 13. Log de Monitoramento Snort 3.....	34
Figura 14. Tela do Atacante IP 10.1.1.10.....	34
Figura 15. Log de Monitoramento Snort 4	35
Figura 16. Tela do Atacante IP 10.1.1.10.....	36
Figura 17. Log de Monitoramento Snort 5.....	36
Figura 18. Tela do BackTrack realizando nmap.....	37
Figura 19. Tela de Monitoramento Snort 6.....	38
Figura 20. Tela do BackTrack realizando ettercap	39
Figura 21. Tela de Monitoramento Snort 7	40
Figura 22. Trafego de Protocolos.....	41
Figura 23. Logs de monitoramentos de pacote de rede salvos.....	42

LISTA DE TABELAS

Tabela 1. Top Level Domains (TLD)	17
---	----

SUMÁRIO

1. Introdução	12
1.1 Objetivos	13
1.2 Justificativas	13
1.3 Motivações.....	13
1.4 Perspectivas de Contribuição	13
1.5 Metodologia de Pesquisa.....	13
1.6 Estrutura do Trabalho	14
2. Segurança de Redes	15
2.1 Sistema de nomes e domínios (DNS).....	17
2.2 Formas de Ataque e invasões	19
2.3 Ferramentas para invasão	20
3. IDS (Intrusion Detection System)	21
3.1 Locais de Monitoramento da Informação.....	22
4. SNORT	24
4.1 Arquitetura e Funcionamento do Snort.....	25
5. Gerando Log no Snort	27
6. Conclusão	43
7. Trabalhos Futuros	43
Referências Bibliográficas	44

1. INTRODUÇÃO

Atualmente, a segurança de redes de computadores é quesito essencial quando se fala em redes de médio e grande porte. Um exemplo claro das vulnerabilidades, apresenta-se à *Internet*, onde o índice de ataques ocorrem com maior freqüência.

A utilização do métodos de detecção de intrusão começou a ser utilizado nos últimos anos. Neste método pode-se coletar pacotes de dados que trafegam na rede e utilizar as informações de tipos de ataque conhecidos para verificação de intrusos tentando invadir a rede. As Informações coletadas apresentam-se como um ótimo histórico para que sejam utilizadas para melhor segurança da rede.

Para controle da informação temos várias ferramentas de avaliação de vulnerabilidades as quais estão disponíveis no mercado. Dentre elas, contamos com um sistema de segurança da informação que consiste em *IDS* - Sistema de Detecção de Intrusão, usado para descobrir se houve invasão do sistema ou tentativa de acesso indevido à rede; pode-se utilizá-la também como Firewall com a finalidade de bloquear o tráfego de dados da rede, tanto de entrada quanto na saída desses dados.

A Finalidade em um *IDS* - Sistema de Detecção de Intrusão é detectar uma invasão. Refere-se aos processos de monitoramento de atividades em computadores e rede e de análise dos eventos para a busca de sinais de invasão. O foco dos sistema de detecção de intrusão está na procura por sinais de possíveis invasões e alertar os administradores de redes para potenciais ameaças e falhas na segurança de rede ou sistemas.

De modo geral, o *IDS* - Sistema de Detecção de Intrusão é uma solução passiva, onde detecta uma possível violação da segurança, registra a informação em (*Log*) e emite um alerta para os administradores de sistemas ou de redes ou reprogramando um Firewall para bloquear as fontes maliciosas e suspeitas.

1.1 OBJETIVO

O presente trabalho objetiva o estudo e prática sobre ferramentas *IDS* - Sistema de Detecção de Intrusão as quais irão favorecer a integridade em redes de computadores e nos nós que estão conectados à rede. Realizar monitoramento e varreduras constantes nos pacotes da rede, identificando como anomalias.

1.2 JUSTIFICATIVA

A necessidade da Segurança da Informação e integridade dos sistemas levou à implementação desse modelo de segurança, uma vez que as redes de computadores, quer em seu porte médio, quer em seu porte maior, não apresentam, com frequência, ferramentas de monitoramento suficientes. Podendo sofrer qualquer ação que comprometa as confidencialidades e integridade dos dados.

1.3 MOTIVAÇÃO

Os benefícios de uso que este modelo de segurança pode trazer para uma organização resulta em um nível aceitável de confidencialidade, integridade e disponibilidade das informações. Com esta implementação, haverá uma manutenção mais segura dos sistemas e arquivos das organizações, ocorrendo todo monitoramento da rede, que resultará em um arquivo de histórico de ameaças e de tentativas de intrusões ou uso malicioso dentro da organização.

1.4 PERSPECTIVAS DE CONTRIBUIÇÃO

A perspectiva é de que pequenas e grandes empresas adotem esse programa de segurança em que, seus sistemas e dados serão privativos com a implementação IDS - Sistema de Detecção de Intrusos.

1.5 METODOLOGIA DE PESQUISA

Para o desenvolvimento deste trabalho foram pesquisadas obras de outros autores, trabalhos de conclusões de cursos e apostilas. Também foi utilizado o método de simulação, através de máquina virtual, para implementação da ferramenta.

1.6 ESTRUTURA DO TRABALHO

Este trabalho foi organizado em sete capítulos, sendo o primeiro esta introdução.

No segundo capítulo, será mostrado segurança de redes.

No terceiro capítulo, será apresentado o *IDS*.

No quarto capítulo, será apresentado o *Snort*.

No Quinto capítulo, será apresentado os *logs* de monitoramento do *Snort*.

No sexto capítulo, será apresentado a Conclusão.

No Sétimo capítulo, será apresentado os trabalhos futuros.

2. SEGURANÇA DE REDES

As informações, independente de seus formatos, são um dos maiores patrimônios em uma organização moderna. Nos últimos anos, a informação tem evoluído de forma muito rápida, dessa forma, fazendo as organizações ter em maior eficiência e rapidez nas tomadas de decisões e assim, as chances de uma empresa não utilizar o sistema de informações ficando praticamente nula. Neste caso, o mecanismo de segurança da informação ficou muito importante para a sobrevivência e comunicação desta organização. A informação, no passado, era muito simples, onde ficavam contidas em inúmeros papéis e podiam ser trocadas fisicamente. Com a chegada da tecnologia digital, este processo tornou-se muito mais complexo. Hoje, com a *Internet* e redes de comunicação de dados, os computadores estão conectados uns aos outros e, por consequência, os formatos digitais são portáteis. Tal situação tornou-se atrativo para os ladrões. (LAUREANO, 2005).

Dessa maneira, pode-se dizer que não existe uma segurança absoluta, portanto torna-se extremamente necessário agir no sentido de descobrir quais são os pontos vulneráveis e, a partir dessa situação, avaliar os riscos e os impactos de modo a providenciar a eficácia da segurança da informação.

Conforme demonstrado por (LAUREANO, 2005) a figura abaixo ilustra o ponto de vista estratégico, o relacionamento dos processos, tecnologias e pessoas.

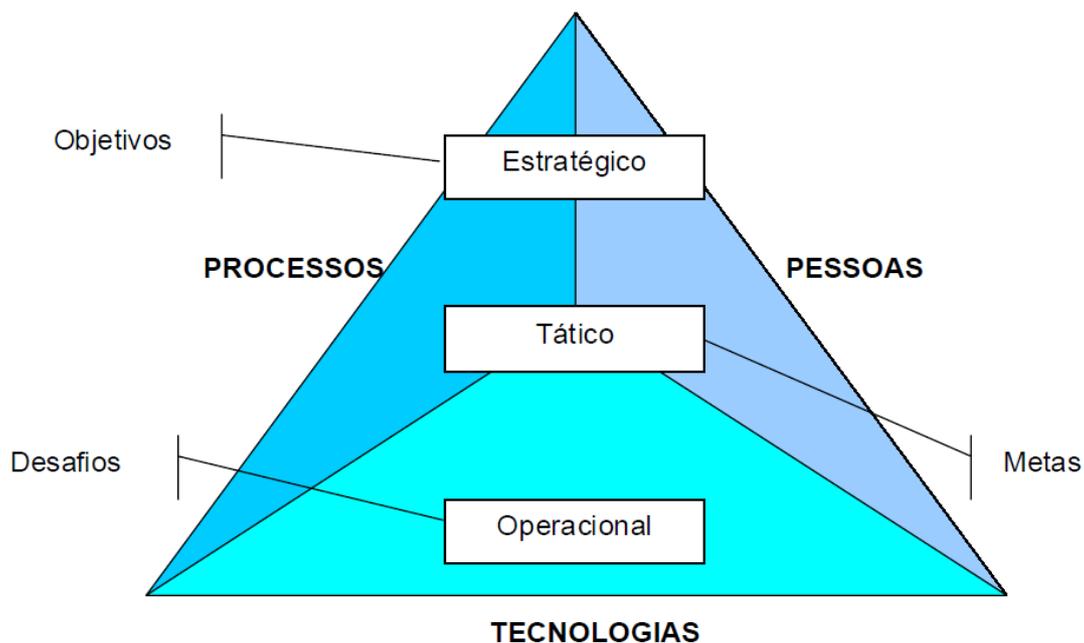


Figura 1 - Demonstrativo estratégico, processos, tecnologias e pessoas

Fonte: (Gestão de Segurança da Informação, p4)

Conforme citado por LAUREANO (2005), uma empresa que utiliza a segurança da informação, constitui-se de alguns princípios básicos para a obtenção dessa garantia. São eles:

Confidencialidade - somente pessoas autorizadas, tendo garantido a identificação e autenticação dos usuários.

Disponibilidade - garantir que as informações, quando necessárias, estejam disponíveis para uso.

Integridade - garantir que as informações estejam protegidas contra modificações intencionais ou acidentais e que a informação possa retornar no momento original em que foi gravada.

Auditoria - Identificar os passos de processos que foram realizados, onde se identificam os participantes do processo, os horários e locais de cada etapa realizadas. Com esse histórico de eventos podemos determinar quando e onde ocorreu uma violação da informação. A figura abaixo ilustra os princípios da segurança da informação.

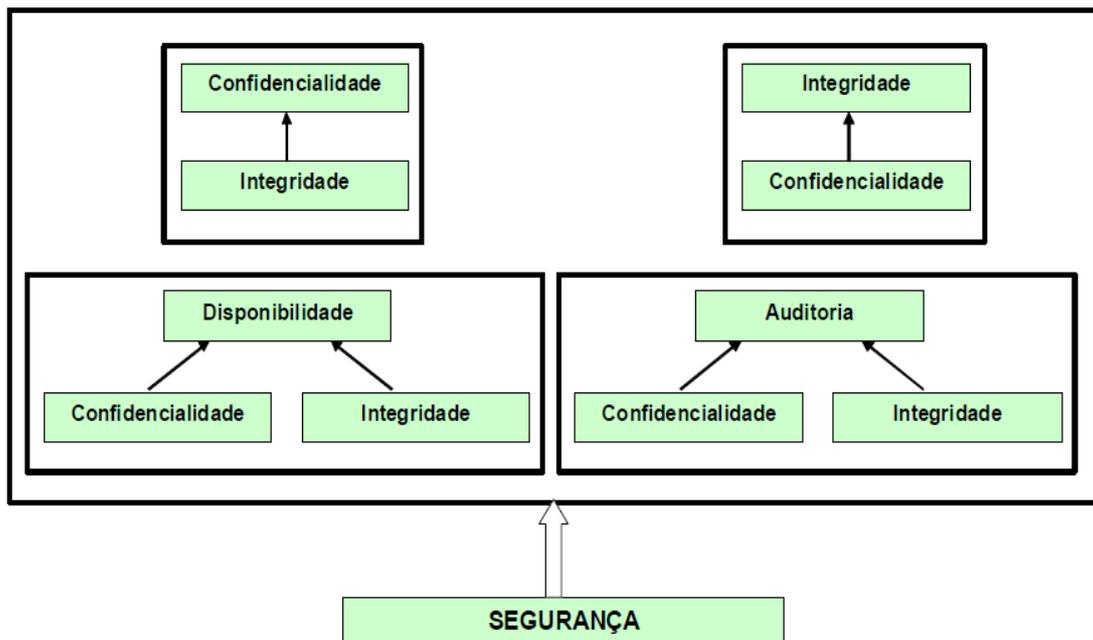


Figura 2 - Princípio da Segurança da Informação.
Fonte: (Gestão de Segurança da Informação, p13)

2.1 Sistema de nomes e domínios (DNS)

Os Sistemas de DNS servem basicamente para mapear nomes em endereços IP e vice-versa.

