Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto

SISTEMA DE DETECÇÃO DE INTRUSÃO EM REDES INFORMÁTICAS

JOSÉ CARLOS OLIVEIRA PINTO

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Telecomunicações

2009

Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto

SISTEMA DE DETECÇÃO DE INTRUSÃO EM REDES INFORMÁTICAS

Dissertação Apresentada para Obtenção do Grau de Mestre
em Engenharia Electrotécnica e de Computadores
Área de Especialização de Telecomunicações

Orientação Científica: Eng. Paula Viana, pviana@isep.ipp.pt
Candidato: José Carlos Oliveira Pinto, Nº 1010613, josecpinto@gmail.com
Empresa: ANA Aeroportos, ASC Aeroporto Sá Carneiro
Supervisão: Eng. António Jorge Pinho, ajpinho@ana.pt

2009

Know your enemy and know yourself and you can fight a hundred battles without disaster.

Sun Tzu, The Art of War

Agradecimentos

Expresso aqui os meus agradecimentos sinceros aos meus pais, pelo apoio incansável prestado ao longo deste percurso. Muitos outros familiares e amigos, desde professores, investigadores científicos, colegas do ISEP e colegas da ANA, contribuíram para que esta viagem fosse agradável.

Deixo um agradecimento especial à Eng. Paula Viana e ao Eng. Eduardo Vasconcelos pelo apoio e paciência.

Gostava de também deixar uma referência à colaboração existente entre o ISEP e a ANA, e como tal permito-me individualizar a Eng. Paula Viana e o Eng. António Pinho.

Resumo

Na segurança de redes de informática, a principal ameaça existente são os ataques ou intrusões. Com as actividades não autorizadas a aumentar, as tradicionais técnicas de *firewall* não conseguem implementar a defesa completa contra intrusos.

Um *Intrusion Detection System* (IDS) pode ser a possível resposta a esta lacuna. Neste trabalho foi feito um estudo sobre as várias componentes de um IDS e, também, uma comparação dos IDS existentes tanto no mercado comercial como no de *open source*.

Um IDS foi implementado e testado na rede informática do Aeroporto Sá Carneiro, com algumas das ferramentas abordadas. Nesta tese será dada ênfase à análise das técnicas de detecção assim como aos dados recolhidos, incluindo a necessidade da utilização de um *test access port* (TAP) para melhor compreender e facilitar o solucionamento das alertas obtidas.

Para complementar um sistema de segurança de redes, é essencial que se possibilite ao administrador informação sobre os intrusos. Para esse efeito utilizaram-se os *honeypots*, que criam redes de informática virtuais para servirem de isca. Nos *honeypots* mais desenvolvidos pode se fazer uma monitorização dos intrusos, tanto durante como depois dos ataques, oferecendo valiosa informação sobre os seus métodos. Nesta tese será apresentada também a arquitectura do *honeypot* implementado e dos testes realizados.

Abstract

Attacks on network infrastructure are presently the main threat against networks and information security. Because traditional firewall techniques cannot provide complete protection against rapidly growing unauthorized activities in networks, it has become necessary that an IDS (Intrusion Detection System) be implemented as a component of defense-in-depth.

A survey of the various available components of IDS technology is presented, including a comparison of commercial and open source software.

An IDS sensor was implemented using some of the components studied. The sensor´s functionality was tested before being strategically placed in a highly complicated network of the Sá Carneiro Airport. An emphasis will be made, on the techniques used by the IDS and the analyses of the data collected; this will include the necessary utilization of a test access port (TAP) to facilitate the solving of the obtained alerts.

To complement a network security system, it is essential that the administrator receives as much information as possible about the intruders; honeypots were created for that very reason. Honeypots are virtual networks that serve as decoys, distracting adversaries from more valuable machines on a network and also providing an in-depth examination of the adversaries during the exploitation.

Índice

Índice de Tabelas

Índice de figuras

Acrónimos

ACL – Access Control Lists

ADSL – Asymmetric Digital Subscriber Line

ANA – Aeroportos e Navegação Aérea

ARP – Address Resolution Protocol

ASC – Aeroporto Sá Carneiro

CCMP – Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

CiFS – Common internet File System

CUI – Console User Interface

DASC – Direcção do Aeroporto Sá Carneiro

DDoS – Denial of Service

DMZ – Demilitarized Zone

DoS – Distributed Denial of Service

EMEA – Europe Middle East and Africa

GUI – Graphical User Interface

HIDS – Host Intrusion Detection System

HTTP – Hypertext Transfer Protocol

IDS – Intrusion Detection System

LAN – Local Aerea Network

MAC – Media Access Control

MPLS – Multi Protocol Label Switching

NASL – Nessus Attack Scripting Language

NIDS – Network Intrusion Detection System

NIC – Network Interface Card

NSM – Network Security Management

P2P – Peer - to - Peer

PVID – Port VLAN Identification

SO – Sistema Operativo

SIFACT – Serviços de Facturação

TAP – Test Access Port

TKIP – Temporal Key Integrity Protocol

VLAN – Virtual Local Area Network

VoIP – Voice over Internet Protocol

WAN – Wide Area Networks

WAP – Wireless Application Protocol

WEP – Wired Equivalent Privacy

1. Introdução

Com o aumento exponencial do número de redes informáticas, tem-se verificado um acréscimo dos ataques. Todas as redes de informáticas podem ser alvo de ataques, desde as mais pequenas, como as redes domésticas até às mais vastas e complexas, como as redes das empresas.

Para piorar ainda mais a situação pode facilmente obter-se programas, através da Internet (ou por outros meios), e depois utilizá-los para acções mal intencionadas, fazendo com que qualquer pessoa inexperiente tenha as mesmas capacidades de um profissional.

Surge então a questão: como se pode proteger uma rede de informática? A solução pode passar pela implementação de um *Intrusion Detection System* (IDS). Um IDS inspecciona o tráfego de rede, identifica os pacotes que podem comprometer a rede, e depois alerta o administrador de redes informática. Um IDS mais evoluído pode fazer um estudo mais intensivo do ataque e, assim, descobrir entre outros aspectos, o conteúdo do pacote, a sua origem e destino e as consequências do ataque no sistema.

1.1 Motivação

As redes de informática foram para mim sempre um assunto de interesse, e este interesse veio a fortalecer-se com o conhecimento da existência dos *crackers* (pessoas com intenções maliciosas, também conhecidos como *black hats*).

Sendo assim, ao ser-me apresentada uma oportunidade de estagiar numa empresa com redes complexas, podendo melhorar e aplicar os meus conhecimentos na segurança de redes informáticas, não hesitei.

Este estágio não só me permitiu tornar-me num *white hat hacker* (pessoas que exploram e detectam erros de concepção das redes), mas deu-me também a possibilidade de obter conhecimentos práticos, numa empresa com uma rede de dimensão considerável e bastante apetecível aos intrusos.

1.2 Objectivos

Pretendeu-se com este estágio melhorar o sistema de segurança IDS da rede informática do Aeroporto Sá Carneiro (ASC) e, para tal, foi necessário:

- Efectuar um *upgrade* ao *software* do actual IDS; isto incluiu o servidor e os sensores (*server* + *probes* remotas);
- Integrar a monitorização sem fios na solução de IDS já existente;
- Apresentar propostas de monitorização para a estrutura de rede do ASC (tanto a cablada como a de rede sem fios);
- Efectuar uma actualização do *software* de geração de ataques (Nessus);

- Simular ataques e confirmar a sua correcta detecção pelo servidor do IDS;
- Propor a correcção de vulnerabilidades pelo servidor e fazer a aplicação efectiva das mesmas nos sistemas.

1.3 Apresentação do local de estágio

O estágio foi realizado na ANA, Aeroportos de Portugal, empresa que é responsável pela gestão dos aeroportos nacionais. Foi desenvolvido na delegação do Aeroporto Francisco Sá Carneiro no departamento de informática.

1.4 Calendarização

Sendo que o objectivo principal deste trabalho foi o estudo e implementação do IDS, a tarefa foi dividida em cinco partes:

- Investigação acerca do funcionamento de um IDS e do *honeypot*, assim como a escolha do melhor *software* para a implementação eficaz e robusta dos mesmos;
- Utilizaçao das ferramentas necessárias para testar a segurança das redes;
- Implementação do IDS que inclui a instalação e o teste do *software* utilizado para a tarefa e, posteriormente, a sua colocação numa rede para a obtenção de resultados;
- Estudo das alertas obtidas para poder aplicar a correcção necessária tanto nas máquinas como nos servidores;
- Escrita da tese.

ID	Task Name	Início
1	**Tese: Sistema de Detecção de Intrusão de**	**01-11-07**
2	Investigação da Teorico	01-11-07
3	**Investigação Experimental**	**01-04-08**
4	**Computador de teste**	**01-04-08**
7	**Upgrade do software de IDS**	**15-04-08**
9	**Integração da monitorização do**	**25-04-08**
11	**Propostas de monitorização da**	**09-05-08**
13	**Upgrade do software Nessus**	**14-05-08**
15	**Simulação de ataques e sua**	**28-05-08**
17	**Proposta de correcção de**	**17-06-08**
19	**Estudo de soluções na**	**20-06-08**
21	**Honeypots**	**04-07-08**
23	**Implementação do Sistema de**	**15-12-08**
24	Implementação do IDS com sonda	15-12-08
25	Correcção das vunerabilidades	02-01-09
26	Implementação do Honeypot	19-02-09
27	Escrita da Tese	20-02-09

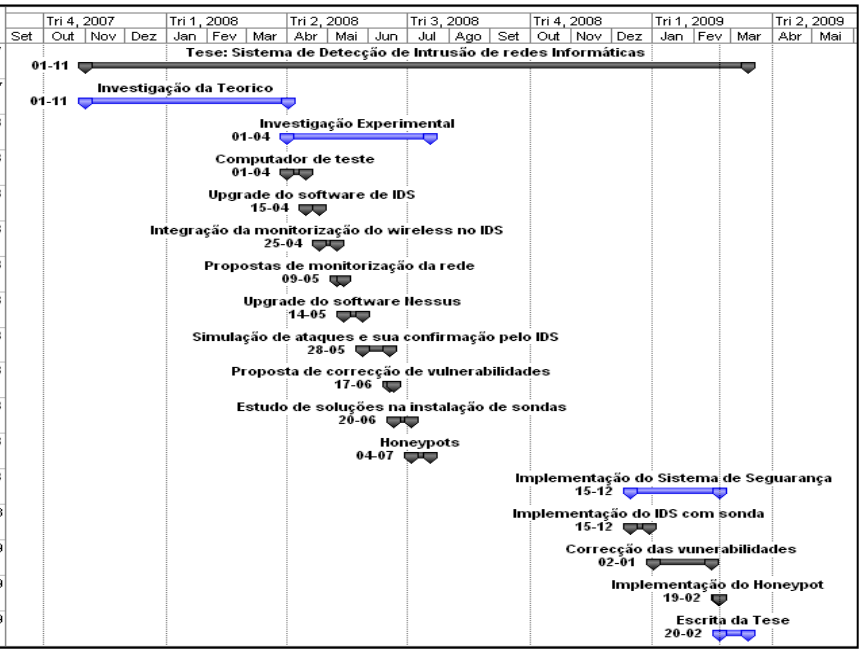

Tabela 1. 1 - Calendarização da Tese

1.5 Organização da Tese

Esta tese está dividida em sete capítulos:

- Capítulo 1 - Apresentação do trabalho, dos motivos pessoais da sua escolha, local da realização do mesmo, sua calendarização e organização da tese;
- Capítulo 2 - Apresenta uma parte teórica sobre um IDS, e as razões para a necessidade de implementar uma boa defesa contra intrusos. Apresenta também alguns tipos de possíveis ataques e as suas consequências e ainda formas possíveis de tornar a actividade dos *hackers* mais difícil, e ao mesmo tempo possibilitar um estudo dos *hackers* com os *honeypots*;
- Capítulo 3 - Comparação dos diferentes tipos de *software* existentes no mercado que podem ser utilizados na implementação de um IDS, tanto os de *open source* como os comerciais;
- Capítulo 4 - Apresentação da rede do ASC, tanto na sua infra-estrutura como também alguns dos equipamentos utilizados;
- Capítulo 5 - Apresenta a implementação de um IDS com as justificações do *software* escolhido. Como também possíveis soluções no endurecimento do IDS, e a implementação de um *honeypot*.
- Capítulo 6 - Apresenta o sistema de segurança implementado para a rede do Aeroporto Sá Carneiro, foram utilizados todos os conhecimentos obtidos durante a fase de teste para poder obter uma melhor defesa;
- Capítulo 7 - Apresenta as conclusões e linhas de orientação para trabalhos futuros.