Os serviços de DNS caracterizam em uma interação de cliente servidor onde um aplicativo necessita de outro para a tradução de nomes em endereços IP; o cliente solicita uma mensagem de requisição para um servidor de nomes, retornando uma mensagem de resposta ou torna-se cliente de outro servidor de DNS até que seja encontrada a requisição solicitada. Os bancos de dados de nomes nos servidores estão localizados em diversas localidades do mundo, assim tornando um DNS de banco de dados distribuído.(BURITI, 2006)

“Os nomes de domínio obedecem a uma hierarquia, sendo lida da esquerda para direita o nível hierárquico. Quando mais á direita, maior o nível. Os nomes á esquerda representam os nomes das máquinas e os da direita os subdomínios e domínio a qual a máquina pertence. O DNS especifica valores para os níveis mais altos, chamados TLDs (Top Level Domains). Todos os TLD´s são mostrados na Tabela 1(Atualizado de acordo com o ICANN, em 05-jan-2006):”

TLD	Data de introdução	Propósito	Responsável
.aero	2001	Indústria de transportes aéreos	Societe Internationale de Telecommunications Aeronautiques SC, (SITA)
.biz	2001	Negócios	NeuLevel
.cat	2005	Comunidade lingüística e cultural Catalã	Fundació puntCAT - Espanha
.com	1995	Irrestrito, mas com intenções de registros comerciais.	VeriSign, Inc., USA
.coop	2001	Cooperativas	DotCooperation, LLC
.edu	1995	Organizações educacionais	EDUCAUSE, USA
.gov	1995	Organizações governamentais	US General Services Administration
.info	1998	Uso irrestrito	Afilias Limited, Irlanda
.int	1998	Organizações estabelecidas em tratados governamentais	Internet Assigned Numbers Authority, USA
.jobs	2005	Comunidade Internacional de gerência de recursos humanos.	Employ Media LLC, USA
.mil	1995	Organizações militares	US DoD Network Information Center, USA
.museum	2001	Museus	Museum Domain Management Association, (MuseDoma)
.name	2001	Para registros individuais	Global Name Registry, LTD, UK
.net	1995	Irrestrito, mas voltado para organizações que de alguma forma colaboram ou administram a Internet	VeriSign, Inc., USA
.org	1995	Organizações não-governamentais e sem fins lucrativos	Public Interest Registry. Until 31 December 2002, .org was operated by VeriSign Global Registry Services, USA.
.pro	2002	Profissionais liberais	RegistryPro, LTD, USA
.travel	2005	Comunidade de viagens e turismo	Tralliance Corporation Tralliance Corporation, USA

Tabela 1 – Top Level Domains (TLD)

Fonte: (Extensões de Segurança para o DNS- Set. 2012, p16)

2.2 Formas de ataque e invasões

Programas ou códigos, por serem escritos por seres humanos, estão sujeitos a falhas, desse modo, vulneráveis a *bugs*. As vulnerabilidades não estão somente nos códigos do programa, os administradores utilizam também as desatualizações dos sistemas onde as *patch* das correções não são atualizadas. (SANTOS, 2005).

Os erros podem acontecer em qualquer sistema, sendo que, os mais graves, são aqueles conectados à rede. O sistema operacional, atualmente, não oferece garantia com relação à segurança da rede, por não apresentar organização imune às falhas. Neste caso, os administradores se valem do uso de ferramentas, já disponibilizadas na seguranças do programa. SANTOS (2005).

Segundo SANTOS (2005) existem recomendações básicas para minimizar os problemas de segurança, sendo elas:

- Os Sistemas operacionais deve estar sempre atualizados, principalmente relacionado aos dispositivos de redes.
- Deixar somente os serviços necessários executando.
- Exigir senhas administradoras para executar as tarefas.
- Para segurança de servidores usar ferramentas apropriadas e seguras.

Atualmente, os invasores exploram vulnerabilidades como, por exemplo, uma má configuração do sistema operacional, existindo diversas formas de ataques.

Probing

São informações obtidas pela rede onde é possível utilizá-las para uma possível invasão.

Podem ser empregados para descobrir os serviços disponíveis na rede. (SANTOS, 2005).

Trojan Horses

São programas que são executados para realizar alguma tarefa sem que o usuário perceba assim afetando a segurança do sistema.

A execução dos *Trojan* permite que terceiros acompanhem o que está sendo digitado no teclado. (SANTOS, 2005).

2.3 Ferramentas para invasão

Nmap

O *nmap* é um escaneador de portas lógicas que é usado para verificar o estado do seu alvo, de maneira geral, ele fornece uma relação de computadores e serviços ativos como, por exemplo, as portas de protocolos abertas identificam os serviços da rede de computadores para determinar os nomes das aplicações e o números das versões que estão sendo executado na máquina destino. (Nogueira, 2009).

Ettercap

O *ettercap* é um programa com *sniffer*, ou seja, um capturador de dados em redes locais. O *sniffer* são farejadores para capturar os tráfego da rede. Com as capturas dos tráfegos da rede, consegue-se as senhas digitadas por outros usuários e, com análises dos dados, pode-se capturar as conversas ou mensagens do tipo *MSN*.

O *ettercap* contém também técnicas de *Spoofing* de DNS; com isso, utilizamos para todas as pessoas de uma determinada rede que consultam servidores DNS, direcionados para onde desejarem. (RIBEIRO, 2011).

Metasploit

O *metasploit* é uma plataforma que permite verificações do estado de segurança dos computadores; ele ataca as falhas de segurança existentes em diversos softwares. *Metasploit framework* está organizado em diversos módulos de ataques que contém programas para verificar as vulnerabilidades encontradas nos *softwares* e sistemas operacionais, desse modo, permitindo a execução de códigos maliciosos, levando à invasão da máquina. (ARAGÃO, 2011).

Os programas são chamados de *exploits* e os código maliciosos de *payload*, onde os *exploits* atacam as falhas dos softwares e, em seguida, executando o *payload*, após esse processo, cria-se uma sessão remota de SSH ou *Telnet*, assim permitindo o controle do computador atacado. (ARAGÃO, 2011).

3. IDS (Intrusion Detection System - Sistema de Detecção de Intrusos)

Conforme dito por *Roesch, M. (2001)* as ferramentas IDS surgiram por conta dos ataques pertinentes em redes de computadores. O objetivo do IDS é gerar alertas de intrusos e detectar pacotes que possam ser um possível ataque.

As diversas Ferramentas IDS, basicamente todas elas, têm funcionamento parecido, ou seja, todas baseadas em assinaturas; os pacotes das redes são comparados com anomalias detectados como possíveis ataques.

O objetivo básico das ferramentas IDS é nos mostrar log's de informações como: qual foi origem dos ataques, quantidade de ataques por dia e o tipo que foi usado para ataque.

Segundo ROESCH (2001) não é fácil descobrir se o computador foi invadido ou não por pessoas más intencionadas; com isso, é preciso analisar a rede, podendo ser verificados os seguintes itens:

- Processos sendo executados, mas não autorizados;
- Registros de *log's* (Registros de armazenamentos de eventos);
- Contas de Usuários;
- Sistema de Arquivos Alterados.

Para os responsáveis da área de segurança de rede é muito difícil detectar invasões. Sem total segurança há a necessidade de utilizar os sistemas de detecção de intrusos (*SDI*).

O *SDI* se constitui em um banco de dados que armazena as assinaturas ou códigos de ataques. Se essas assinaturas, como referência, forem identificadas em um fluxo na rede serão detectadas como um possível ataque; desse modo, é essencial que esse banco de dados ou assinaturas propriamente ditas, devam ser sempre atualizadas. (ROESCH, 2001).

Conforme as regras de alertas configuradas no *IDS*, pode haver a comunicação com o administrador da rede para um alerta sobre detecção de intrusos como, por exemplo, enviando arquivos de *log*, emitindo alguns alertas de sons ou enviando um e-mail. Com esse avisos, podem se tomar as devidas providências para que sejam barrados essas tentativas de invasões. (ROESCH, 2001).

Nos *IDS* temos dois tipos de análise de *log's*, sendo eles: os falsos positivos e falsos negativos.

Falsos positivos são uma atividade normal da rede, onde começa-se a enviar *log's* de detecção de intrusos para o administrador; na verdade o IDS está configurado de modo errado. (ROESCH, 2001).

Falsos negativos são ataques nos *IDS*, mas não identificados como ameaças, portanto, passam despercebidos. Assim, não enviando os *log's*, uns dos motivos por acontecer essa divergência é devido à não atualização das assinaturas do banco de dados. (ROESCH, 2001).

3.1 Locais de Monitoramento da Informação

Os locais de monitoramentos da informação do IDS podem ser classificados por dois tipos de segurança (ROESCH, 2001).

NIDS - (*Network Intrusion Detetion System* - Sistemas de Detecção de Intrusão de Rede): tem por objetivo controlar e monitorar o tráfego de pacotes na rede *WAN*.

Conforme dito por (ROESCH, 2001), os *NIDS* basicamente são sensores ocultos posicionados na rede para rastrear e monitorar todos os pacotes que estão trafegando na rede que podem ser um possível ataque.

Vantagens dos NIDS são:

- Um único sensor no seguimento da rede pode monitorar todos os dados que trafegam com isso reduzindo a implementação.
- Os *NIDS* não conseguem manipular o tráfego da rede, somente analisar os dados que estão passando na rede.
- Os sensores são invisíveis para os invasores. Eles não geram nenhuma informação que possa indicar sua localização na rede.

Desvantagens dos NIDS:

- Se estiver vários dados trafegando na rede, o IDS não consegue identificar o ataque, acaba passando despercebido.
- Ele não consegue comparar as regras com os conteúdos de dados criptografados.
- Alguns *NIDS* não conseguem remontar dados quebrados ou fragmentados, assim, não sabendo se o ataque foi bem sucedido ou não.

HIDS - (*Host Intrusion Detection System* - Sistemas de Detecção de Intrusos de Host), tem por objetivo controlar e monitorar os Sistemas e arquivos dos Hosts.

Os *HIDS* são monitoramentos com a capacidade de verificar as mudanças de atividades das aplicações dos hosts em nível de sistema operacional. (ROESCH, 2001).

Vantagens do HIDS:

- Por analisar os *hosts*, consegue-se monitorar os dados antes de serem criptografados ou depois dos dados serem criptografados;
- Consegue monitorar as alterações dos arquivos nos sistema;
- Monitora ataques de vírus ou qualquer outro tipo de brecha e integridade no sistema.

Desvantagens do HIDS:

- Há dificuldades no gerenciamento de vários *HIDS*,
- Caso algum ataque consiga entrar em algum Host que esteja instalado, o SDI poderá comprometer toda a segurança do sistema;
- Dependendo da estrutura da segurança da empresa exige-se certa quantidade de espaço em disco para armazenamento dos *log's*,
- Prejudica os *Hosts* onde estão instalados o *SDIH*, porque consome os recursos da máquina, com os processamentos das regras dos *IDS*.

4. SNORT

De acordo com *Roesch(2001)*, o *SNORT* é uma das melhores ferramentas de Sistema de Detecção de Intrusão "open-source", a ferramenta *SNORT* foi desenvolvida por Martin Roesch, fundador e CTO da *Sourcefire* em 2001.

O *Snort* tem a capacidade de analisar os registros de pacotes em redes TCP/IP em tempo real; executa análises de protocolo, associa e busca padrões dos conteúdos podendo ser usados para detectar os ataques, sendo eles: *porstscan*, *stelh*, *Buffer*, *overflows*, entre outros. (MEDRADO, PEREIRA, 2011).

As atualizações e desenvolvimento das regras de detecção de intrusão do *Snort* são realizados diariamente.

O *Snort* contém módulos poderosos capazes de controlar e produzir uma grande quantidade de dados gerados pelos ataques, podendo avaliar, tanto o corpo dos pacotes, quanto o cabeçalho dos pacotes; além disso, tem a opção de capturar a sessão inteira.

Snort pode assumir três modalidades *Sniffer*, *Packet logger* e *Network intrusion detection system*.