2. Tipos de Ataques e Técnicas de Segurança

2.1 Introdução

A Internet tem uma enorme influência na sociedade, tendo criado oportunidades e mercados onde antes era impensável. Como em todas as tecnologias, há sempre aspectos negativos, e um deles é o risco de insegurança, que infelizmente nem todos os utilizadores conhecem suficientemente.

Quem estiver atento aos acontecimentos nacionais e internacionais verifica que há notícias perturbadoras. Como exemplo temos na agência *Reuters* de 17 de Março 2008, o título "*Credit card data stolen from supermarket chain"*, ou para quem pensa que só acontece no estrangeiro, veja-se o Correio da Manhã de 4 de Maio 2008 onde se lê: *"Portugal indefeso perante ataques ciberterroristas"*.

Que fazer então diante da praga das intrusões? Primeiro é necessário saber quem são os intrusos.

2.2 Os Intrusos

Exactamente o que é uma intrusão? O dicionário da Porto Editora considera que uma intrusão é um "*acto de se introduzir sem direito ou usando astúcia*". Para a informática é basicamente qualquer processo utilizado para obter acesso a zonas não autorizadas ou impedir o acesso a sistemas autorizados. As intrusões variam muito, podendo por exemplo ser ataques às vulnerabilidades, *malware*, *phishing* e *spam*.

2.2.1 Os *Crackers* e os *Hackers*

Os responsáveis pelas intrusões são designados por *crackers* e existem várias categorias, tendo cada um os seus motivos e estilos próprios de actuar [1] [2]:

- *Black hat/Crackers* - conhecidos também por *dark side crackers,* são especializados em invasões maliciosas. As suas intenções têm como objectivo conseguir sempre algo lucrativo, desde a obtenção de informação para uso posterior, como exemplo, o equipamento de uma rede ou duma empresa, ou por interesses económicos. São pessoas que fazem sempre uma pesquisa intensiva

sobre o alvo, antes de executar qualquer tipo de intrusão e têm conhecimentos elevados de informática [1].

- *Grey hat* - são *crackers* que nem sempre têm intenções maliciosas, gostam de atacar apenas pelo prazer. Este tipo de *cracker* tem uma ética diferente dos *black hats* pois consideram que é aceitável atacar sistemas desde que não haja nenhum roubo, vandalismo ou violação da confidencialidade do alvo. São pessoas também com um elevado conhecimento de informática [2].
- *Script kiddies* - são *crackers* inexperientes, normalmente pessoas jovens e com poucos conhecimentos de programação. Utilizam as ferramentas disponíveis na Internet (por vezes cedidas por *black hats*) para ganhar fama entre a comunidade de *crackers*. São pessoas curiosas e aventureiros no mundo virtual, que simplesmente querem descobrir mais sobre a Internet e divertir-se. Grande parte dos seus ataques provocam alterações às páginas Web (*defacement*) e podem por vezes até conseguir obter informação importante, que depois utilizam ou vendem a *crackers* mais experientes [2].

Existe também uma outra categoria de pessoas designadas por *White hats* ou *hackers*, que são especialistas em segurança de redes. Utilizam as mesmas ferramentas dos *crackers* para verificar onde existem os pontos mais vulneráveis de uma rede, para melhor os defender e proteger.

2.2.2 O Método Comum de Ataque

Existem vários métodos ao dispor de um *cracker* para utilizar na exploração de um sistema. O método mais generalizado contempla os seguintes passos [1] [3]:

- Reconhecimento passivo - para poder explorar um sistema, o *cracker* necessita de ter alguma informação básica sobre o mesmo (sua localização na rede, a quem pertence e, se possível, algo sobre os equipamentos utilizados na rede). A recolha de informação passiva nem sempre parece ser de muita utilidade mas é necessária e pode posteriormente ajudar a ultrapassar algumas barreiras.
- Reconhecimento activo (*scanning*) - nesta etapa o *cracker* já obteve informação suficiente para realizar um reconhecimento activo. Para agora se poder colocar numa posição mais estratégica necessita de saber quais os *hosts* acessíveis, os sistemas operativos em utilização, os serviços activos e as portas que estão abertas.
- Exploração do sistema -

  Sistemas operativos – os sistemas operativos instalados por defeito trazem muitos serviços activos. A política habitual das empresas que constroem os sistemas operativos é a de facilitar a utilização mesmo que isso implique graves erros na segurança.

Havendo muitos serviços activos, haverá um grande número de portas abertas, o que facilita a entrada do *cracker*.

Ataques ao nível da aplicação – na construção das aplicações os programadores nem sempre dão a devida importância à segurança devido aos prazos a cumprir e às diversas funcionalidades das aplicações. A existência de alguns erros pode comprometer a segurança.

Ataques de Script – este tipo de ataque aparece muito em desenvolvimento de aplicações Web. Algumas versões iniciais de *browsers* eram acompanhadas de *scripts* que tinham vulnerabilidades. Existem aplicações Web que põem em risco a segurança, podendo ser utilizadas pelo próprio *cracker*.

Ataques de má configuração – num sistema com muitos programas, é necessário ter maior cuidado, não só por ter mais portas abertas mas também por ser necessário ter todos os programas actualizados. Os programas mal configurados e não actualizados são um grave risco para a segurança.

Aumento de permissões – o objectivo do *cracker* é obter o controlo total do sistema, necessitando para isso de obter um acesso como *root* ou administrador.

Denial of service – este ataque consegue negar a utilizadores legítimos o acesso a informação. Pode ser através de bloqueio de utilizadores ou ao acesso a uma página Web.

- *Upload* de programas - depois de um *cracker* obter acesso ao sistema normalmente faz *upload* de programas, que lhe vão permitir que tenha mais fácil acesso ao sistema e também comprometer a segurança de outros sistemas da rede.
- *Download* dos dados - se o objectivo do *cracker* é, digamos, espionagem de uma empresa, procurará fazer um *download* dos seus dados mais importantes como, por exemplo, resultados de investigação e desenvolvimentos de produtos novos, listagem de clientes com os seus dados, etc. É necessário ter em mente que se este ataque não for detectado antes ou durante o *download*, não haverá mais possibilidade de o parar, sendo que o restante ataque é feito *off-line.*
- Manter acesso - depois de o *cracker* ter conseguido o acesso, não vai querer passar pelo mesmo processo cada vez que decide utilizar o sistema, por isso coloca um *backdoor* permitindo um acesso cada vez que o desejar. Um *backdoor* pode ser tão simples de acrescentar como adicionar uma "conta" num sistema. A probabilidade de ser detectado diminui se for adicionada num sistema onde haja muitos utilizadores. Um outro método mais sofisticado é substituir um ficheiro por outro parecido, mas com características escondidas.
- Eliminar pistas - depois de ter realizado todo o trabalho, o *cracker* não quer ser descoberto, e por isso elimina todas as pistas da sua intrusão. O método mais

fácil é eliminar a sua entrada nos ficheiros de registo. Outro método é desactivar o registo logo que o *cracker* acede ao sistema.

2.3 Tipos de Intrusões

Existem vários métodos que podem ser utilizados para explorar um sistema. O uso da Internet, permite acesso a uma rede remota. Pode também obter-se acesso a uma rede interna, como uma LAN, através dos sistemas da empresa. Este método é habitualmente utilizado por pessoas de confiança da mesma. Alguns utilizadores de terminais, para facilitar o seu próprio acesso à rede, colocam as palavras-chave em lugares bem visíveis e acessíveis. Existem também meios de exploração mais agressivos como o roubo de equipamento informático, por exemplo portáteis ou de *pen´s* USB. Dependendo dos objectivos de cada *cracker*, podem ser utilizados um ou vários meios para alcançar o seu objectivo [1] [2].

2.3.1 Através da Internet

Considerando que hoje muitas empresas e particulares têm acesso à Internet, este meio tornou-se o mais utilizado e, para facilitar ainda mais as acções do *cracker*, existem equipamentos que estão sempre ligados à Internet e sem qualquer vigilância. Os métodos de intrusões mais utilizados para explorar sistemas através da *Internet* são:

Intrusões coordenadas

A *Internet* veio facilitar a comunicação entre pessoas de todo mundo, mas também facilitou a união dos *crackers*, especialmente para planear intrusões simultâneas a sistemas predefinidos. Exemplos de intrusões coordenadas são o *Denial of Service* (DoS) e o *Distributed Denial of Service* (DDoS).

A intrusão DoS é um ataque à disponibilidade do sistema, podendo inutilizá-lo ou torná-lo lento. Existem dois tipos de ataques DoS: um que conduz à falha do sistema e à necessidade de o reiniciar, e outro que o inunda com pacotes impedindo que ele responda. No primeiro caso o *cracker*, para provocar esta intrusão, inunda o sistema com pedidos e no segundo, envia pacotes até saturar o sistema. Como o sistema não vai ter possibilidade para os resolver todos, começa a perder a capacidade de processamento tornando-se inutilizável. Qualquer sistema que processa pacotes, por exemplo os equipamentos de redes, está vulnerável a este tipo de ataque. Do ponto de vista de um *cracker* este é um ataque fácil, já que com alguns pacotes pode inutilizar um sistema.

Num ataque DoS é habitual usar só uma máquina para lançar o ataque, mas num ataque DDoS são utilizadas várias máquinas ao mesmo tempo, por um grupo de *crackers* que se associaram. Torna-se assim muito difícil defender sistemas contra intrusões deste tipo, porque os ataques vêm de diferentes máquinas com diferentes IP´s. Dificilmente se consegue

bloquear todos os IP´s, havendo sempre a possibilidade de alguns pacotes passarem sem serem detectados pelo IDS.

Tomar uma sessão de assalto

Por vezes torna-se mais fácil conseguir uma intrusão num sistema através de um utilizador autorizado. Este método começa por encontrar uma sessão aberta através de um utilizador autorizado e depois toma a sessão por assalto. Depois de a sessão ser tomada por assalto o *cracker* pode continuar com a sessão aberta durante tempo suficiente para obter mais acessos, obter permissões de *root* ou colocar *backdoors.*

Uma das principais razões para fazer este tipo de ataque é não haver necessidade de o *cracker* se identificar no sistema, sendo que o utilizador já o fez, e enquanto a sessão estiver activa não necessita de o voltar a fazer.

Neste tipo de ataque existem duas opções: o método passivo e o activo. No passivo, depois de o *cracker* tomar a sessão por assalto, limita-se a observar todo o tráfego entre o servidor e o utilizador. O método passivo é muito útil para obter informação importante como, por exemplo, palavras-chave.

No método activo o *cracker* toma a sessão por assalto e envia um DoS ao utilizador, figura 2.1, forçando-o a abandonar a sessão. Pode então obter a informação que necessita para o seu objectivo e até conseguir fazer *uploads*.

Estes tipos de ataques estão mais orientados para as sessões com aplicações que possibilitam executar comandos no sistema, como FTP ou telnet, não havendo muito interesse em sessões mais estáticas como, por exemplo, HTTP.

Embora possível este tipo de ataque é difícil de implementar porque exige que o *cracker* engane o sistema pensando que a máquina do *cracker* é o utilizador e enviando para este o tráfego que se destinava ao utilizador.