- *Sniffer*: Captura os pacotes de dados e imprime continuamente na console;
- *Packet logger*: Registra todos os pacotes de dados no disco rígido;
- *Network Intrusion detection*: Analisa todo o tráfego da rede, conforme as regras já definidas pelo Administrador, onde são executadas as devidas ações.

4.1 Arquitetura e Funcionamento do SNORT

Conforme Medrado, S. A., Pereira, J. C. (2011), O *Snort* é composto de três subsistemas primários, sendo eles: Decodificador de pacotes, engenharia de detecção e subsistema de *log* e alerta.

Decodificação de pacotes é quando sai do nível de dados, vai para o nível de transporte e termina no nível da aplicação. A funcionalidade de decodificação de pacotes consiste de ponteiros para identificação dos pacotes de dados, para que, mais tarde, possam ser analisados pela arquitetura de detecção de Intrusos. (ROESCH, 2001).

Engenharia de detecção mantém suas regras de detecção de intrusão em duas listas que seria: *Chain Headers* - Cabeçalho de regras e *Chain Options* - Cabeçalho de opções. (ROESCH, 2001).

- *Chain Headers* basicamente contém os atributos comuns de uma regra.
- *Chain Options* armazena dentro de cada pacote os padrões de ataques, e, conforme o tipo do padrão, as ações serão tomadas.

Subsistema de log e alerta: os *logs* de sistema podem ser armazenados parcialmente ou incompletos para que seja agilizada a performance. Quanto aos alertas, o administrador pode ser avisado por mensagens enviadas através do *syslog*, que seria um *software* responsável de armazenar os *logs* no Sistema Operacional ou armazenados em um arquivo texto, desde que possa ser utilizada em uma multiplataforma. (ROESCH, 2001).

O *SNORT*, conforme ilustra a figura 03, contém uma arquitetura composta por quatro componentes sendo: Farejador, Pré-processador, Mecanismo de detecção e Alertas/Registro.

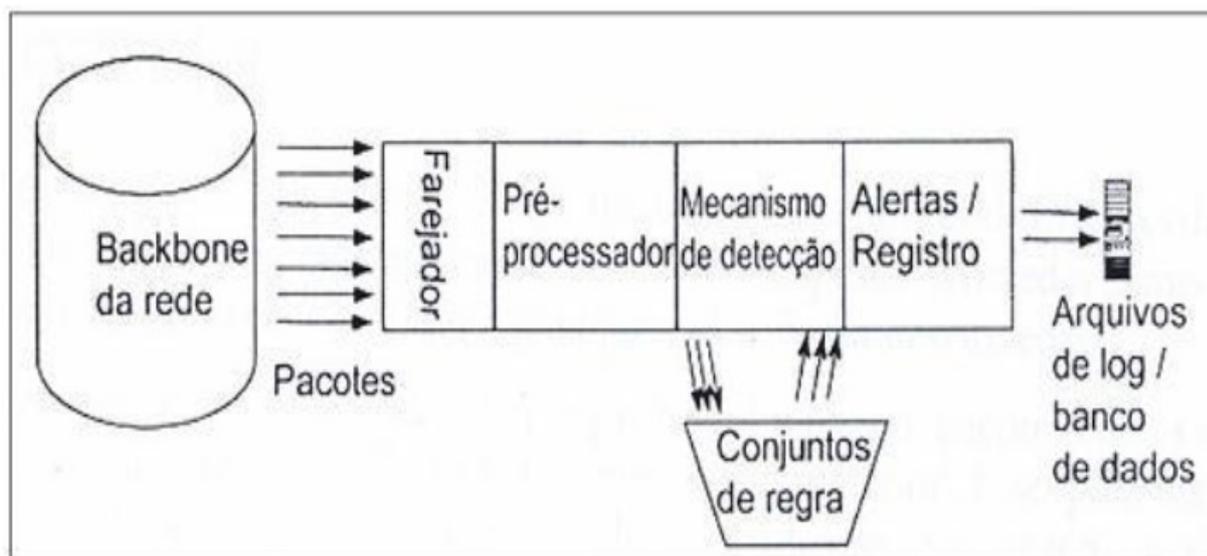


Figura 3 - Arquitetura do *Snort*

Fonte: (<<ftp://fipp.unoeste.br/artigos/artigos-ids/Sergio-Medrado.pdf>>,p3)

Farejador: captura todos os pacotes da rede, verifica se realmente são pacotes maliciosos e como devem proceder ao pré-processador. Em seguida, ordena os tipos de pacotes e finalmente o *Snort* registra, armazena no banco de dados, para que, desta maneira, possa gerar alertas ou ativar regras e, assim, o administrador decidirá o que fazer com os pacotes. (ROESCH, 2001).

Regras do Snort

O *Snort* consiste em um formato de regras. É utilizado por profissionais da segurança da informação em todo o mundo. As regras abertas dão aos clientes a capacidade de verificar se as regras já criadas estão fornecendo proteção completa contra a vulnerabilidade da rede. Podendo, assim, criar novas regras ou modificar as existentes para detectar problemas com os serviços personalizados ou problemas incomuns. (ROESCH, 2001).

5. Gerando logs no Snort

Para gerar os *logs* no *Snort* serão utilizadas as seguintes ferramentas conforme abaixo na Figura 4: Oracle VM Virtual Box, para virtualizar os Sistemas operacionais, sendo Windows XP como cliente e o S.O Debian como servidor do *Snort* e Virtualização do *BackTrack*, para simular os ataques com *Nmap* e *Ettercap*.

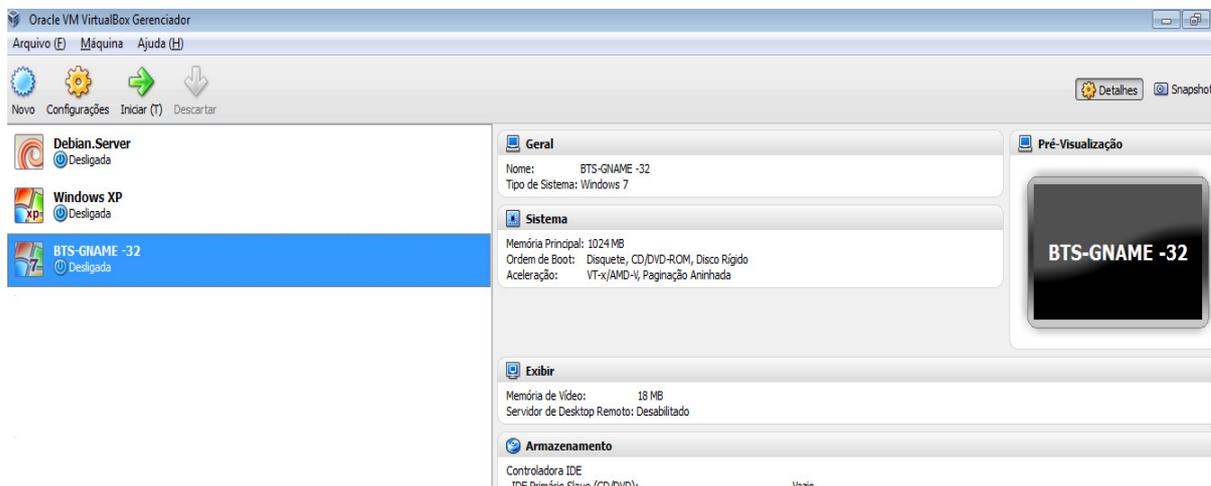


Figura 4 - Oracle VM Virtual Box Gerenciador

Na Figura 5, virtualização na virtual Box dos sistemas operacionais, a função é verificar se nas placas de rede estão todas em modo Promísco, para que tenha o tráfego de todos os pacotes entre as virtuais.

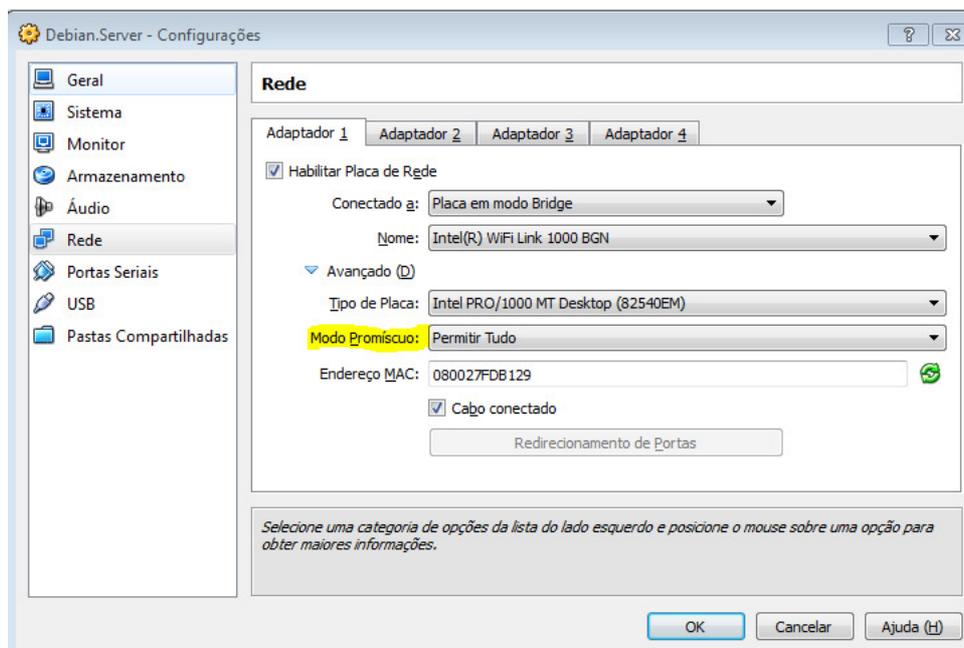


Figura 5 - Tela de Configurações dos Sistema Operacional Virtual

De modo que a máquina possa se comunicar, é preciso configurar as placas de rede `/etc/network/interfaces`, onde foi identificado com *IPs*, o Debian IP 10.1.1.1, Windows XP IP 10.1.1.10 e o *Back Track* IP 10.1.1.12

Para instalar o *Snort* no Debian é muito simples: estando conectado na internet, basta executar o seguinte comando (`apt-get install Snort`).

O comando para acessar diretório de configuração *Snort*: `/etc/snort/snort.conf`.

Na figura 6, mostra o arquivo de configuração, onde dentro do arquivo contém as opções para criar suas próprias configurações, sendo elas: Definir as Variáveis para a sua rede; Configurar Dinâmicas bibliotecas carregadas; Configurar o *preprocessors*; Configurar *Plugins de Saída*; Adicionar qualquer *plugins* em tempo de execução; Personalizar seu conjunto de regras.

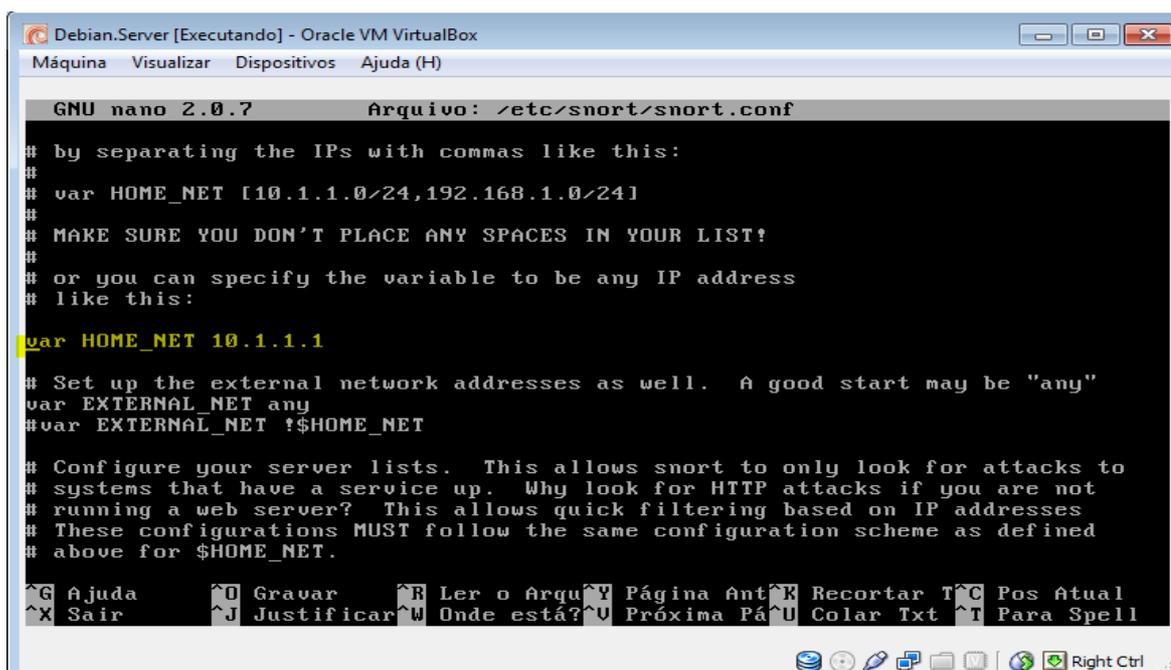
```

GNU nano 2.0.7      Arquivo: /etc/snort/snort.conf
-----
# http://www.snort.org      Snort 2.8.5.2 Ruleset
# Contact: snort-sigs@lists.sourceforge.net
-----
# $Id$
#
#####
# This file contains a sample snort configuration.
# You can take the following steps to create your own custom configuration:
#
# 1) Set the variables for your network
# 2) Configure dynamic loaded libraries
# 3) Configure preprocessors
# 4) Configure output plugins
# 5) Add any runtime config directives
# 6) Customize your rule set
#
#####
# Step #1: Set the network variables:
#
[ 924 linhas lidas ]
^G Ajuda      ^O Gravar    ^R Ler o Arqu^Y Página Ant^X Recortar T^C Pos Atual
^X Sair      ^J Justificar^W Onde está?^U Próxima Pá^U Colar Txt  ^T Para Spell

```

Figura 6 - Arquivo de Configurações do *Snort*

Na Figura 7 é a continuação das configurações da Figura 6, que mostra em *HOME_NET* se você pode especificar uma máquina, interface ou rede, na qual o *Snort* irá "escutar", ou ainda especificar uma lista dos itens acima mencionados, como demonstrado no exemplo. Em *EXTERNAL_NET* você irá configurar o que o *Snort* deve considerar como sendo uma rede externa; especificamos o parâmetro *any* para indicar qualquer rede.



```

GNU nano 2.0.7      Arquivo: /etc/snort/snort.conf
# by separating the IPs with commas like this:
#
# var HOME_NET [10.1.1.0/24,192.168.1.0/24]
#
# MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
# or you can specify the variable to be any IP address
# like this:
var HOME_NET 10.1.1.1

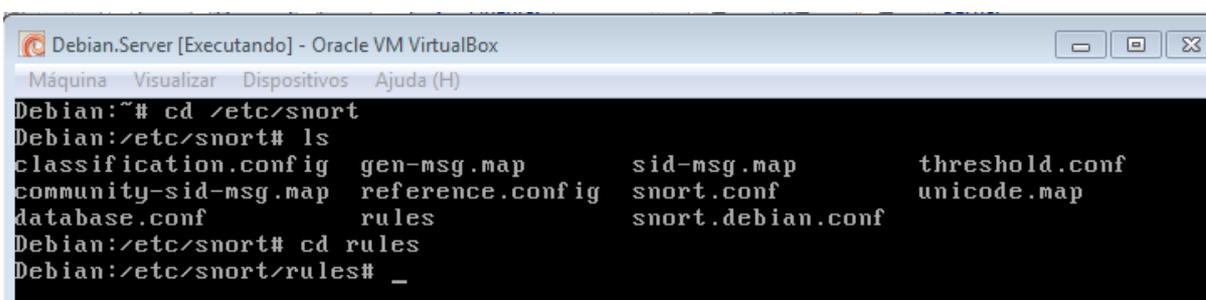
# Set up the external network addresses as well.  A good start may be "any"
var EXTERNAL_NET any
#var EXTERNAL_NET !$HOME_NET

# Configure your server lists.  This allows snort to only look for attacks to
# systems that have a service up.  Why look for HTTP attacks if you are not
# running a web server?  This allows quick filtering based on IP addresses
# These configurations MUST follow the same configuration scheme as defined
# above for $HOME_NET.
Ajuda      Gravar    Ler o Arq  Página Ant Recortar T  Pos Atual
X Sair      J Justificar W Onde está? U Próxima Pá U Colar Txt  T Para Spell

```

Figura 7 - Continuação Arquivo de Configuração do *Snort*

Na Figura 8 mostra o diretório do *Snort*. Temos as "rules" que seriam as regras de monitoramento para gerar os *logs* do *Snort*.



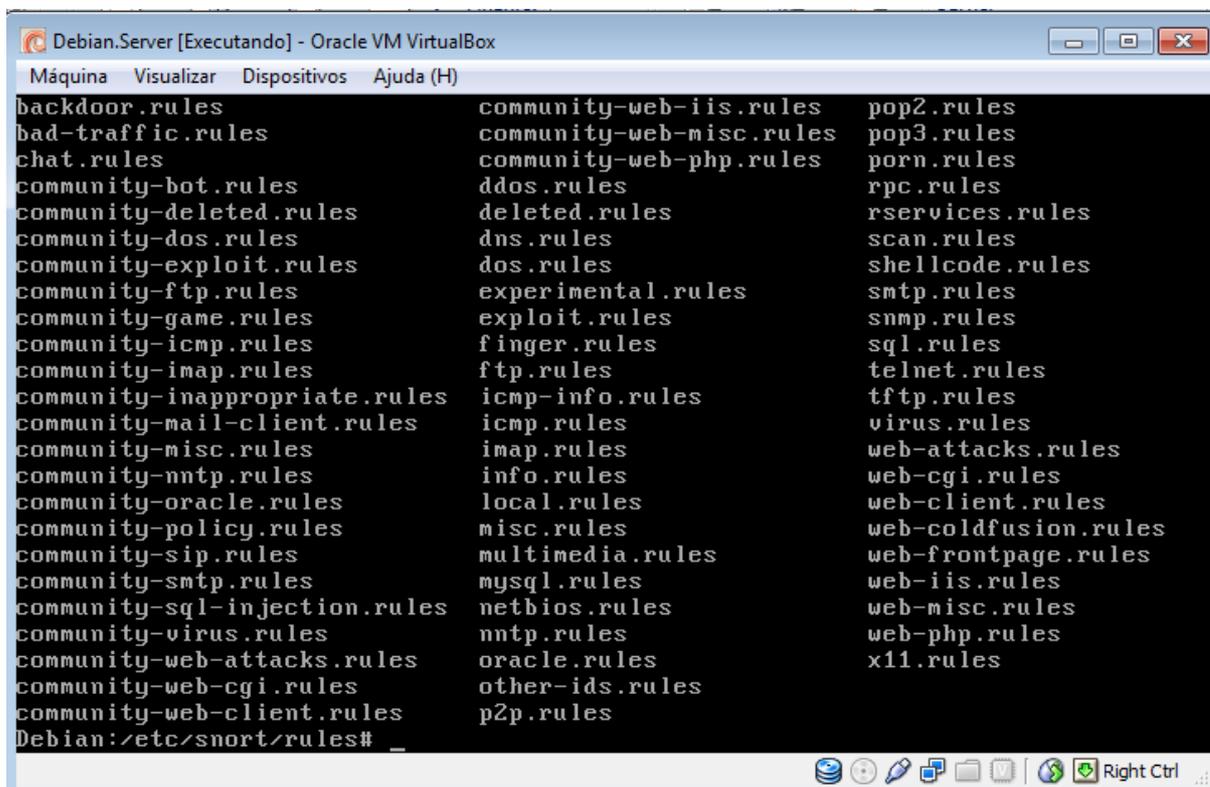
```

Debian:~# cd /etc/snort
Debian:/etc/snort# ls
classification.config  gen-msg.map          sid-msg.map          threshold.conf
community-sid-msg.map  reference.config     snort.conf           unicode.map
database.conf          rules                snort.debian.conf
Debian:/etc/snort# cd rules
Debian:/etc/snort/rules# _

```

Figura 8 - Tela de *Rules* ou regras

Conforme a Figura 9 abaixo, o *Snort* contém várias regras ou "rules" de monitoramento de tráfego dos pacote de rede, para emitir alertas de invasões para o administrador de rede, conforme a definição de cada regra.



```

backdoor.rules      community-web-iis.rules  pop2.rules
bad-traffic.rules   community-web-misc.rules pop3.rules
chat.rules          community-web-php.rules  porn.rules
community-bot.rules ddos.rules              rpc.rules
community-deleted.rules deleted.rules          rservices.rules
community-dos.rules dns.rules              scan.rules
community-exploit.rules dos.rules              shellcode.rules
community-ftp.rules experimental.rules     smtp.rules
community-game.rules exploit.rules          snmp.rules
community-icmp.rules finger.rules          sql.rules
community-imap.rules ftp.rules             telnet.rules
community-inappropriate.rules icmp-info.rules     tftp.rules
community-mail-client.rules icmp.rules          virus.rules
community-misc.rules imap.rules           web-attacks.rules
community-nntp.rules info.rules          web-cgi.rules
community-oracle.rules local.rules         web-client.rules
community-policy.rules misc.rules          web-coldfusion.rules
community-sip.rules multimedia.rules     web-frontpage.rules
community-smtp.rules mysql.rules         web-iis.rules
community-sql-injection.rules netbios.rules     web-misc.rules
community-virus.rules nntp.rules         web-php.rules
community-web-attacks.rules oracle.rules       x11.rules
community-web-cgi.rules other-ids.rules
community-web-client.rules p2p.rules
Debian:/etc/snort/rules# _

```

Figura 9 - Arquivo de *Rules* ou Regras.

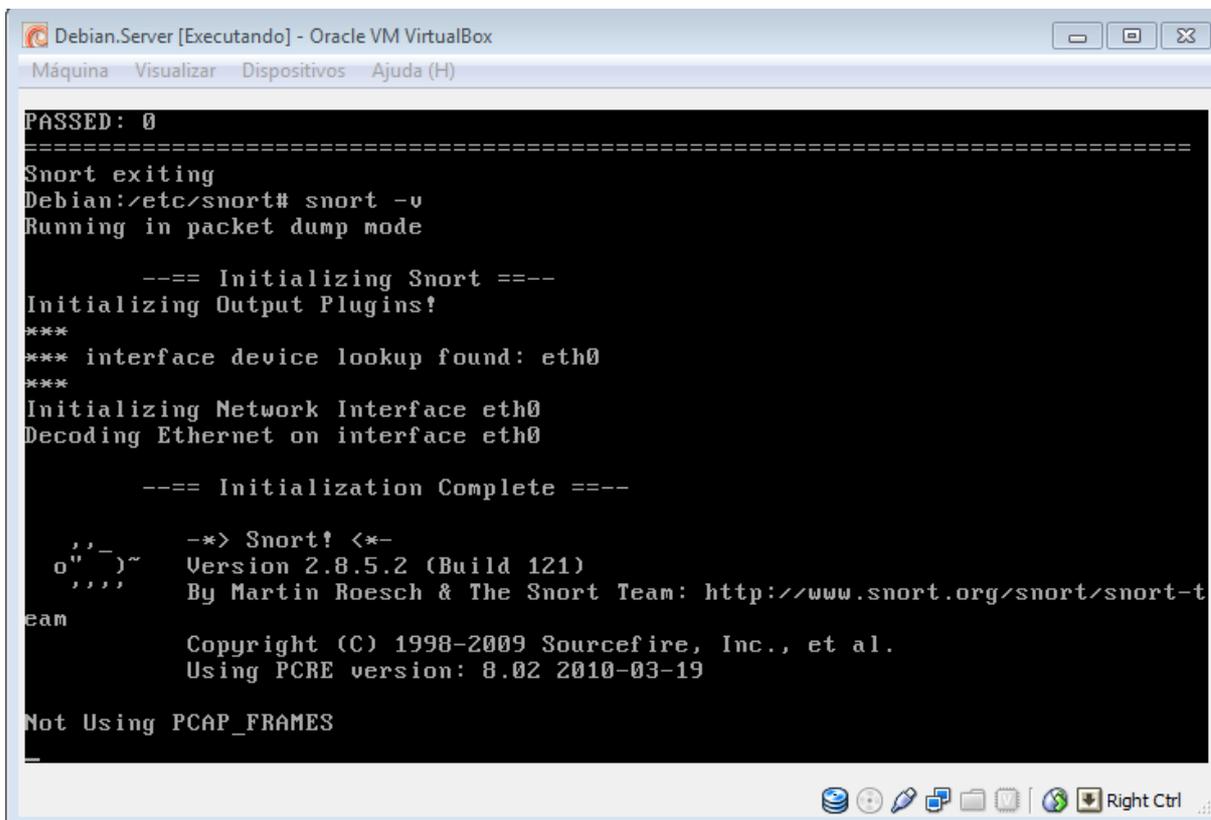
O *Snort* contém alguns comandos de inicialização de captura do tráfego de pacotes de rede.