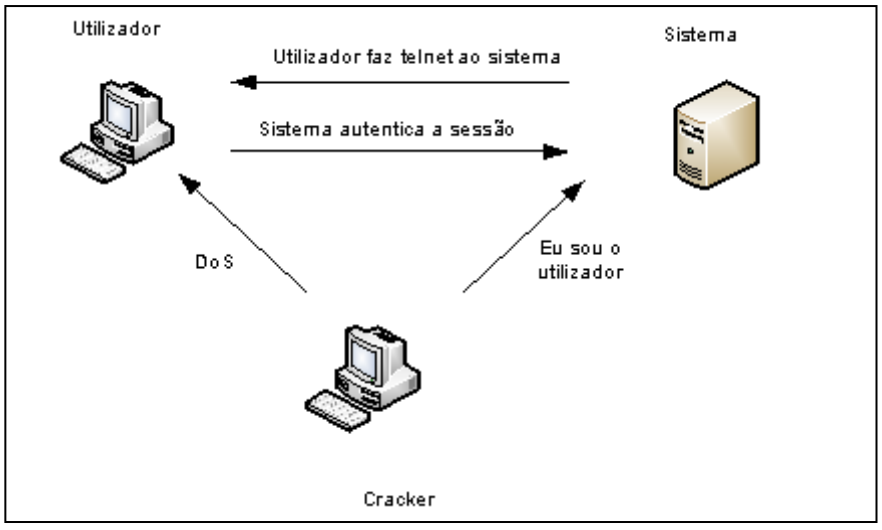

Figura 2. 1 - Processo da sessão tomada por assalto

Spoofing

O método *spoofing* e o assalto a uma sessão têm algumas semelhanças, mas existe uma diferença importante que convém sublinhar. No método de assalto a uma sessão o *cracker* necessita que o utilizador tenha activado uma sessão, caso contrário não consegue aceder ao sistema. No método de *spoofing* o utilizador não necessita de estar activo. Os métodos de *spoofing* mais utilizados são:

- *Spoofing* de IP – este ataque consiste apenas na alteração do IP do *cracker* mas não será suficiente para conseguir visualizar as respostas do utilizador, porque estas serão enviadas para o verdadeiro IP, o do utilizador. Mas se o *cracker* conseguir estar entre o utilizador e o sistema, figura 2.2, poderá visualizar a comunicação entre os dois.
- *Spoofing* de correio electrónico – este método é utilizado para esconder a verdadeira origem do correio electrónico, fingindo ser do verdadeiro utilizador, e podendo fazer envios de *malware* ou pedidos de informação importante.
- *Spoofing* de páginas Web – este método caracteriza-se por enganar o utilizador com uma falsa página da Web, onde são solicitados dados pessoais confidenciais. Usualmente o *spoofing* é utilizado em páginas que necessitam de autenticação do utilizador como por exemplo páginas Web de bancos ou de comércio electrónico.
- *Spoofing* do MAC – este método corresponde a alterar o endereço MAC para poder-se assemelhar a outro, permitindo á máquina com o MAC alterado ultrapassar alguns métodos de segurança como as ACL ou políticas dos *routers*, permitindo depois aceder a recursos ou simplesmente conseguir esconder-se na rede.

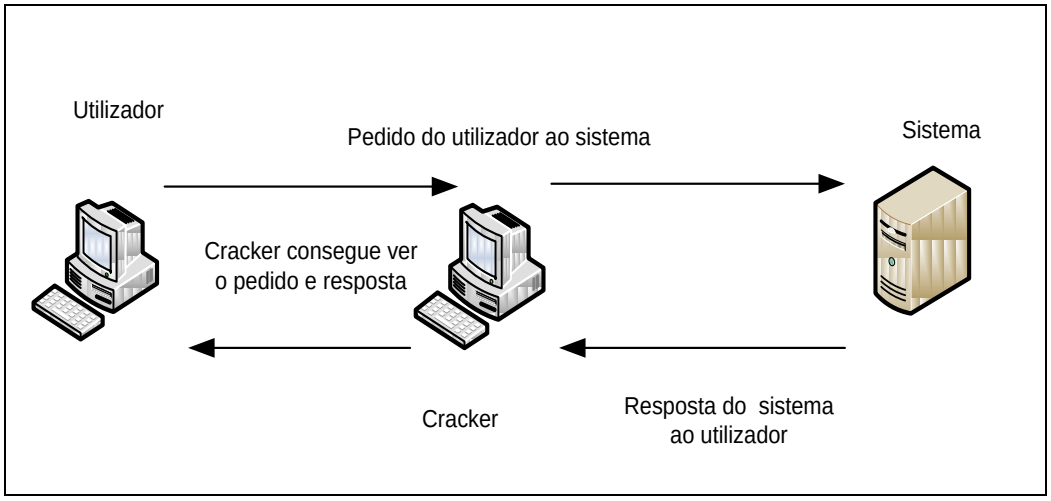

Figura 2. 2 - Processo de *Spoofing* [1]

Retransmissão

Como o *cracker* não pretende ser descoberto, utiliza um método que basicamente consiste em utilizar um terceiro sistema ao qual o *cracker* já obteve acesso para iniciar aí as suas intrusões. Deste modo este terceiro sistema será responsabilizado pelas acções do *cracker*.

***Trojan horse* e vírus**

O *trojan horse* é um *malware* constituído por duas partes: uma visível para o utilizador e outra oculta. A parte visível tem como finalidade captar o interesse do utilizador de maneira que este execute o programa sem desconfiar da sua origem. Enquanto este está a ser executado, a parte oculta executa *scripts*, *backdoor* ou provoca intrusões de outro tipo.

Um vírus tem como objectivo infectar o máximo possível de computadores. Um computador quando fica infectado torna-se num propagador desse vírus. Os vírus variam muito no seu efeito, podendo ser apenas impertinentes ou provocar algo de muito grave como, por exemplo, eliminar pastas essenciais no disco.

2.3.2 Através da LAN

Os ataques LAN são feitos por pessoas que têm acesso à rede interna da empresa. Por vezes os administradores de redes não lhe atribuem muita importância porque consideram que o seu pessoal é de confiança. Mas esta confiança pode ter consequências más, já que um grande número de intrusões é de origem interna. Pode haver intrusões a uma conta de um utilizador da empresa pelo exterior, permitindo deste modo ao *cracker* obter o acesso como se fosse um utilizador interno.

As intrusões mais utilizadas na LAN são:

Broadcasts

Através de um *broadcast* pode ser enviado um pacote para todos os endereços IP de um segmento de rede. Se nesse segmento de rede houver poucas máquinas, o tráfego gerado será reduzido, caso contrario haverá mais tráfego podendo provocar um DoS na rede.

Acesso a Ficheiros

Muitas empresas não têm listas de controlo (*Access Control Lists* - ACL) implementadas correctamente pelo que não podem limitar o acesso de utilizadores a zonas mais restritas. Se um *cracker* conseguir obter o acesso de um utilizador, para o que, normalmente basta a identificação do mesmo e a sua palavra-chave, pode aceder à informação mais importante da empresa.

2.3.3 As *Botnets*

A proliferação autónoma de *malware* é um problema que se tem vindo a agravar nos últimos anos e é conhecido por *Botnet* [4] (figura 2.3). Quando os *backdoors* são controlados remotamente podem tornar-se num problema muito sério, não só para os utilizadores domésticos como também para as empresas que receiem a espionagem industrial. Um *bot* é um programa, como por exemplo o *trojan horse*, que se instala numa máquina, sem o conhecimento do utilizador, permitindo depois ao *cracker* controlá-lo quando o desejar através de um protocolo, como por exemplo o IRC, HTTP ou P2P. *Botnets* são uma ameaça muito grave à segurança das redes por duas razões: em primeiro lugar por os recursos disponíveis serem imensos, se considerarmos que um *cracker* pode ter sob o seu controlo um elevado número de máquinas, podendo depois lançar ataques coordenados em grande escala às suas vítimas quer sejam empresas ou instituições bancárias; por ter um controlo tão vasto de recursos, pode enviar grandes quantidades de *spam*; a segunda razão, é que sendo uma *botnet* composta por vários computadores, permite obter uma quantidade de informação importante de cada bot, como por exemplo dados bancários.

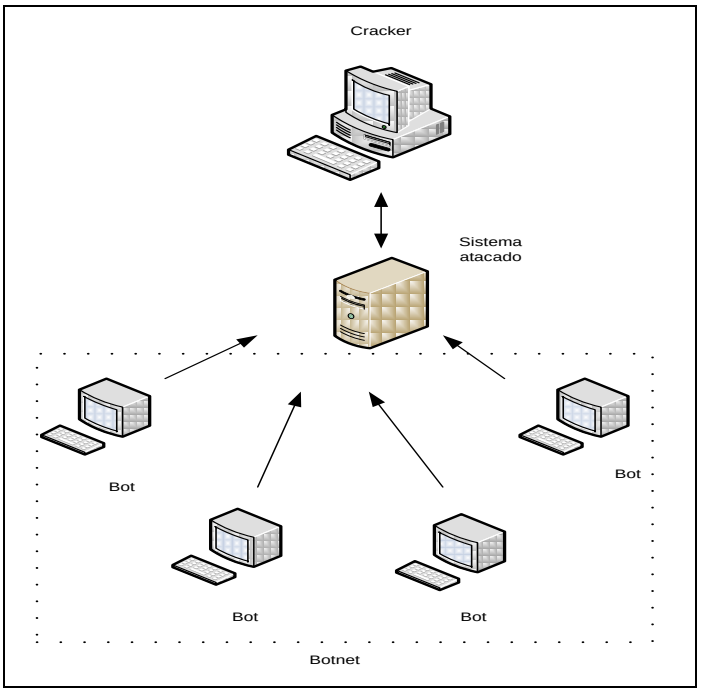

Figura 2. 3 - Uma *Botnet*

2.4 Dados Obtidos pela *Symantec*

Existe uma economia clandestina, criada por crackers, onde pode vender-se informação importante obtida por meios ilícitos. Os valores apresentados na tabela 2.1 referem-se ao período de Junho - Dezembro 2007, obtidos pela *Symantec* através de sondas colocados na Internet e dados obtidos pelos seus clientes.

Posição	***Item***	***Percentagem***	***Preços***
1	Números de Contas Bancárias	22%	€7-€700
2	Números de Cartões de Credito	13%	€0.29-€14
3	Identidade	9%	€0.70-€11
4	Contas de *sites* de leilões *online*	7%	€1-€8
5	Esquemas Fraudulentos	7%	€1.75/semana, €18 pelo projecto
6	Mailers	6%	€0.7-€7
7	Endereços de Correio Electrónico	5%	€0.58/MB-€7/MB
8	Palavras Passe de Correio Electrónico	5%	€2.80-€21
9	*Drop* (ofertas ou requisitos)	5%	10%-50% da quantidade total do *drop*
10	Proxies	6%	€1.05-€21

Tabela 2. 1 - Produtos à venda num mercado clandestino [5]

Há *malware* que utiliza vários meios para se propagar entre computadores, podendo incluir mensagens instantâneas, *Common internet File System (CiFS)*, *peer-to-peer (P2P)* e vulnerabilidades exploráveis remotamente. Na tabela 2.2 pode verificar-se que o meio mais utilizado para propagar *malware* na Europa, Médio Oriente e África (*Europe the Middle East and Africa* - EMEA) é o correio electrónico, seguido por ficheiros executáveis e o CiFS, o protocolo utilizado pelo Windows para permitir acesso a ficheiros arquivados.

Posição	***Meio de Propagação***	***Percentagem Regional (EMEA)***	***Percentagem Global***
1	File Transfer / Anexos de correio electrónico	37%	32%
2	File Sharing / Executáveis	27%	40%
3	File Transfer / CIFS	26%	28%
4	File Sharing / Peer-to-Peer	25%	19%
5	Vulnerabilidades Exploráveis Remotamente	24%	17%
6	SQL	4%	3%
7	Backdoor / Kuang 2	4%	3%
8	Backdoor / SubSeven	4%	3%
9	File Transfer/ MSN instant Messenger	4%	1%
10	File Transfer/ Yahoo! Messenger	2%	2%

Tabela 2. 2 - Meio de propagação de algumas ameaças [5]

O gráfico da figura 2.4 apresenta os resultados de alguns dos métodos utilizados para roubo de informação importante. Verifica-se que as intrusões de maior relevo, *Hacking* externo e o *Hacking* interno, são intrusões com objectivos bem claros, normalmente o roubo de informação importante.