Snort -v: mostra somente os cabeçalhos dos pacote TCP/IP na tela.

Snort -vd: Mostra somente os cabeçalhos do IP, TCP, UDP, e ICMP na tela.

Snort -vde: Mostra todos os cabeçalhos e os dados contidos neles também.

Na Figura 10 mostrada abaixo, foi realizado um dos comando citados acima para Inicialização de captura de pacotes da rede. O comando utilizado foi "*snort -v*"; nesse caso, serão capturados somente os cabeçalhos dos pacotes TCP/IP na tela.



```
PASSED: 0
=====
Snort exiting
Debian:/etc/snort# snort -v
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
***
*** interface device lookup found: eth0
***
Initializing Network Interface eth0
Decoding Ethernet on interface eth0

--== Initialization Complete ==--

,,_  -*> Snort! <*-
o" )~  Version 2.8.5.2 (Build 121)
' ' '  By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
eam
      Copyright (C) 1998-2009 Sourcefire, Inc., et al.
      Using PCRE version: 8.02 2010-03-19

Not Using PCAP_FRAMES
```

Figura 10 - Tela de inicialização para captura de pacotes do *Snort*

Na Figura 11 abaixo é o *log* de captura de pacotes da rede que o atacante está com IP 10.1.1.12 origem e o IP 10.1.1.10 sendo o alvo destino. Temos também as informações do tipo do protocolo *ICMP*. O *TTL* estabelece o tempo de vida do pacote através das métricas de números de saltos e tempo, etc. Já *Echo* seria a pergunta ou chamada do IP 10.1.1.10 para o IP 10.1.1.12; já o *Echo Reply* é a resposta do IP 10.1.1.12 para o IP 10.1.1.10.

```

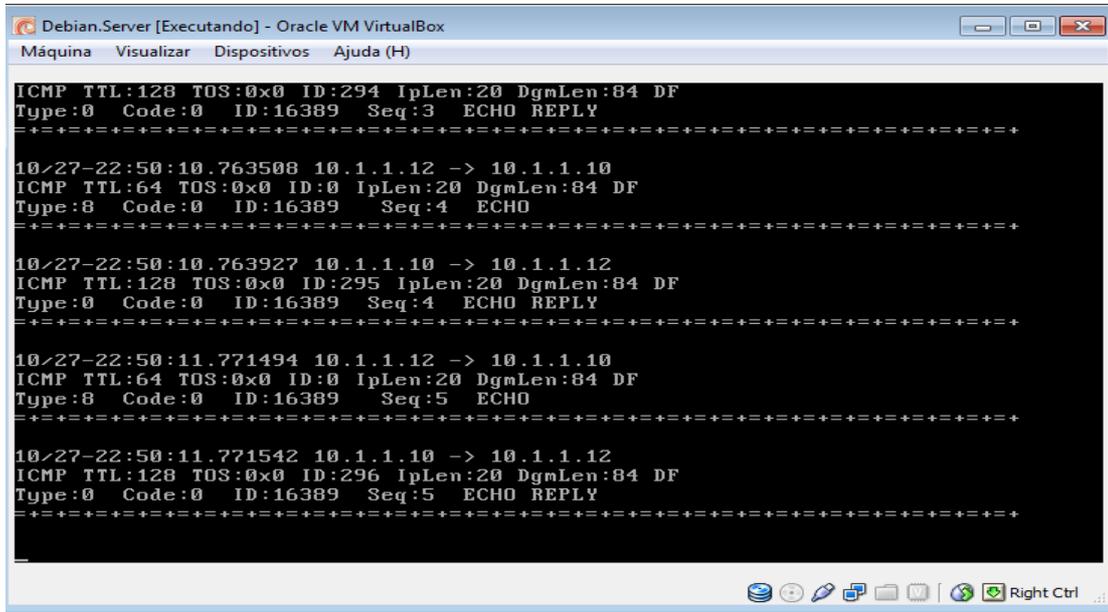
Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)

10/27-22:52:17.878913 10.1.1.10 -> 10.1.1.12
ICMP TTL:128 TOS:0x0 ID:300 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:1792 ECHO
=====
10/27-22:52:17.879287 10.1.1.12 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:42027 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:1792 ECHO REPLY
=====
10/27-22:52:18.885860 10.1.1.10 -> 10.1.1.12
ICMP TTL:128 TOS:0x0 ID:301 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:2048 ECHO
=====
10/27-22:52:18.887402 10.1.1.12 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:42028 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:2048 ECHO REPLY

```

Figura 11 - Log de Monitoramento Snort 1

Na Figura 12 mostrada abaixo é o mesmo tipo de *log* da figura 11. Nesse caso, é a captura de pacotes da rede que o atacante está com IP 10.1.1.10 origem e o IP 10.1.1.12 sendo o alvo destino.

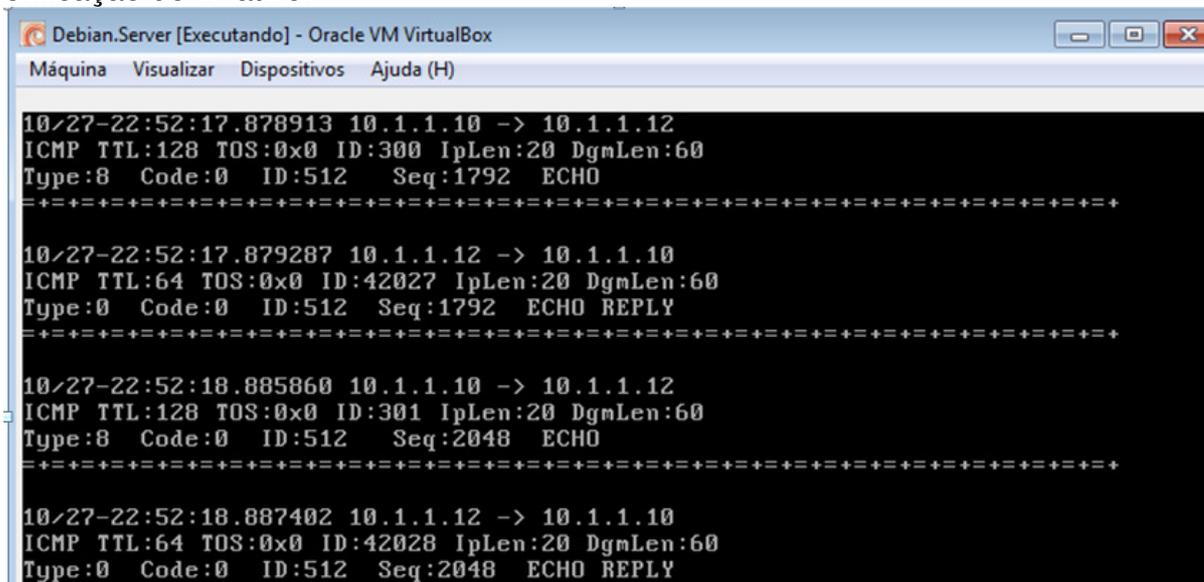


```
Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)

ICMP TTL:128 TOS:0x0 ID:294 IpLen:20 DgmLen:84 DF
Type:0 Code:0 ID:16389 Seq:3 ECHO REPLY
=====
10/27-22:50:10.763508 10.1.1.12 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:16389 Seq:4 ECHO
=====
10/27-22:50:10.763927 10.1.1.10 -> 10.1.1.12
ICMP TTL:128 TOS:0x0 ID:295 IpLen:20 DgmLen:84 DF
Type:0 Code:0 ID:16389 Seq:4 ECHO REPLY
=====
10/27-22:50:11.771494 10.1.1.12 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8 Code:0 ID:16389 Seq:5 ECHO
=====
10/27-22:50:11.771542 10.1.1.10 -> 10.1.1.12
ICMP TTL:128 TOS:0x0 ID:296 IpLen:20 DgmLen:84 DF
Type:0 Code:0 ID:16389 Seq:5 ECHO REPLY
=====
```

Figura 12 - Log de Monitoramento Snort 2

A imagem 13 demonstra o *log* de monitoramento da ferramenta *Snort* no momento de um possível ataque, O protocolo é ICMP (*Internet Control Message Protocol*) que é definida na camada de Internet e é usado pelo protocolo IP. Sendo assim, o atacante do IP 10.1.1.10 abriu um *ping* na vítima com IP 10.1.1.12 para verificação do IP ativo.



```

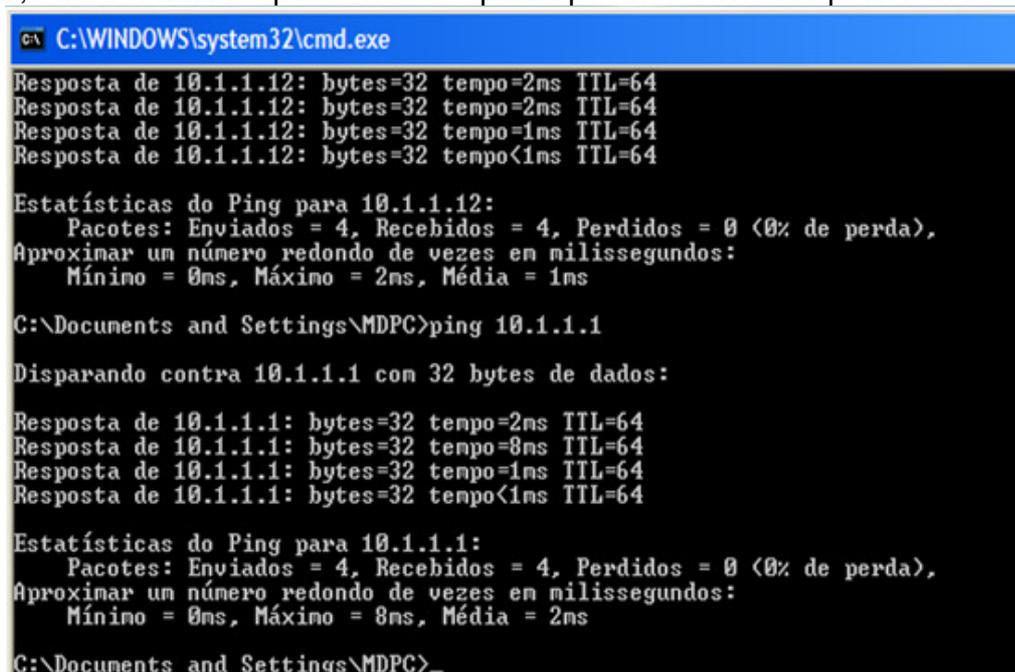
Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)

10/27-22:52:17.878913 10.1.1.10 -> 10.1.1.12
ICMP TTL:128 TOS:0x0 ID:300 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:1792 ECHO
=====
10/27-22:52:17.879287 10.1.1.12 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:42027 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:1792 ECHO REPLY
=====
10/27-22:52:18.885860 10.1.1.10 -> 10.1.1.12
ICMP TTL:128 TOS:0x0 ID:301 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:2048 ECHO
=====
10/27-22:52:18.887402 10.1.1.12 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:42028 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:2048 ECHO REPLY

```

Figura 13 - Log de Monitoramento Snort 3

Na figura 14 abaixo é a tela do atacante que está com o IP 10.1.1.10 verificando a conectividade com envio de pacotes para o IP 10.1.1.1; logo em seguida, o IP 10.1.1.1 respondendo os quatro pacotes enviados para IP 10.1.1.10.



```

C:\WINDOWS\system32\cmd.exe
Resposta de 10.1.1.12: bytes=32 tempo=2ms TTL=64
Resposta de 10.1.1.12: bytes=32 tempo=2ms TTL=64
Resposta de 10.1.1.12: bytes=32 tempo=1ms TTL=64
Resposta de 10.1.1.12: bytes=32 tempo<1ms TTL=64

Estatísticas do Ping para 10.1.1.12:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 2ms, Média = 1ms

C:\Documents and Settings\MDPC>ping 10.1.1.1

Disparando contra 10.1.1.1 com 32 bytes de dados:

Resposta de 10.1.1.1: bytes=32 tempo=2ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo=8ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo=1ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo<1ms TTL=64

Estatísticas do Ping para 10.1.1.1:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 8ms, Média = 2ms

C:\Documents and Settings\MDPC>

```

Figura 14 - Tela do Atacante IP 10.1.1.10

Na Figura 15 abaixo é o *log* de captura de pacotes da rede. O atacante está com IP 10.1.1.12 origem e o IP 10.1.1.10 sendo o alvo destino. Temos também as informações do protocolo *ICMP*; *TTL*, que estabelece o tempo de vida do pacote, através das métricas de números de saltos e tempo, etc. Já *Echo* seria a pergunta ou envio do IP 10.1.1.10 para o IP 10.1.1.12; o *Echo Reply* é a resposta do IP 10.1.1.12 para o IP 10.1.1.10. Os códigos "61 62 63..." logo após *ECHO*, são as áreas de dados do pacote *ICMP*, podem ser usados pelo *checksum* para checagem do pacote, ou seja, saber se o pacote chegou no destino íntegro, usado principalmente para enviar uma mensagem de acordo com o tipo "*Type*" e código "*Code*", gerado no cabeça-rio. Esses dados são gerados aleatoriamente na área de carga do pacote (*payload*), de acordo com o tipo e código.

```

Debian.Server [Executando] - Oracle VM VirtualBox
Máquina Visualizar Dispositivos Ajuda (H)

10/27-22:57:48.231660 10.1.1.10 -> 10.1.1.1
ICMP TTL:128 TOS:0x0 ID:313 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:4864 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi

=====

10/27-22:57:48.231785 10.1.1.1 -> 10.1.1.10
ICMP TTL:64 TOS:0x0 ID:32395 IpLen:20 DgmLen:60
Type:0 Code:0 ID:512 Seq:4864 ECHO REPLY
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi

=====

10/27-22:57:49.237169 10.1.1.10 -> 10.1.1.1
ICMP TTL:128 TOS:0x0 ID:314 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:5120 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi

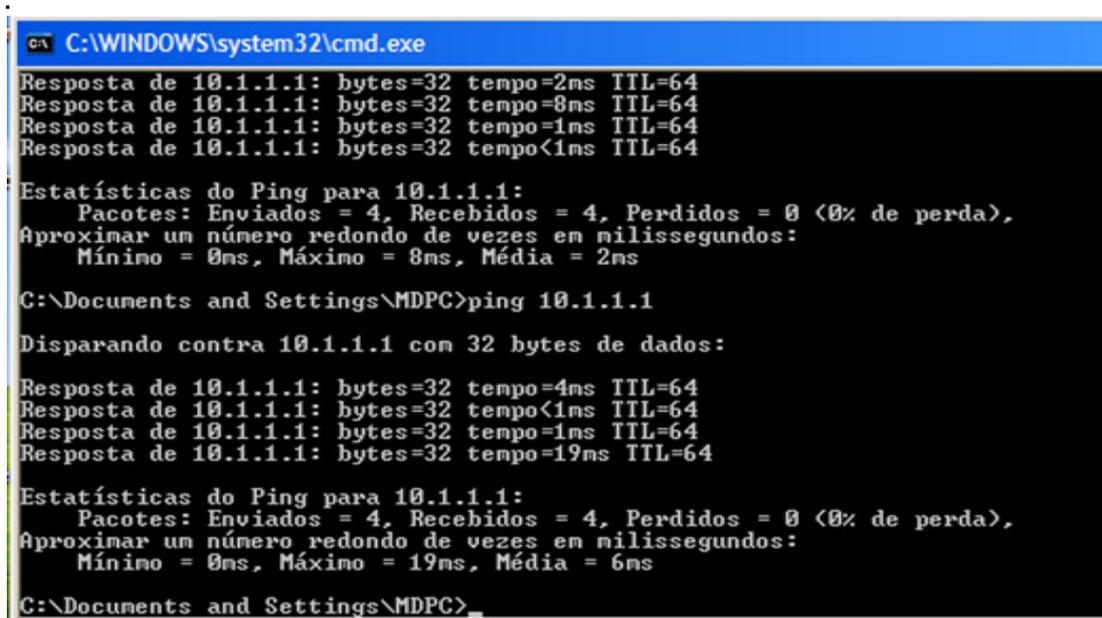
=====

10/27-22:57:49.237243 10.1.1.1 -> 10.1.1.10

```

Figura 15 - Log de Monitoramento *Snort* 4

Na figura 16 abaixo é a mesma situação da Figura 14, só que o atacante está com o IP 10.1.1.10 verificando a conectividade, com envio de pacotes para o IP 10.1.1.1; logo em seguida, o IP 10.1.1.1 respondendo os quatro pacotes enviados para IP 10.1.1.10.



```

C:\WINDOWS\system32\cmd.exe
Resposta de 10.1.1.1: bytes=32 tempo=2ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo=8ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo=1ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo<1ms TTL=64

Estatísticas do Ping para 10.1.1.1:
  Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
  Mínimo = 0ms, Máximo = 8ms, Média = 2ms

C:\Documents and Settings\MDPC>ping 10.1.1.1

Disparando contra 10.1.1.1 com 32 bytes de dados:

Resposta de 10.1.1.1: bytes=32 tempo=4ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo<1ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo=1ms TTL=64
Resposta de 10.1.1.1: bytes=32 tempo=19ms TTL=64

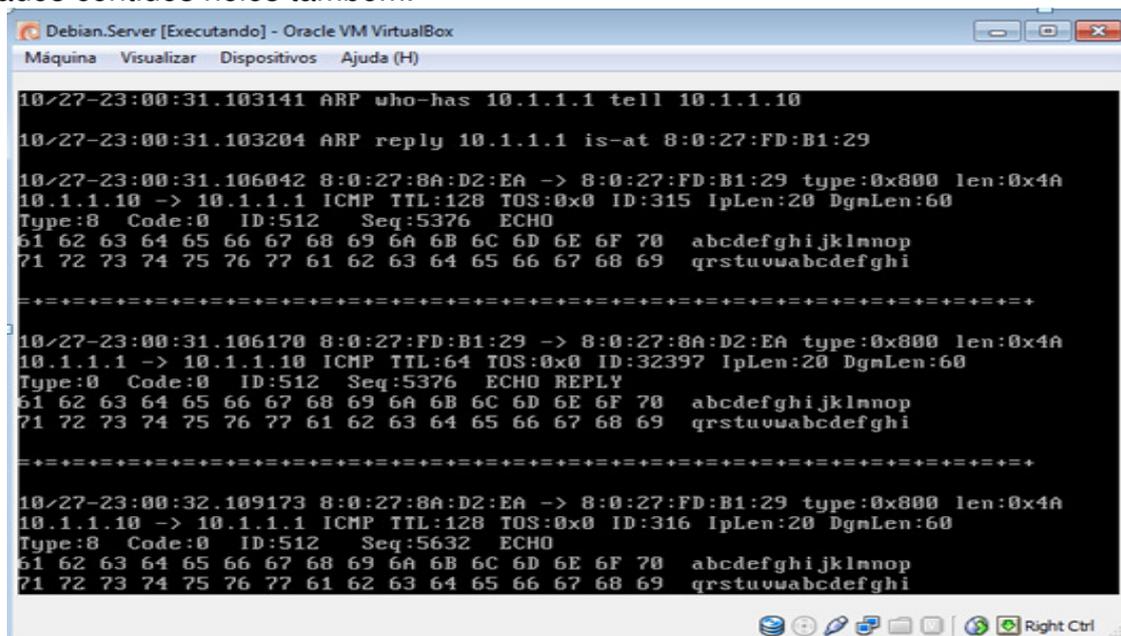
Estatísticas do Ping para 10.1.1.1:
  Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
  Mínimo = 0ms, Máximo = 19ms, Média = 6ms

C:\Documents and Settings\MDPC>

```

Figura 16 - Tela do Atacante IP 10.1.1.10

A Figura 17 abaixo está mostrando o monitoramento de pacotes que trafegou na rede. O comando utilizado para captura "*snort -dve*", mostra todos os cabeçalhos e os dados contidos neles também.



```

Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)

10/27-23:00:31.103141 ARP who-has 10.1.1.1 tell 10.1.1.10
10/27-23:00:31.103204 ARP reply 10.1.1.1 is-at 8:0:27:FD:B1:29

10/27-23:00:31.106042 8:0:27:8A:D2:EA -> 8:0:27:FD:B1:29 type:0x800 len:0x4A
10.1.1.10 -> 10.1.1.1 ICMP TTL:128 TOS:0x0 ID:315 IpLen:20 DgnLen:60
Type:8 Code:0 ID:512 Seq:5376 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi

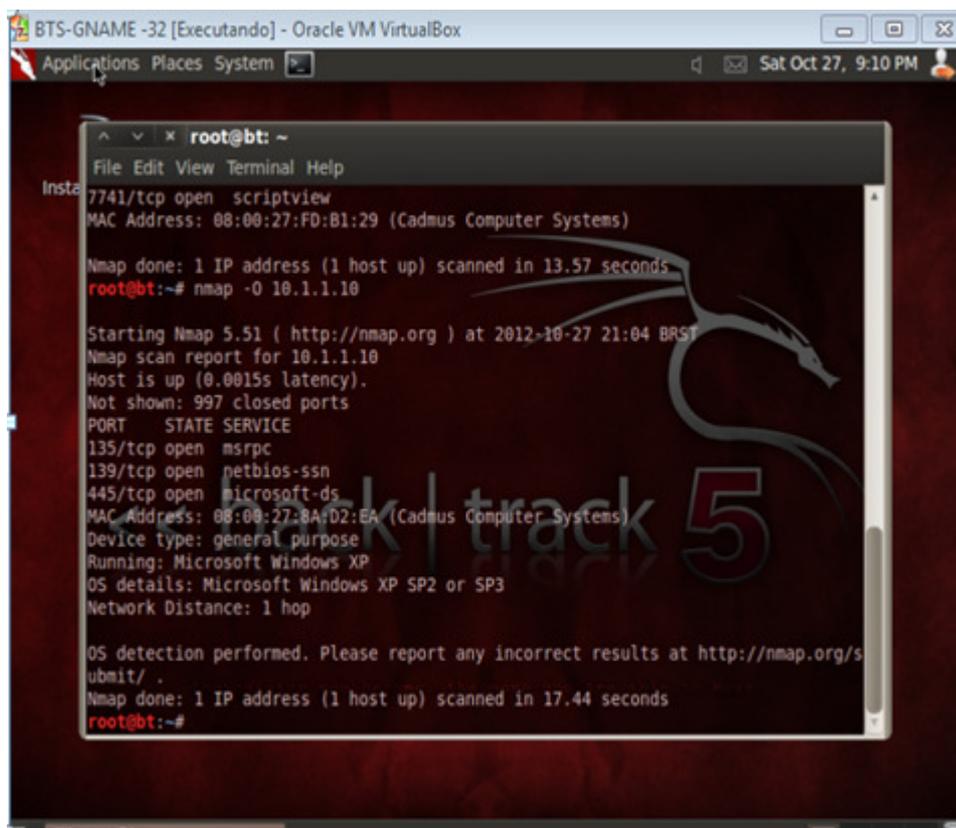
=====
10/27-23:00:31.106170 8:0:27:FD:B1:29 -> 8:0:27:8A:D2:EA type:0x800 len:0x4A
10.1.1.1 -> 10.1.1.10 ICMP TTL:64 TOS:0x0 ID:32397 IpLen:20 DgnLen:60
Type:0 Code:0 ID:512 Seq:5376 ECHO REPLY
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi

=====
10/27-23:00:32.109173 8:0:27:8A:D2:EA -> 8:0:27:FD:B1:29 type:0x800 len:0x4A
10.1.1.10 -> 10.1.1.1 ICMP TTL:128 TOS:0x0 ID:316 IpLen:20 DgnLen:60
Type:8 Code:0 ID:512 Seq:5632 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwabcdefghi

```

Figura 17 - Log de Monitoramento *Snort* 5

Na figura 18, é o atacante do IP 10.1.1.12 que está utilizando o *nmap*: Ferramenta de exploração de Rede e Rastreamento de Segurança / Portas, para a captura de informação da vítima do IP 10.1.1.10. O comando "*nmap -O*" ativa a identificação do host remoto via TCP/IP, apresenta versão do sistema operacional e tempo ativo.



```
BTS-GNAME -32 [Executando] - Oracle VM VirtualBox
Applications Places System Sat Oct 27, 9:10 PM

root@bt: ~
File Edit View Terminal Help
7741/tcp open  scriptview
MAC Address: 08:00:27:FD:B1:29 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 13.57 seconds
root@bt:~# nmap -O 10.1.1.10

Starting Nmap 5.51 ( http://nmap.org ) at 2012-10-27 21:04 BRST
Nmap scan report for 10.1.1.10
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 08:00:27:BA:D2:EA (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.44 seconds
root@bt:~#
```

Figura 18 - Tela do *BackTrack* realizando *nmap*.