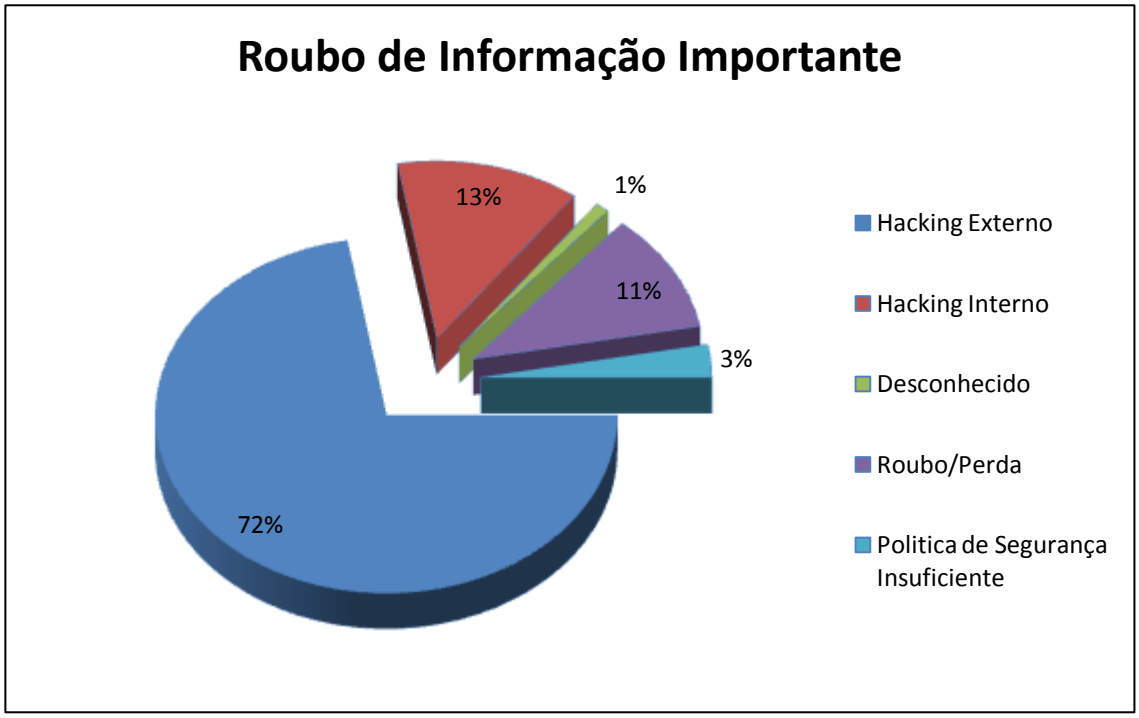

Figura 2. 4 - Roubo de Informação Importante [6]

Os *bots* como já foi referido são programas instalados num computador controlado por um *cracker*. Este problema está a revelar-se gravíssimo derivado ao poder de processamento e largura de banda que um *cracker* pode vir a ter. A *Symantec* tem conseguido identificar os *bots*, figura 2.5, através dos comportamentos coordenados dos ataques dos mesmos.

Um computador infectado com um *bot* activo significa que faz pelo menos um ataque por dia. No segundo semestre de 2007 os *bots* activos na EMEA eram cerca de 41% de todos os *bots* activos do mundo.

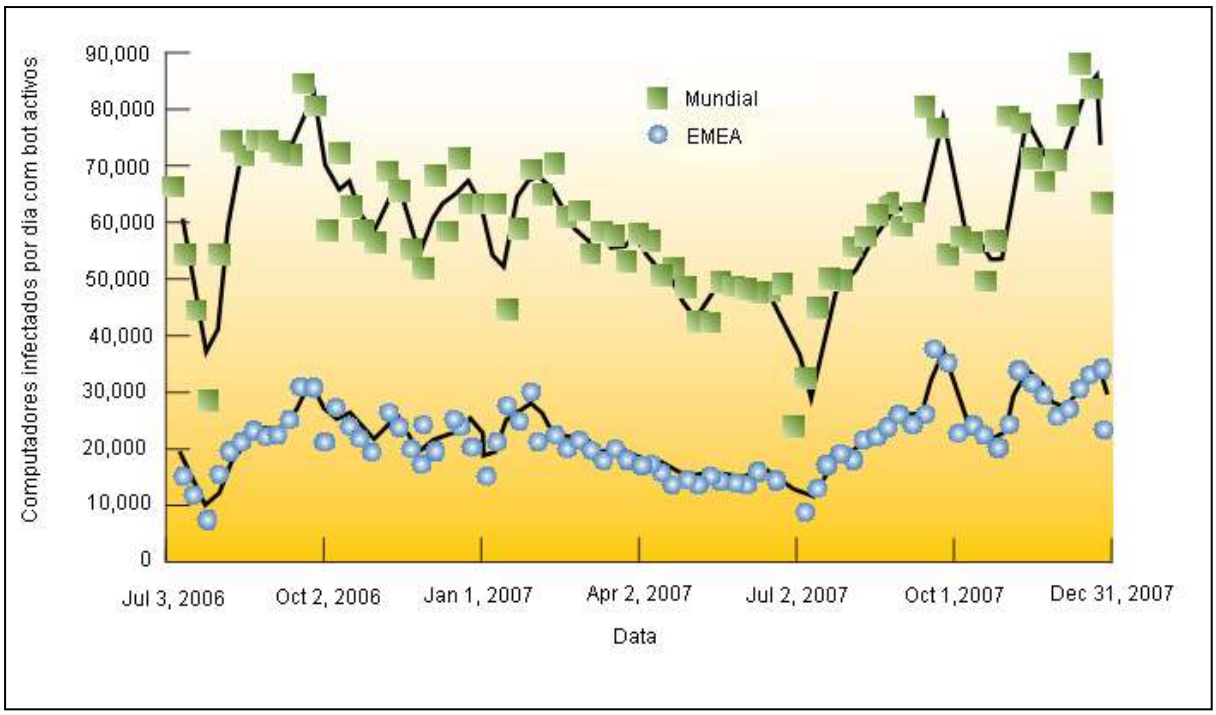

Figura 2. 5 - Computadores infectados com bots activos por dia, mundialmente e na EMEA [5]

2.5 A Segurança

A segurança é o processo de manter um risco perceptível a um nível aceitável [2]. A figura 2.6 mostra o equilibro que deverá haver entre funcionalidade, facilidade de utilização e segurança do sistema. Como se pode constatar, um sistema totalmente seguro, perde funcionalidade e facilidade de utilização.

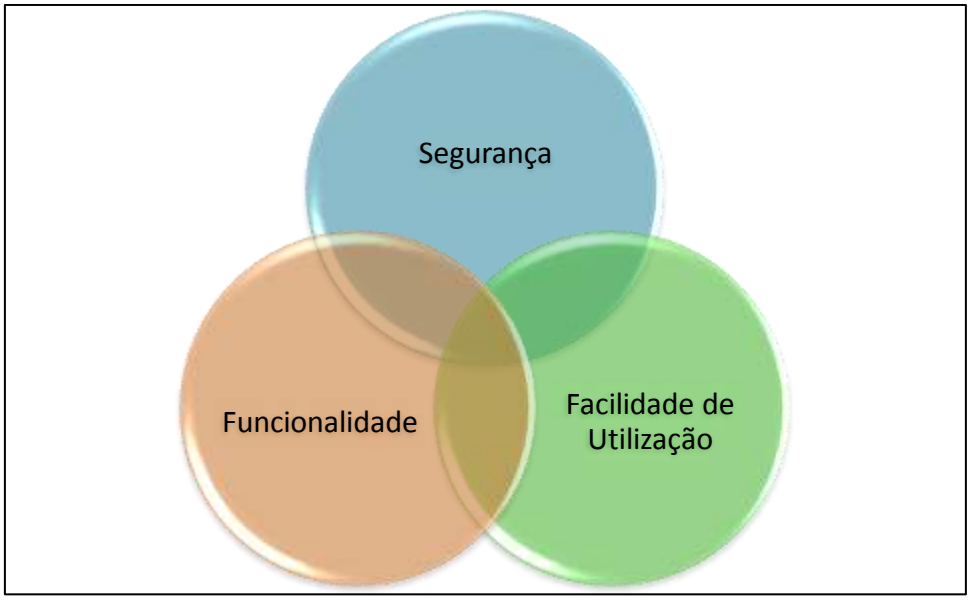

Figura 2. 6 - Diagrama entre a Segurança, Funcionalidade e Facilidade de Utilização [1]

O ex-director da *International Computer Security Association* escreveu que "a segurança é um processo, não um fim de linha" [2]. Pensar que implementado um bom sistema de segurança não há mais necessidade de intervir é errado e torna-se perigoso para a empresa.

2.5.1 IDS e IPS

A segurança nas redes é normalmente baseada na reunião de vários tipos de tecnologias como *routers, switches* e ferramentas como IDS e *firewalls*, para que em conjunto consigam oferecer uma maior protecção.

Um IDS é uma ferramenta de várias utilidades permitindo, por exemplo, a detecção de intrusões, a obtenção de informação sobre as áreas atacadas e a protecção da empresa de futuras consequências legais.

A detecção de anomalias por um IDS pode ser feita através do padrão de utilização ou pelos modelos de anomalias predefinidos (assinaturas). Se for baseado nos padrões de utilização, o comportamento normal do utilizador é registado como modelo de referência. Quando o IDS detecta um comportamento diferente do habitual, gera um alerta. Este processo tem como desvantagem o elevado número de alertas gerados, porque qualquer alteração ao padrão do utilizador é considerada como uma intrusão.

Um IDS que se baseia em modelos de anomalias predefinidos faz uma comparação do tráfego da rede com os modelos e, se houver uma igualdade, é gerado um alerta. Infelizmente este tipo de IDS só faz a comparação do tráfego com os modelos especificados, pelo que se houver um pacote com uma intrusão que não tenha modelo para comparação, não será detectado (falso negativo). Este método obriga, por isso, a uma constante actualização dos modelos. É no entanto muito mais utilizado que o anterior por gerar um menor número de alarmes falsos.

Existem dois tipos de IDS: o que monitoriza todo o tráfego da rede (*Network Intrusion Detection System* - NIDS) e o que monitoriza só o tráfego de um *host* (*Host Intrusion Detection System* - HIDS).

O *NIDS* monitoriza o tráfego de uma rede, podendo utilizar tanto o padrão de utilização como os modelos de anomalias predefinidos. A placa de rede é colocada em modo promíscuo para poder capturar todo o tráfego da rede. Os alertas são em tempo real e existe registo para o administrador de redes poder aprofundar melhor o tipo de intrusão e as suas consequências. Pode também ter-se várias sondas colocadas em pontos estratégicos na rede de forma a enviarem os alertas para um servidor [7] [8].

Enquanto um NIDS verifica todo o tráfego de uma rede para detectar as intrusões, um HIDS só monitoriza as intrusões do *host*. A verificação baseada no *host* teve o seu começo nos anos 80, antes de as redes serem tão complexas e estarem interligadas como hoje. Na altura era prática comum verificar todos os registos para qualquer anomalia na segurança. Os HIDS utilizados hoje têm como base o mesmo princípio mas o processo está mais simplificado e automatizado com as detecções em tempo real. O HIDS monitoriza todo o tráfego da rede direccionado para o host, não sendo necessário colocar a placa de rede em modo promíscuo. Pode utilizar-se tanto o método do padrão de utilização como o modelo de

anomalias predefinidas. O HIDS pode examinar os registos do sistema para quaisquer anomalias, como, por exemplo várias tentativas falhadas de *login*, e depois responder devidamente. É possível também verificar a integridade dos ficheiros no sistema, podendo avisar em tempo real de qualquer alteração. Este processo possibilita, também, saber se algum ficheiro foi criado ou eliminado [7] [8]. A utilização do HIDS contribui para uma defesa mais completa da rede, sendo possível detectar um cracker que conseguiu evitar o firewall.