Na figura 19 mostra o *snort* que capturou o tráfego de dados na rede gerado do *nmap* máquina atacante com destino à máquina da vítima.

```

10/27-23:05:29.977595 8:0:27:A0:89:B2 -> 8:0:27:8A:D2:EA type:0x800 len:0x4A
10.1.1.12:63561 -> 10.1.1.10:1 TCP TTL:37 TOS:0x0 ID:14566 IpLen:20 DgnLen:60
***A*** Seq: 0xCB1D32BF Ack: 0xAB05A97 Win: 0x8000 TcpLen: 40
TCP Options (5) => WS: 10 NOP MSS: 265 TS: 4294967295 0 SackOK

10/27-23:05:29.977869 8:0:27:8A:D2:EA -> 8:0:27:A0:89:B2 type:0x800 len:0x3C
10.1.1.10:1 -> 10.1.1.12:63561 TCP TTL:128 TOS:0x0 ID:1334 IpLen:20 DgnLen:40
****R** Seq: 0xAB05A97 Ack: 0xAB05A97 Win: 0x0 TcpLen: 20

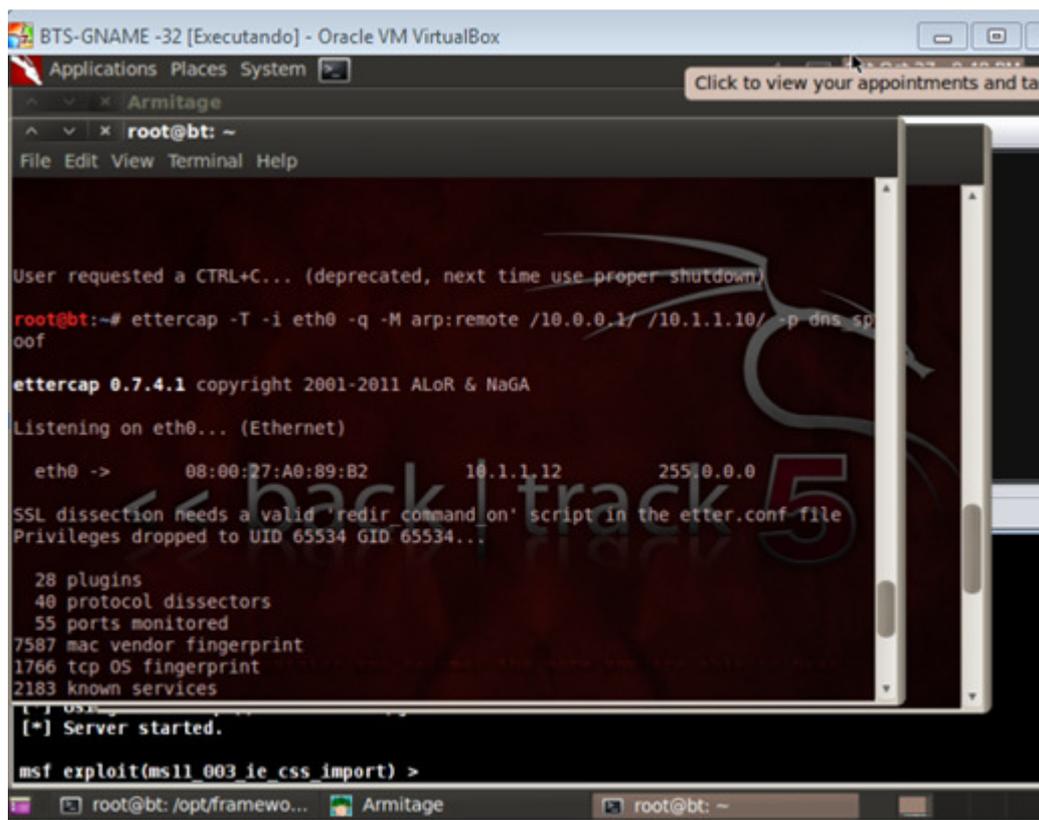
10/27-23:05:30.004117 8:0:27:A0:89:B2 -> 8:0:27:8A:D2:EA type:0x800 len:0x4A
10.1.1.12:63562 -> 10.1.1.10:1 TCP TTL:48 TOS:0x0 ID:58984 IpLen:20 DgnLen:60
**U*P**F Seq: 0xCB1D32BF Ack: 0xAB05A97 Win: 0xFFFF TcpLen: 40 UrgPtr: 0x0
TCP Options (5) => WS: 15 NOP MSS: 265 TS: 4294967295 0 SackOK

10/27-23:05:30.004195 8:0:27:8A:D2:EA -> 8:0:27:A0:89:B2 type:0x800 len:0x3C
10.1.1.10:1 -> 10.1.1.12:63562 TCP TTL:128 TOS:0x0 ID:1335 IpLen:20 DgnLen:40
***A*R** Seq: 0x0 Ack: 0xCB1D32C0 Win: 0x0 TcpLen: 20

```

Figura 19 - Tela de Monitoramento *Snort* 6

Conforme a Figura 20 O *ettercap* é um *sniffer* de rede utilizado para capturar pacotes na rede. Nesse caso, ele está sendo usado também para injetar pacotes. "-T" é a interface de modo texto; "-i" é a referência da interface que vai sair para a rede em modo promíscuo; "-q" fará o *ettercap* ser *superQuiet* (não imprimir pacotes-primas na janela de; terminal); "-M" homem do modo de meio; "*arp:remote*" é um tipo de envenenamento, é um parâmetro para o comando necessário para esse tipo de ataque; "/10.0.0.1/" é o *gateway* da vítima, e o ip, que o *ettercap* vai se passar; "/10.1.1.10" é o ip do Atacante, o *ettercap* injetará pacotes para estes destinos; "-p *dns_spoof*" é o *plug* que o *ettercap* usará para *spoof* da rede da vítima.



```

BTS-GNAME -32 [Executando] - Oracle VM VirtualBox
Applications Places System
Armitage
root@bt: ~
File Edit View Terminal Help

User requested a CTRL+C... (deprecated, next time use proper shutdown)
root@bt:~# ettercap -T -i eth0 -q -M arp:remote /10.0.0.1/ /10.1.1.10/ -p dns_spoof
ettercap 0.7.4.1 copyright 2001-2011 ALOR & NaGA

Listening on eth0... (Ethernet)

eth0 ->      08:00:27:A0:89:B2      10.1.1.12      255.0.0.0
SSL dissection needs a valid 'redir command' script in the etter.conf file
Privileges dropped to UID 65534 GID 65534...

 28 plugins
 40 protocol dissectors
 55 ports monitored
7587 mac vendor fingerprint
1766 tcp OS fingerprint
2183 known services

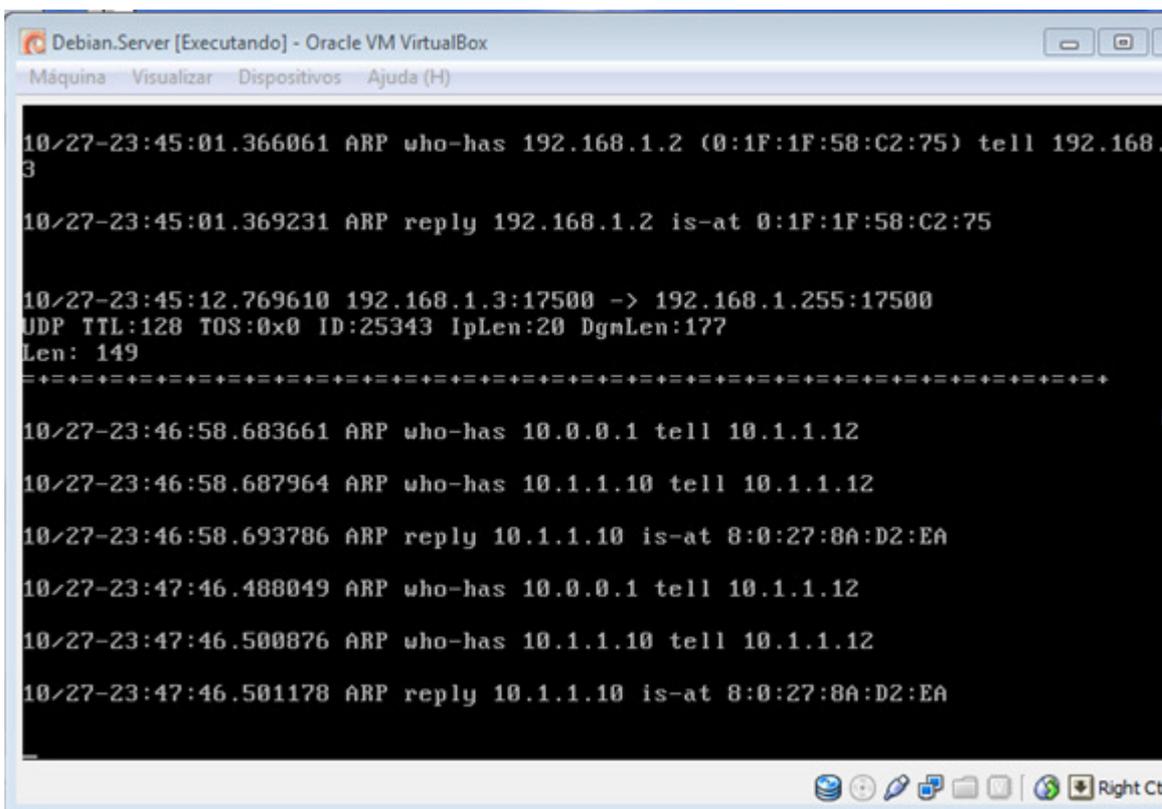
[*] Server started.

msf exploit(ms11_003_ie_css_import) >
root@bt: /opt/framewo...  Armitage  root@bt: ~

```

Figura 20 - Tela do *BackTrack* realizando *ettercap*.

A figura 21 demonstra o monitoramento de tráfego de pacote na rede que o atacante do IP 10.1.1.12 está *spoof*. Assim sendo, utilizando esta técnica, o atacante pode direcionar o email ou navegador da vítima para o seu próprio servidor. Com isso, ele pode, por exemplo, capturar senhas de cartões de crédito em sites de compras. O protocolo *ARP* -Protocolo de Resolução de Endereço, ele é responsável por localizar o endereço de *hardware* de um dispositivo a partir de seu endereço *ip* conhecido.