O IDS é uma ferramenta passiva da segurança, não tomando nenhuma acção para impedir a intrusão apenas avisando o administrador de redes. Sendo necessário que haja uma pessoa competente para verificar as intrusões e decidir qual a acção correcta a tomar. Esta ferramenta para ter um bom funcionamento exige que esteja bem afinada ao tráfego da rede. Só assim se pode eliminar o maior número possível de alertas falsos.

A conjugação de várias ferramentas e tecnologias de segurança permite-nos obter uma maior protecção da rede embora não funcionem em conjunto. O sistema de prevenção das intrusões (*Intrusion Prevention System* - IPS) é um IDS e ainda tem capacidade de analisar o alerta, avaliar o impacto da intrusão e, se for necessário, tomar as acções apropriadas para eliminar a sua proliferação, alterando por exemplo as configurações do *firewall* para não permitir tráfego ofensivo ou terminando a sessão que originou o ataque através do bloqueio do endereço IP. Permite assim obter uma resposta automática à intrusão, evitando a necessidade do administrador ter de verificar os alertas para os solucionar. Mas para que um IPS funcione correctamente é necessário que a estrutura da rede seja tomada em consideração durante a implementação, o que implica um cuidado maior por parte do administrador. Em caso de alerta haverá uma maior informação dos diversos equipamentos de rede, permitindo uma resposta mais adequada.

Um bom sistema de segurança pode ser composto por um IDS e por um IPS na mesma rede. O IPS monitorizará o tráfego à entrada da rede, actuando contra os vírus e *worms*, enquanto o IDS se ocupará do tráfego interno da rede.

Infelizmente tem-se verificado que nem sempre as acções tomadas pelo IPS em caso de alertas são as mais correctas, porque um alerta falso origina o accionamento do IPS e a tomada de medidas provocando atrasos ou perdas na rede [2].

2.5.2 As desvantagens dos IDS

Um IDS não é infalível, sendo uma das suas maiores desvantagens o excesso de alertas falsos, o que acontece quando pacotes normais são considerados como tráfego maléfico. Existem também alguns métodos que se podem utilizar para enviar pacotes sem serem detectados. Um exemplo é a crição de uma avalanche de dados para que o IDS os não consiga processar todos. É também possível que o administrador de redes receba tantos alertas geradas por este processo que decida não os analisar e por isso alguns pacotes maliciosos conseguem passar sem serem detectados. Para este método podem usar-se ferramentas como o *Stick & Snot* ou o *scan decoy* do *Nmap*. Outro método possível é enviar o ataque através de pacotes não reconhecidos pelos modelos predefinidos do IDS, podendo utilizar-se as ferramentas *Fragrouter* ou *Nikto*.

2.5.3 Honeypots

Os *honeypots* são diferentes das outras ferramentas de segurança porque não estão limitados a resolver problemas específicos. São antes muito flexíveis e podem ser aplicados a muitas situações. Uma descrição genérica dos *honeypots* é que são recursos da segurança de redes, que têm como propósito serem atacados ou comprometidos, num ambiente que permite o registo e controlo dessas actividades [9].

O *honeypot* pode simular qualquer tipo de equipamento de rede como computadores, impressoras, routers ou switchs, o que possibilita a um *cracker* interagir com o equipamento virtual, podendo iludir o *cracker* enquanto se obtém mais informação sobre os métodos utilizados no ataque. Esta ferramenta tem uma diferença fundamental em relção às anteriores porque, para funcionar, necessita de ser atacada. Os *honeypots* geram poucos alertas, o que facilita a verificação dos registos pelo administrador de redes. Assim caso haja uma intrusão que não seja reconhecida pelos modelos de anomalias de um IDS, o *honeypot* pode possibilitar a obtenção da informação necessária para a resolução do ataque.

Os administradores de redes têm vindo a utilizar em conjunto com o *honeypot* o *honeytoken*, que funciona como um sinal de aviso. O *honeytoken* pode simular um utilizador falso, que normalmente tem o nome *Administrator*, um *email* falso ou um utilizador que tem permissões especiais para o acesso a uma base de dados. Se houver intenções maliciosas direccionadas a estes utilizadores, poderá ser o primeiro sinal de uma intrusão, permitindo uma maior atenção por parte do administrador de redes.

Existem duas categorias de *honeypots*:

- *Honeypot* de baixa interactividade – este tipo de *honeypot* tem uma interacção limitada e funciona normalmente através da emulação de serviços e sistemas operativos. A actividade do *cracker* é limitada aos serviços emulados e são os próprios serviços que restringem o risco ao conter a actividade do *cracker*. Estas acções também são conhecidas como *tarpit*. As vantagens dos *honeypots* de baixa interactividade são a simplicidade, a facilidade de manutenção e o facto de representarem um risco mínimo para o sistema. A configuração é também muito simples bastando apenas escolher quais os sistemas operativos, equipamentos e serviços a emular. As desvantagens deste *honeypot* são a informação limitada da actividade do *cracker* e o facto de serem concebidos apenas para capturar actividade conhecida. Independentemente da qualidade da emulação, um *cracker* pode detectar um *honeypot* de baixa interactividade. Os serviços emulados não os enganam eternamente e eventualmente percebem que os serviços têm sempre as mesmas respostas aos seus pedidos. Alguns exemplos deste tipo de honeypot são *honeyd* e *Specter* [9] [10].
- *Honeypot* de alta interactividade – este tipo de *honeypot* é mais complexo porque envolve sistemas e aplicações reais. Nenhuma aplicação é emulada e o *cracker* pode assim atacar sistemas com todas as funcionalidades. Existem duas vantagens deste método: a primeira, porque os *crackers* tem um sistema real para atacar, e o facto de se poder obter mais informação sobre os atacantes como, por exemplo,

os métodos utilizados e as ferramentas instaladas; com alguns *honeypots* é até possível saber quais as teclas carregadas; a segunda vantagem é facto de o *honeypot* de alta interactividade providenciar um ambiente para poder capturar toda a actividade incluindo qualquer novo método desconhecido. Infelizmente o risco para este *honeypot* é muito mais elevado porque, como o sistema é real, pode ser utilizado pelo *cracker* para atacar outros sistemas, obrigando à utilização de ferramentas adicionais como método de prevenção. Este tipo de *honeypot* é mais complexo, tanto na utilização como na manutenção. Alguns exemplos de *honeypots* de alta interactividade são *Honeynets* e *Symantec Decoy Server* [9] [10].

Os *honeypots*, para serem bem aproveitados, são posicionados estrategicamente numa rede, normalmente onde há IP´s livres e onde será mais provável existir um ataque. Em situações em que há vários IP´s livres é possível criar mais que um *honeypot* englobando-os depois num *honeynet*. Nestes casos existe também um *honeywall*, que se considera como um *gateway* para a *honeynet*, que não só serve como uma entrada invisível para a *honeynet* mas também permite ao administrador monitorizar todo o tráfego de entrada e saída da mesma. A figura 2.7 exemplifica a utilização de um *honeypot* numa rede.

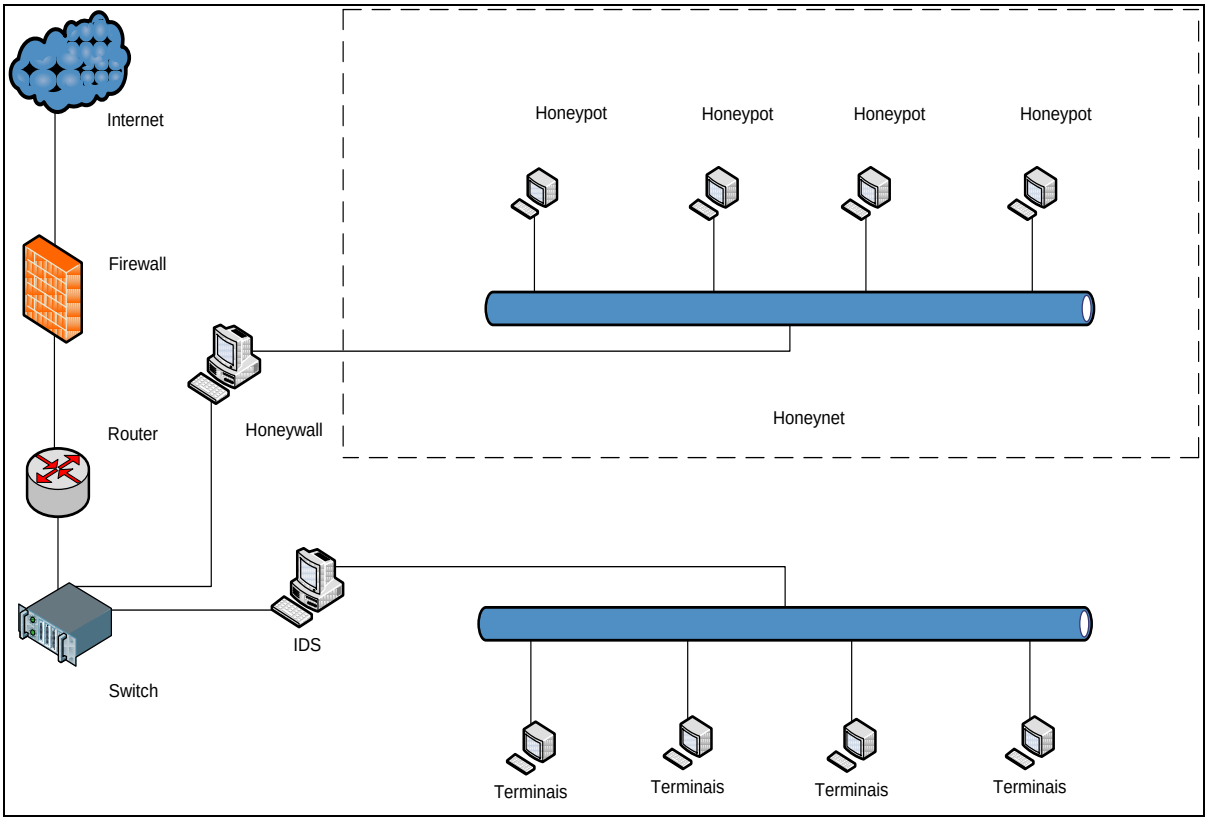

Figura 2. 7 - Exemplo da utilização de um *honeypot.*

Um *honeypot*, para bem cumprir os seus objectivos, é constituído pelos seguintes elementos:

- Ferramentas de monitorização: a razão de existir um *honeypot* é para obter o máximo de informação sobre o *cracker*.
- Dispositivo de alerta: para evitar que o administrador esteja constantemente a verificar o *honeypot*, normalmente existe um método de aviso, como o correio electrónico, SMS ou outros.
- Arquivo das teclas pressionadas: em *honeypots* mais evoluídos pode obter-se um registo de todas as teclas utilizadas pelo *cracker*.

2.5.4 As desvantagens dos honeypots

Pelo facto de apresentarem algumas desvantagens, os *honeypots* ainda não conseguem substituir as ferramentas existentes na segurança de redes, mas apenas serem utilizados para a melhoria da segurança já instalada [9].

O maior problema relacionado com o *honeypot* é só poder visualizar a actividade que lhe é direccionado. Se houver um ataque à rede mas não ao *honeypot*, este nem o saberá. Se um sistema for reconhecido como *honeypot*, os ataques a este serão evitados e os esforços concentrados nos outros sistemas [9] [10].

Se um *cracker* identificar um *honeypot*, poderá fazer *spoofing* da identidade do *honeypot* e atacar outros *honeypots* na mesma rede, o que vai provocar alertas falsas, talvez os suficientes para distrair o administrador, enquanto é feito um ataque verdadeiro ao sistema.

Um *cracker* poderá descobrir um *honeypot* pelo seu comportamento ou através dos pequenos erros que podem existir nas emulações como por exemplo, pela resposta do servidor Web [10].

2.6 Aumento do Sistema de Segurança

Independentemente do sistema operativo onde o IDS foi instalado, será sempre aconselhável um endurecimento do sistema. Os sistemas operativos põem à disposição do utilizador o máximo de ferramentas possível, mas para isso têm de usar outros programas e bibliotecas. Isto aumenta porém os riscos, com mais portas abertas e com a necessidade de fazer mais actualizações para assegurar a máxima segurança do IDS [11].

Durante a instalação do sistema operativo é aconselhável instalar só os programas necessários, evitando a instalação dos jogos, ferramentas multimédia e documentação de ajuda. É preferível a não utilização do sistema gráfico, não só pela falta de segurança mas também porque depois terá muito pouco ou nenhum uso durante o funcionamento do IDS. É igualmente aconselhável eliminar programas que foram utilizados durante a instalação do IDS, como os compiladores de rpm, C e C++, que podem ser utilizados para compilar programas maliciosos [7] [11].

2.7 Segurança Através do Hardware

2.7.1 TAPS

Por vezes é necessário criar portas permanentes de acesso distribuídas pela rede, de forma a possibilitar ao administrador a monitorização de segmentos de rede. Para que não haja a necessidade de desligar equipamento ou alterar configurações, podem utilizar-se *span ports* ou uma porta de acesso para teste (Test Access Port - TAP). Estas portas podem ser colocadas entre equipamentos como *routers* e *switches* para poder monitorizar passivamente todo o tráfego [12]. Este equipamento é muito utilizado na segurança das redes permitindo a colocação de sondas entre equipamentos para a sua monitorização. Como benefício, uma TAP proporciona ao equipamento de monitorização a possibilidade de não ser detectado pelos *crackers*. Uma TAP é normalmente constituída pelas portas de entrada e saída do tráfego da rede como também pelas portas para monitorização do tráfego.

Com uma TAP o tráfego da rede é recebido no equipamento de monitorização em tempo real e com todos os erros, sendo que os processos de regeneração ou divisão utilizados por ela não provocam atrasos ou alterações nos pacotes. Porém, o tráfego de rede dos *span ports* de um *switch* necessita de processamento interno o que provoca atrasos e nem todo o tráfego da rede é recebido pelo equipamento de monitorização. Isto porque dependendo da prioridade establecida no *switch* os pacotes com erros na camada 1 podem ser descartados e, serem considerados apenas os erros da camada 2 [12].

A utilização de TAPS tem várias vantagens, como o facto de não serem necessárias portas dedicadas por estes serem colocados entre equipamentos, como mostra a figura 2.8.

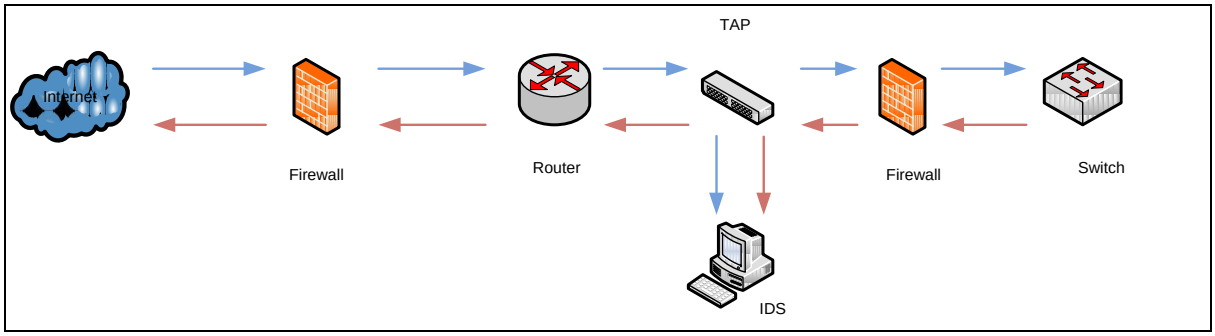

Figura 2. 8 - Exemplo da utilização de uma TAP em rede [12]

2.7.2 One Way Cables

Quando um IDS é utilizado como sistema de segurança de uma rede, é essencial que esteja tão inacessível quanto possível aos *crackers*. É desejável que o IDS não responda às solicitações, mas que continue a monitorizar a rede. Com a utilização de um *one way cable*, o IDS fica fisicamente impossibilitado de enviar dados, no entanto pode continuar a receber o

tráfego necessário para a monitorização. Este tipo de cabo é de fácil construção, basta retirar o par entrelaçado de cor: laranja/branco (1) e laranja (2), para evitar erros de transmissão; ficando-se com apenas 3 pares entrelaçados no cabo. Depois faz-se uma ligação sólida com uma secção do par retirado e o par de cor: branco/verde (3) e verde (6), como mostra a figura 2.9. No final haverá uma extremidade do cabo com 8 fios que será colocado na TAP, *hub* ou *span port* e a outra com 6 fios que será colocada no IDS. Permitindo que o tráfego seja direccionado para o IDS com umas ligações falsificadas.

Este cabo só pode ser utilizado com *hub* porque o tráfego direccionado ao IDS retorna à rede. Caso seja utilizado com um *switch* provocará confusões nas tabelas de MAC, isto porque o tráfego é redireccionado pela mesma porta por onde foi enviado.

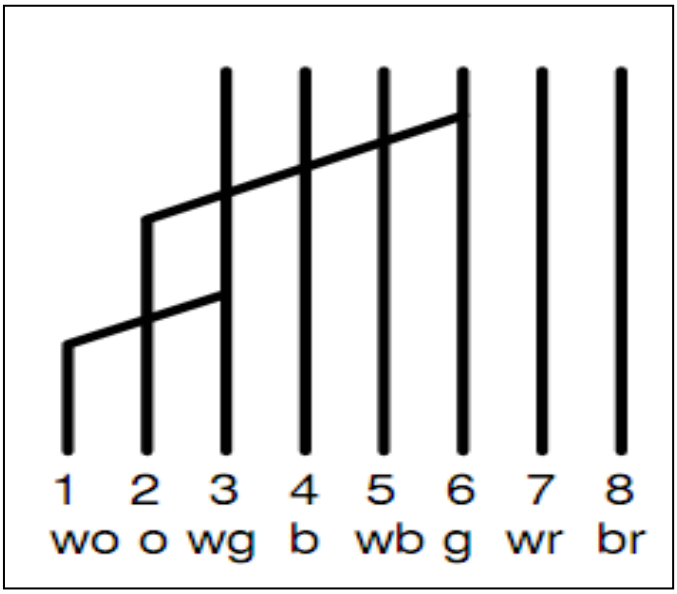

Figura 2. 9 – Ligações de um *one way cable*

3. Análise de Ferramentas Segurança

3.1 Introdução

Existem no mercado vários tipos de ferramentas para segurança de redes, tanto *open source* como comerciais. Para fazer uma melhor escolha, é necessário ter em consideração os atributos individuais de cada *software*.

Um dos factores mais ponderados é normalmente o custo do *software* pelo que muitas empresas optarem por *software open source*. Por vezes a utilização de software *open source* na implementação de um IDS permite superar a qualidade de um *software* comercial.

Neste capítulo vão ser apresentadas algumas das ferramentas mais utilizadas para implementar esquemas de segurança.

3.2 Detectores de Vulnerabilidades

3.2.1 Nessus

O projecto *Nessus* teve o seu começo em 1998 através de Renaud Deraison. Este projecto avançou porque os detectores *open source* de vulnerabilidades existentes, eram de fraca qualidade, em comparação com os produtos comerciais existentes no mercado.

Nessus tornou-se no detector mais utilizado mas, em 2005 o código do programa deixou de ser disponibilizado e em 2008 a licença *open source* foi retirada aos utilizadores comerciais [21].

O *Nessus* é um detector de vulnerabilidades robusto que possibilita a escolha do tipo de detecção que se pretende fazer, através dos vários *plugins* existentes, que são constantemente actualizados.

Entre as principais características do *Nessus* incluem-se a verificação de segurança local e remota, e o facto de através da *Nessus Attack Scipting Language* (NASL) o administrador de redes poder criar os seus próprios *plugins* [13].

O *Nessus* pode ser utilizado pelos sistemas operativos (SO) Windows, Unix, Linux ou Mac. Apresenta uma interface gráfica (*Graphical User Interface* - GUI) de fácil utilização, mas não tem interface de consola (*Console User Interface* - CUI). A empresa Tenable é responsável pelas actualizações dos *plugin.*

Esta ferramenta foi utilizada neste trabalho para detectar vulnerabilidades nas configurações das máquinas dos utilizadores e também nos servidores. Foram feitos testes às máquinas que apresentaram alertas de NetBios os quais correspondiam a possíveis portas abertas. Em todos os testes realizados não se utilizaram os *plugins* que podiam provocar um DoS, por haver a possibilidade de provocar o bloqueio dos servidores. O *Nessus* tem um

suporte de informação interligada com outras bases de dados, como por exemplo a base de dados do *Snort*, que permite a rápida resolução de problemas verificados durante o teste de vulnerabilidades.

3.2.2 Nmap (Network Mapper)

Nmap foi criado por um *hacker* especialista em segurança de redes conhecido pelo pseudónimo de Fyodor.

É um *software open source* muito utilizado para avaliar a segurança dos computadores nas redes facilitando também a tarefa do administrador quando este necessita fazer o inventário dos equipamentos disponíveis na rede (nem sempre um administrador tem conhecimento de todos os equipamentos, o que pode facilitar os ataques).

Nmap foi concebido para trabalhar com redes vastas mas funciona muito bem em redes pequenas. Mesmo que existam *firewalls* ou *routers* a dificultar a tarefa, *Nmap* consegue localizar os equipamentos instalados porque dispõe de técnicas evoluídas, que lhe possibilitam a obtenção de informação essencial como, por exemplo, qual o sistema operativo, as versões dos *softwares* instaladas e quais as portas abertas [22].

Nmap pode ser utilizado em sistemas operativos como Windows, Linux, Unix e Mac. Por ser *open source* e ter muitos utilizadores, está muito bem documentado. O GUI oficial é o *Zenmap,* derivado do *Umit* e criado por Adriano Marques. Também é possível utilizar o *Nmap* através do CUI.

Os testes efectuados durante a realização deste trabalho permitem concluir que o *Nessus* é mais simples de utilizar que o *nmap* e os resultados de ambos são semelhantes. Em geral o *Nessus* é também mais eficaz a detectar vulnerabilidades, tanto nas configurações como nas versões dos *softwares* instalados em cada máquina. No futuro poderá verificar-se que, por ser necessária licença para utilizar o *Nessus 3,* os utilizadores venham a preferir o *Nmap,* e daí resultar uma melhoria substancial desta ferramenta.

3.2.3 GFI LANguard

LANguard foi concebido pela empresa GFI que é líder no mercado dos produtos de segurança de redes. Esta empresa tem uma parceria com a Microsoft e tem acesso a alguns dos projectos mais restritos como o *Exchange Server JDP* [23].

O *LANguard* é um *software* que tem em consideração os três factores mais importantes da segurança de redes - gestão de *patches*, auditoria da rede e estudo de vulnerabilidades:

- Gestão de *patches* - depois de completar o exame da rede, o *LANguard* proporciona meios para instalar e administrar todos os *patches* em máquinas da Microsoft conseguindo-se assim obter um ambiente menos vulnerável;

- Auditoria da rede - disponibiliza toda a informação sobre a rede tal como o equipamento *USB* ligado, o tipo de *software* instalado, as portas abertas, e palavras--chave pouco seguras;
- Estudo de vulnerabilidades - é feito aproximadamente 15 000 avaliações de segurança. *LANguard* tem a capacidade de analisar vários sistemas operativos na rede como Linux, Unix, Microsoft e Mac.