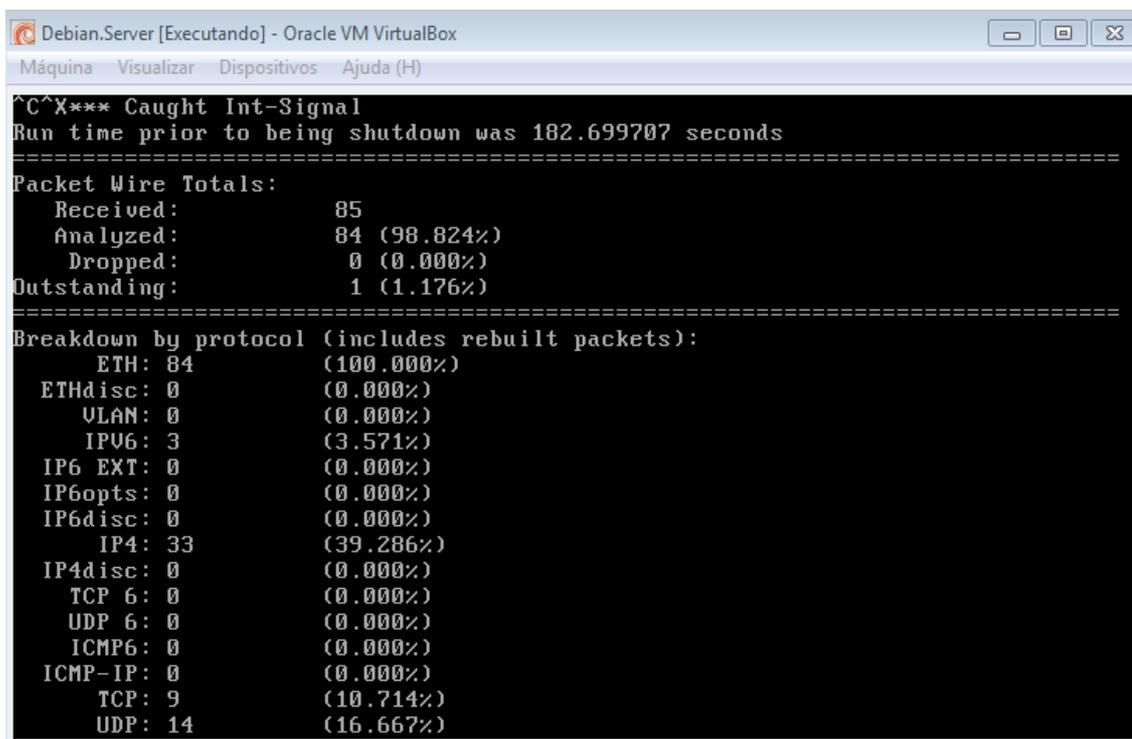
```
Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)

10/27-23:45:01.366061 ARP who-has 192.168.1.2 (0:1F:1F:58:C2:75) tell 192.168.
3
10/27-23:45:01.369231 ARP reply 192.168.1.2 is-at 0:1F:1F:58:C2:75

10/27-23:45:12.769610 192.168.1.3:17500 -> 192.168.1.255:17500
UDP TTL:128 TOS:0x0 ID:25343 IpLen:20 DgnLen:177
Len: 149
=====
10/27-23:46:58.683661 ARP who-has 10.0.0.1 tell 10.1.1.12
10/27-23:46:58.687964 ARP who-has 10.1.1.10 tell 10.1.1.12
10/27-23:46:58.693786 ARP reply 10.1.1.10 is-at 8:0:27:8A:D2:EA
10/27-23:47:46.488049 ARP who-has 10.0.0.1 tell 10.1.1.12
10/27-23:47:46.500876 ARP who-has 10.1.1.10 tell 10.1.1.12
10/27-23:47:46.501178 ARP reply 10.1.1.10 is-at 8:0:27:8A:D2:EA
```

Figura 21 - Tela de Monitoramento *Snort 7*

Utilizado a opção de captura de dados, o *Snort* contém os *logs* de monitoramento dos protocolos que trafegou na rede. Conforme a figura 22, temos captura total de pacotes dentro do tempo de execução. Dos 100% de pacotes recebidos foram analisados 98,824%, caiu 0,0% e marcante 1,176%. Temos também o percentual de captura de pacotes trafegado na rede por protocolo.

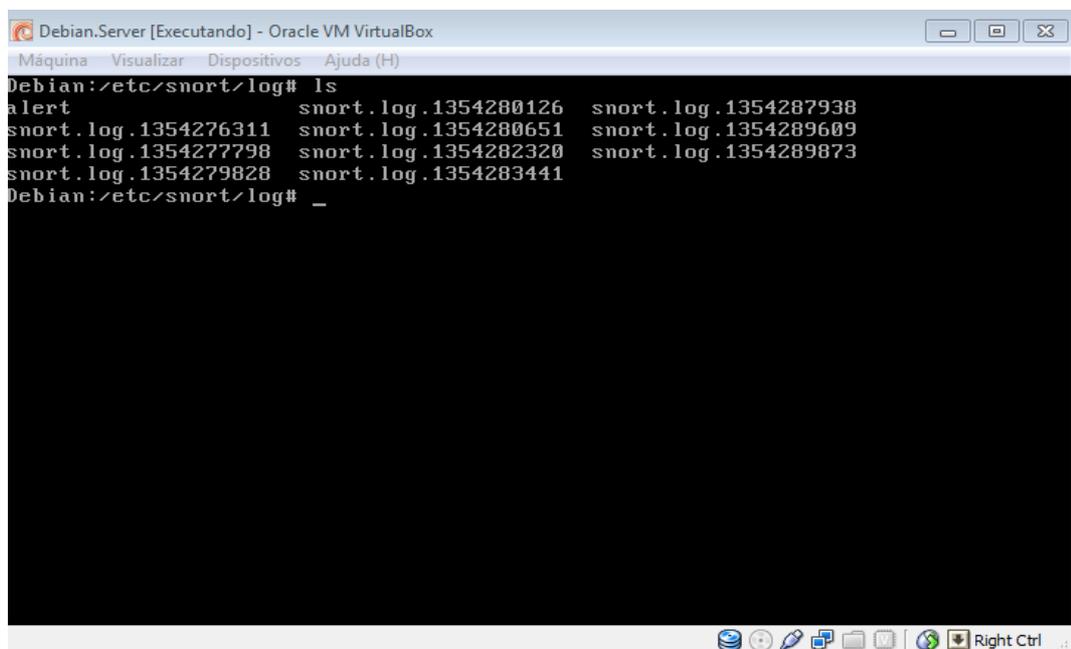


```
Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)

^C^X*** Caught Int-Signal
Run time prior to being shutdown was 182.699707 seconds
=====
Packet Wire Totals:
  Received:      85
  Analyzed:      84 (98.824%)
  Dropped:       0 (0.000%)
  Outstanding:   1 (1.176%)
=====
Breakdown by protocol (includes rebuilt packets):
  ETH: 84 (100.000%)
  ETHdisc: 0 (0.000%)
  ULAN: 0 (0.000%)
  IPV6: 3 (3.571%)
  IP6 EXT: 0 (0.000%)
  IP6opts: 0 (0.000%)
  IP6disc: 0 (0.000%)
  IP4: 33 (39.286%)
  IP4disc: 0 (0.000%)
  TCP 6: 0 (0.000%)
  UDP 6: 0 (0.000%)
  ICMP6: 0 (0.000%)
  ICMP-IP: 0 (0.000%)
  TCP: 9 (10.714%)
  UDP: 14 (16.667%)
```

Figura 22 - Tráfego de Protocolos

Conforme a figura 23, são apresentados os logs de monitoramento que foram salvos no diretório do Snort. Para monitorar os pacotes da rede e salvar, deve ser executado o seguinte comando: "*snort -l /etc/snort/log*"; isso, depois de criado a pasta Log no diretório do snort. Passando para a leitura dos logs salvos o comando é: "*snort -dvr /etc/snort/log/snort.log.1354209873*".



```
Debian.Server [Executando] - Oracle VM VirtualBox
Máquina  Visualizar  Dispositivos  Ajuda (H)
Debian:/etc/snort/log# ls
alert          snort.log.1354280126  snort.log.1354287938
snort.log.1354276311  snort.log.1354280651  snort.log.1354289609
snort.log.1354277798  snort.log.1354282320  snort.log.1354289873
snort.log.1354279828  snort.log.1354283441
Debian:/etc/snort/log# _
```

Figura 23 - Logs de monitoramentos de pacote de rede salvos

6. Conclusão

No atual ambiente competitivo, a informações são muito importante em qualquer gestão organizacional, por serem características indispensáveis nos contextos internos e externos das empresas.

O Sistema de Detecção de Intrusos possui uma grande importância na segurança de redes. No entanto, o sistema de detecção não é totalmente seguro; porém, cada vez mais, deve-se investir nas técnicas do IDS para tornar difíceis as invasões dos intrusos.

Com o avanço da área de segurança da informação, o monitoramento IDS é muito importante para o tráfego de dados das redes. O Snort é uma ferramenta excelente para monitorar os ataques, desde de que tenha regras de monitoramento criadas com qualidade. Desse modo, controlando todo o tráfego de entrada e saída da rede.

A cada dia que passa novas técnicas de invasões do sistema crescem, sendo muito importante implementações de boas políticas do IDS, pois, com o surgimento de novas assinaturas, deve-se buscar atualizações constantemente. Para prevenção das futuras invasões, deverão ser gerados os históricos dos relatórios.

7. Trabalhos Futuros

Com base na pesquisa desenvolvida, várias vertentes para futuros trabalhos podem ser identificadas, Como possíveis trabalhos futuros, pode-se apontar:

- Pesquisa para criar regras específicas para alguma segurança onde monitorará o tráfego da rede com mais detalhes;
- Controle dos tráfegos de dados na rede em modo gráfico para melhor análise.

REFERÊNCIAS BIBLIOGRÁFICAS

Alves Dias, Rômulo. **Um modelo de Atualização Automática do Mecanismo de Detecção de Ataques de Rede para Sistema de Detecção de Intrusão.** Disponível em: <http://www.tedebc.ufma.br//tde_busca/arquivo.php?codArquivo=238> Acesso em: 05/04/2012.

Aragão, Francisco. **Metasploit.** Disponível em: <<http://pplware.sapo.pt/internet/metasploit-sabe-o-que-e/>>. Acesso em: 01/10/2012.

Cavallari Militelli, Leonardo. **Proposta de um Agente de Aplicação para Detecção, Prevenção e Contenção de Ataques em Ambientes Computacionais.** Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3142/tde-13032007-152557/pt-br.php>> Acesso em: 05/04/2012.

Costa Claudino Silva, Emanuel. **Gerenciamento e Integração das Bases de Dados de Sistemas de Detecção de Intrusão.** Disponível em: <http://www.tedebc.ufma.br//tde_busca/arquivo.php?codArquivo=51> Acesso em: 05/04/2012.

ICANN. **Lista de Registros de Domínios de Primeiro nível.** Disponível em: <<http://www.icann.org/registries/listing.html>>. Acesso em: 01/10/2012.

Laureano, Marcos Aurelio Pchek. **Gestão de Segurança da informação.** Disponível em: <http://www.mlaureano.org/aulas_material/gst/apostila_versao_20.pdf>. Acesso em: 16/08/2012.

Luiz Cardoso Buriti, George. **Extensões de Segurança para DNS.** Disponível em: <http://www.projotoderedes.com.br/apostilas/apostilas_seguranca.php>. Acesso em: 30/09/2012.

Medrado, Sergio Augusto; Pereira, Júlio César. **Tecnologias (IDS) SNORT X OSSEC.** Disponível em: [ftp://fipp.unoeste.br/artigos/artigos-ids/< Sergio-Medrado.pdf](ftp://fipp.unoeste.br/artigos/artigos-ids/<Sergio-Medrado.pdf)>. Acesso em: 10/06/2012.

Nogueira, André. **Nmap.** Disponível em: <<http://pplware.sapo.pt/windows/software/nmap-sabe-o-que-e-e-para-que-serve/>>. Acesso em: 05/10/2012.

Paula, Marco Antonio.; Montoro, Rodrigo. **Snort Rules.** Disponível em: <http://www.staysafepodcast.com.br/wp-content/uploads/2010/07/RevistaStaySafe_1.pdf>. Acesso em: 10/06/2012.

Ribeiro, Jarley. **Spoofing pelo Ettercap.** Disponível em: <<http://www.securityhacker.org/artigos/item/spoofing-pelo-ettercap>>. Acesso em: 05/10/2012.

Roesch, Martin. **Desenvolvedor do Snort e Fundador do Sourcefire** Disponível em: <<http://www.snort.com.br/arquiteturasnort.asp>> Acesso em: 10/04/2012.

Salvadori Viri, Émerson. **Implementação de um IDS Utilizando SNMP e Lógica Difusa**. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/11475>> Acesso em: 05/04/2012.