Tudo isto é possível através de um GUI de fácil utilização. No entanto, sendo uma ferramenta comercial tem custos e o preço começa nos €185 para 4 IP´s. Existe a possibilidade de se experimentar a ferramenta durante 30 dias. Outro aspecto que limita a utilização da ferramenta é só poder ser utilizada nos sistemas operativos da Microsoft (Windows 2000 (SP4), XP (SP2), 2003 e VISTA) [23].

Em comparação com as aplicações *open source*, este *software* foi bastante mais fácil de instalar, necessitando apenas de uma palavra-chave para o uso do administrador. O GUI pode parecer muito complexo no início, mas rapidamente se percebe que é bastante intuitivo. Esta ferramenta tem como vantagem o fornecimento de informação detalhada das vulnerabilidades, permitindo assim resolvê-las facilmente. Um pormenor interessante desta ferramenta é a possibilidade de poder comparar vários testes de vulnerabilidades e identificar as diferenças. Um grupo de utilizadores do *Nmap* que avalia as ferramentas de segurança de redes concorrentes colocou esta ferramenta em segundo lugar logo a seguir ao *Nessus* [24].

3.3 Analisadores de Protocolos de Rede (*Packet Sniffer*)

3.3.1 Wireshark

Wireshark é a continuação de um projecto que teve início em 1998 como *Ethereal*, tendo o seu nome sido alterado por razões legais. A concepção do *Wireshark* teve a participação de vários especialistas de redes, os quais continuam a colaborar activamente na sua evolução. Esta ferramenta tornou-se líder na sua área, quer comparado com ferramenta open source quer com as de mercado comercial [25]. *Wireshark* possibilita analisar pacotes da rede em tempo real ou arquivados. É possível escolher o nível de profundidade na análise dos protocolos sendo uma das suas vantagens permitir visualizar os pacotes capturados num GUI, o qual proporciona cor bastante para se visualizar facilmente os diferentes protocolos. Para facilitar a análise dos pacotes é possível filtrar por IP, MAC ou VLANID. A informação obtida pode ser escrita em vários tipos de formatos, fazendo com que o *Wireshark* seja mais compatível com as ferramentas similares do mercado. É um *software* multi-plataforma podendo ser utilizado com o sistema operativo Windows, Linux, Unix e outros. Infelizmente *Wireshark* tem sido alvo de vários ataques sendo fundamental estar actualizado.

Os testes efectuados no decorrer deste trabalho permitiram verificar que, em alguns casos, a quantidade de informação oferecida pelo *Wireshark* é demasiado abundante o que dificulta a solução dos problemas. Daí ter sido utilizado também o *PacScope*, infelizmente devido à falência da empresa que detinha esta ferramenta houve uma estagnação no seu

desenvolvimento. No entanto esta ferramenta continua a ser do mesmo nível de profissionalismo que o *Wireshark*, mas a informação facultada pelo GUI é de mais fácil compreensão.

3.3.2 TcpDump

O TcpDump que, tal como a anterior ferramenta, permite capturar pacotes foi criado por Van Jacobson, Craig Leres e Steven McCanne, na altura investigadores de redes nos laboratórios de *Lawrence Berkeley*. É um *software* livre e foi muito utilizado até o aparecimento do *Wireshark*. O *TcpDump* funciona em vários sistemas operativos como Linux e Unix onde utiliza o *libpcap* para fazer a captura dos pacotes [26]. Para o sistema operativo Windows existe o *Windump* que utiliza o *WinPcap*. Esta ferramenta não possui um GUI e também não tem recebido funcionalidades novas, mas é constantemente actualizada e é mais segura que o *Wireshark*.

O *TcpDump* ao ser avaliado pelo grupo de utilizadores do *Nmap* foi colocado em terceiro lugar. É portanto uma ferramenta excelente que utiliza poucos recursos da máquina mas exige a utilização de um CUI.

3.4 Intrusion Detection System (IDS)

3.4.1 Snort (*The Pig*)

O *Snort* foi criado por Martin Roesch e é agora mantido pela *Sourcefire*, uma empresa fundada pelo próprio Martin. O objectivo inicial do projecto *Snort*, foi criar um IDS através de ferramentas *open source* existentes. Primeiro era necessário algo que fizesse a captura de pacotes e para tal foi escolhido o *tcpdump*, sendo depois adicionado um método de comparação de pacotes com os modelos de tipos de ataques conhecidos [7] [11].

Através de análise de protocolos, verificação de conteúdo e de vários pré-processadores existentes, o *Snort* consegue detectar as vulnerabilidades e comportamentos suspeitos. O administrador de redes também pode criar regras próprias que sejam mais apropriadas para a verificação de uma determinada rede.

O *Snort* funciona bem tanto em redes pequenas como nas grandes redes das empresas. Existe uma versão comercial da *Sourcefire* que oferece mais opções a nível empresarial e actualizações em tempo real dos modelos de tipos de ataques. Quem não desejar a versão comercial terá as actualizações das regras com cinco dias de atraso [27].

O *Snort*, combinado com o *BASE* ou *Sguil*, proporciona uma interface *Web* para a visualização dos dados obtidos.

É multi-plataforma podendo ser utilizado em sistemas operativos como o Windows, Linux, Mac e Unix. O *Snort* não apresenta um GUI sendo totalmente controlado pelo CUI. Programas como *Barnyard*, *Logwatch* e *Swatch* podem assistir o *Snort* no seu desempenho e velocidade [27]:

- *Barnyard* - este programa é responsável pelo manuseamento dos eventos com a base de dados. O *Snort* cria um ficheiro de *output* no formato *unified* (formato binário). Este ficheiro é posteriormente lido pelo *Barnyard* e os eventos enviados para uma base de dados. Se houver uma falha de comunicação entre a base de dados e o *Barnyard*, o envio dos dados é interrompido até que a comunicação seja restabelecida. Ao contrário do *plugin* utilizado no *Snort* para o envio dos eventos, havendo uma interrupção de comunicação entre a base de dados e o *Snort* o envio dos eventos continua, sendo assim possível a perda de dados. O *Snort* é assim libertado da tarefa do envio de eventos para a base de dados, podendo concentrar-se mais na análise de tráfego. É aconselhável, em redes de alta velocidade, implementar o *output* do *Snort* em binário, visto que, em comparação com os outros formatos, utiliza menos recursos.
- *Logwatch* – é um programa que analisa registos e gera relatórios periódicos. Estes relatórios podem ser depois enviados por correio electrónico. Análises de vários arquivos em diferentes máquinas podem ser enviadas num simples e bem explícito relatório facilitando assim qualquer intervenção necessária do administrador de redes. Através de instruções explícitas (fornecidas) podem criar-se *plugins* específicos.
- *Swatch* ou *Simple Watcher* – é um analisador de registos e tem a seu favor a possibilidade de fazer a análise em tempo real. Pode também ser configurado para, quando encontrar um certo registo, agir de determinada maneira. Depois de concluída a análise, o relatório pode ser enviado por correio electrónico.

Uma das grandes vantagens do *Snort* é a sua base de dados já que fornece informação adequada para a resolução rápida dos alertas. A existência de muitos utilizadores pode ajudar na resolução mais rápida de vulnerabilidades de *zero day*.

3.4.2 OSSEC HIDS

OSSEC HIDS é uma ferramenta *open source* concebido pela empresa *Third Brigade* [28]. Tem obtido sucesso em ambientes domésticos e também nas universidades, ISP´s e empresas. Esta ferramenta é servida por um potente motor de análise de dados e a sua eficácia não diminui com o aumento de tráfego sendo utilizado para a monitorização de *firewalls*, IDS, servidores *Web*, registo de autenticações e na detecção de *rootkits*. Esta ferramenta é multi-plataforma funcionando assim na maioria dos sistemas operativos como Windows, Linux, Mac e Unix. Cada 3 a 4 meses é lançada uma nova versão e existe também uma versão comercial que inclui apoio. Este sistema é totalmente controlado através do CUI.

Esta ferramenta está vocacionada para ser utilizada como um IDS num só *host*. A comparação desta ferramenta com a anterior não é fácil pois depende de vários factores como os objectivos que se pretendem atingir e o conhecimento técnico de que se dispõe para as poder aperfeiçoar na detecção das vulnerabilidades, dado que a maior parte dos IDS usam assinaturas e identificação de anomalias. Tem-se obtido bons resultados com o uso das duas ferramentas em simultâneo, tendo o *Snort* a funcionar como NIDS e o *OSSEC* como HIDS, e a fazendo a monitorização e visualização dos alertas obtidos pelo *Snort*.

3.4.3 O Dragon da *Enterasys*

O *Dragon Intrusion Defense* é um IPS/IDS concebido especificamente para as exigências de segurança dos ambientes das empresas, oferece avançadas funcionalidades que permite aumentar o nível de segurança. O IDS *Dragon* é normalmente constituído por vários sensores distribuídos pela rede e um servidor centralizado para a gestão (EMS - *Enterprise Management Server*). Através do EMS pode-se configurar os sensores e gerir os alertas. No caso de alerta o EMS pode actuar, por exemplo, com alterações às configurações do *firewall* ou dos *switches*. Existe uma extensa base de dados de modelos de anomalias predefinidos da *Enterasys* por onde o IDS pode comparar o tráfego, mas também possibilita um suporte para os modelos de anomalias predefinidos da base de dados do *Snort.* As plataformas do EMS são Linux ou Solaris.

3.5 Analisadores de Dados

3.5.1 BASE

Base (Basic Analysis and Security Engine) é uma aplicação baseada no código do projecto *ACID* (Analysis Console for Intrusion Detection) [27]. *Base* possui uma interface *Web* que processa e busca eventos numa base de dados gerados por um IDS, *firewall* e ferramentas de monitorização das redes. As suas principais características incluem a possibilidade de encontrar alertas que correspondam a um padrão específico. Utiliza um sistema de identificação de utilizadores, permitindo impedir o acesso a informação reservada. Podem-se visualizar graficamente os dados em função do tempo, sensores, protocolo e IP´s. Este sistema é multi-plataforma sendo possível utilizá-lo com Windows, Linux, Mac ou Unix.

3.5.2 Sguil

Sguil é um projecto concebido por *Robert "Bamm" Visscher*, um analista de segurança de redes. O projecto foi adquirido pela *Cisco* no princípio do ano 2008, continuando a ser actualizado pela equipa de *Robert*, mas agora como funcionário da *Cisco* [2] [29]. *Sguil* é uma ferramenta que integra os dados obtidos pelo *Snort* e os das sessões de *Security Analyst Network Connection Profiler* (SANCP). O componente principal do *Sguil* é um GUI que proporciona em tempo real os alertas obtidos, sendo seu objectivo final fornecer a melhor informação no tempo mais curto possível. Este sistema é multi-plataforma sendo possível utilizá-lo com os sistemas operativos Windows, Linux, Mac e Unix.

Enquanto o *BASE* permite visualizar os alertas armazenados na base de dados MySQL, podendo ordenar, apagar ou visualizar informação sobre os pacotes dos alertas e visualizar a informação de forma gráfica, o *Sguil* aborda os alertas de modo diferente permitindo não só

ordenar, apagar e visualizar o conteúdo dos mesmos, mas também compreender melhor se o ataque foi bem-sucedido e se teve acesso a mais informação.

No trabalho de campo efectuado verificou-se que depois de ter os alertas armazenados na base de dados, havia a necessidade de utilizar uma ferramenta que fosse capaz não só de ler ficheiros arquivados no formato *unified alert*, mas também que facilitasse a análise dos alertas obtidos e ser *standalone*, sem necessidade de qualquer outro *software*, como uma base de dados. O *Cerebus* veio preencher essa vaga, permitindo ao gestor de redes dividir os alertas e possibilitando depois eliminar os de menor gravidade para se concentrar nos restantes. Este *software* processa facilmente ficheiros de grande volume e possibilita uma ordenação dos alertas por IP, tipos de alertas ou por portas de envio. Infelizmente não permite a verificação do conteúdo dos pacotes.

3.6 Ferramentas para redes sem fios

3.6.1 Kismet

Kismet foi concebido por um especialista em segurança de redes Mike Kershaw também conhecido pelo pseudónimo de "*Dragorn*". É uma ferramenta *open source* que pode funcionar como um analisador de pacotes na rede, detector de redes ou um IDS. Funciona em redes sem fio com as normas 802.11a/b/g.

Comparado com programas semelhantes, o *Kismet* tem a particularidade de detectar passivamente redes sem fios (sem enviar pacotes detecta a presença de *access points* e seus clientes) ao contrário de programas como o *NetStumbler* [30].

Depois de as redes serem detectadas podem ser sinalizadas num mapa com um possível raio de acção. Esta ferramenta tem grande utilidade em *wardriving* (localizar zonas de redes sem fio ao percorrer áreas urbanas com um carro, para depois as utilizar).

Os pacotes analisados podem ser arquivados em formatos compatíveis com o *wireshark* ou *Airsnort*.

Para poder encontrar um maior número de redes, o *Kismet* suporta channel-hopping (salta de canal em canal não sequencial), podendo assim tirar maior vantagem na captura de pacotes sem haver sobreposição dos canais [30].

Este sistema é multi-plataforma, sendo possível utilizá-lo com os sistemas operativos Windows, Linux, Mac e Unix, e é totalmente controlado através de um CUI.

Infelizmente não é compatível com todas as placas de rede sem fios.

3.6.2 Aircrack-ng

O *Aircrack-ng* evoluiu do *Aircrack,* ambas ferramentas *open source* e desenvolvidas por Thomas d'Otreppe. *Aircrack-ng* pode ser utilizado em redes sem fios com as normas 802.11a/b/g e é constituído por programas de detecção de redes, analisador de pacotes e recuperadores de palavras-chave de WEP/WPA. Os programas incluídos são *airodump* (programa de captura de pacotes), *aireplay* (programa de injecção de pacotes), *aircrack* (recupera palavras-chave de WEP/WPA) e *airdecap* (descodificador de ficheiros capturados WEP/WPA) [31].

Aircrack-ng consegue facilmente recuperar palavras-chave em WEP de 40 a 512 bit (explorando os graves erros de segurança existentes) enquanto que para o WPA necessita de utilizar métodos avançados de criptografia [31].

O protocolo WEP já era conhecido como sendo inseguro. Em 2001 Scott Fluhrer, Itsik Mantin e Adi Shamir publicaram "*Weaknesses in the Key Scheduling Algorithm of RC4*", e aplicando o conhecimento obtido foi criado o ataque FMS, e a palavra-chave foi recuperada com 4,000,000 a 6,000,000 pacotes recuperados. Em 2004 um *cracker* conhecido por KoReK melhorou o ataque, necessitando apenas de 500,000 a 2,000,000 pacotes capturados.

Em 2005 Andreas Klein apresentou outra análise do algoritmo RC4 tendo esta análise sido depois utilizado por Andrei Pyshkin, Erik Tews e Ralf-Philipp Weinmann para criar um ataque PTW necessitando apenas de 40,000 a 85,000 pacotes capturados.

O *Aircrck-ng* utiliza o ataque FMS normal com uma optimização através dos ataques KoreK e PTW e depois de ter capturado pacotes suficientes consegue recuperar as palavras-chave. Os métodos utilizados fazem com que esta ferramenta seja uma das mais rápidas na sua área.

Para o protocolo WEP, basta a captura de alguns pacotes com o *airodump-ng*, em particular os vectores de inicialização (*Initialization Vector* - IV), para que o *Aircrack-ng* consiga recuperar uma palavra-chave. Um IV é um número que está em constante alteração e, quando se combina com uma palavra-chave, é utilizado para encriptar os dados. Os IV´s são utilizados para prevenir que haja uma sequência de texto igual à sequência anterior com a mesma mensagem depois de encriptada.

Derivado às muitas vulnerabilidades existentes no protocolo WEP, a recuperação de uma palavra-chave é fácil. O *Aircrack-ng* utiliza dois métodos fundamentais para obter as palavras-chave WEP. O primeiro método é através da abordagem PTW (Andrei Pyshkin, Erik Tews e Ralf-Philipp Weinmann), que tem a vantagem de necessitar de poucos pacotes para obter resultados. No segundo é utilizado uma combinação do método FMS/KoReK (Scott Fluhrer, Itsik Mantin e Adi Shamir) que integra vários ataques estatísticos com força bruta. A força bruta é um método bastante utilizado para obter palavras-chave. Baseia-se em experimentar todos os caracteres e combinações possíveis para recuperar a palavra-chave. No protocolo WPA/WPA2 obtenção de uma palavra-chave só é possível com o método do dicionário (força bruta), que corresponde a criar uma base de dados com palavras em ASCII ou hexadecimal que serão depois utilizadas pelo *Aircrack-ng* para determinar qual a palavra-chave correcta. Este método também pode ser utilizado com o protocolo WEP.

É uma ferramenta que funciona apenas através do CUI e é multi-plataforma podendo ser utilizado em Windows, Linux, Mac e Unix. Infelizmente não funciona com todas as placas de rede sem fios existentes no mercado.

3.6.3 Airsnort

Airsnort é uma ferramenta *open source* desenvolvido pelo grupo Shmoo e possibilita a obtenção de palavras-chave em redes sem fio locais que utilizam a norma 802.11b e WEP como método de segurança. O *Airsnort* fica passivamente a monitorizar todas as transmissões até que tenha reunido pacotes suficientes para recuperar a palavra-chave (necessita de aproximadamente 5-10 milhões de pacotes encriptados). *Airsnort* foi um das primeiras ferramentas no mercado a explorar os numerosos defeitos na segurança do WEP, sendo a mais flagrante a do algoritmo do RC4. Esta ferramenta funciona apenas através do CUI é de multi-plataforma podendo ser utilizado no Windows, Linux, Mac e Unix [32].

3.6.4 Comparação das Ferramentas

A grande diferença entre estas ferramentas é que o *Kismet* e o *AirSnort* são passivos (apenas estão à escuta do AP) enquanto o *aircrack-ng* permite a injecção de pacotes no AP, o que dá maior rapidez na descodificação da palavra-chave. Já foi verificado com o WEP de 128 bits, que as primeiras ferramentas demoraram vários dias a obter pacotes suficientes de um AP com pouco tráfego, para poder fazer a desencriptação da palavra-chave, enquanto com o *aircrack-ng* demorou menos de três horas. O grupo de utilizadores do *nmap* classificou o *Kismet* em primeiro lugar e o *aircrack-ng* em terceiro. Esta classificação é no entanto dicutível visto as características do *Kismet* estarem mais direccionadas para *wardriving* (descoberta de AP para mais tarde poder aceder e utilizar) por permitir a localização do AP num mapa, enquanto o *aircrack-ng* pode interessar mais aos gestores de segurança.

3.7 Honeypots

3.7.1 Honeyd

Honeyd é uma ferramenta *open source* criada por Niel's Provos. Esta ferramenta é um *daemon* que cria múltiplos *hosts* virtuais numa rede. Normalmente são utilizados os endereços IP´s não alocados, sendo depois possível configurar cada endereço para simular determinados serviços com determinados sistemas operativos. A simulação dos serviços é feita através de *scripts* de fácil compreensão, o que permite o utilizador também criar os seus próprios serviços. A configuração do *honeypot* é feita através de um simples e muito compreensível ficheiro de configuração, sendo possível um único *honeywall* ter múltiplos endereços todos com vários equipamentos activos. Este sistema melhora a segurança não só porque detecta potenciais ameaças como também "esconde" os sistemas reais no meio da rede virtual. Esta ferramenta funciona apenas através do CUI e é multi-plataforma podendo ser utilizado em Linux, Mac e Unix. Existe também uma versão para o Windows criada por Mike Davis [14] [33].

3.7.2 Nepenthes

Nepenthes (também conhecido como a planta carnívora) foi maioritariamente desenvolvido por Paul Baecher e Markus Koetter. Este *honeypot* de baixa interactividade consegue emular vulnerabilidades na rede obtendo posteriormente informação sobre potenciais ataques. Comparado com um *honeypot* de alta interactividade é mais seguro, visto a única interacção ser com os emuladores e o sistema real não ser explorado. Esta ferramenta funciona apenas através do CUI e é multi-plataforma podendo ser utilizado em Windows, Linux, Mac e Unix [34].

3.7.3 Comparação das Ferramentas

Estas ferramentas têm os mesmos objectivos, que são o de simular redes com terminais para obter a maior quantidade de informação sobre as máquinas com que o *cracker* interage. Apesar de parecidas nos objectivos, estas ferramentas são bastante diferentes. O *Honeyd* é uma ferramenta com uma boa reputação mas o seu desenvolvimento estagnou, enquanto que o *Nepenthes* é um recém-chegado ao mercado dos *honeypots*. *Nepenthes* tem técnicas de emulação de serviços muito superiores, o que permite obter mais e melhor informação sobre as intrusões. Este sistema foi concebido para extrair ficheiros binários de programas maléficos para posterior análise.

3.8 O Explorador de vulnerabilidades Metasploit Framework

Este programa é utilizado para o desenvolvimento de ferramentas na exploração de vulnerabilidades, como nas anomalias predefinidas (assinaturas) para um IDS. É muito utilizado por administradores de redes na verificação dos *patchs* e também nos testes de intrusão dos seus sistemas. Consiste numa lista extensa de opções que permite o utilizador escolher o tipo de falha a explorar e, quais os seus objectivos depois de o sistema já estar corrompido, como por exemplo adicionar um utilizador ou executar comandos. Este programa é *open source* e pode ser utilizado tanto em Windows, Linux ou Unix. Em Linux apenas existe o CUI mas para o Windows existe também o GUI.

Infelizmente também pode ser utilizado por pessoas (como *script kiddies)* que não estejam envolvidas com a segurança de redes permitindo obter acesso a sistemas que não estejam actualizados.

4. Estrutura da Rede do ASC

4.1 Introdução

Neste capítulo vão ser apresentados alguns dos aspectos mais importantes da infra-estrutura actual da rede de informática do ASC, o que inclui a sua segmentação, os diversos equipamentos tecnológicos presentes e alguns dos locais onde poderão ser encontrados. Será realçada a rede de dados localizada na aerogare já que foi esta a rede monitorizada através da sonda e TAP.

O esquema da estrutura da rede de dados do ASC está apresentado na sua totalidade no apêndice C. Em anexo é também acrescentada a rede de fibra óptica (apêndice D) e a rede *Voice over Internet Protocol* (VoIP - apêndice E), para se poder perceber melhor a localização das redes e dos seus equipamentos.

4.2 Segmentação da Rede

A rede de informática do ASC encontra-se espalhada por diversos pólos, que são interligados através de equipamentos existentes em cada edifício. Existem casos onde um edifício tem mais que um pólo [15]. Em consequência desta divisão, a rede apresenta uma topologia em árvore que é composta por vários níveis hierárquicos [15].

A rede encontra-se dividida em sub-redes e diversas redes locais virtuais (VLAN). A tabela 4.1 apresenta a estrutura e segmentação da rede.

Rede	***Endereço/Máscara***	***Notas***	***VLAN ID***
Rede DMZ (firewall)	10.10.X.X	Entre a *firewall* e a *proxy*	X
***HeartBeat* (*Exchange*)**	10.10.X.X		X
Telefones (VoIP)	10.14.X.X	65534 Equipamentos	X
Floresta ANA/DSI	192.168.X.X		X
Sistema FIDS	192.168.X.X	Também ocupa a rede 81 (máximo 510 máquinas)	X
Acesso Operacional	192.168.X.X	Máximo de 32 operadores com 5 máquinas cada	
Comando e Controlo	192.168.X.X	62 Equipamentos de rede	X
Impressões	192.168.X.X	Sistema de Impressão	X
Sistema BHS	192.168.X.X	62 Equipamentos de rede	X
CCTV	192.168.X.X	62 Equipamentos de rede	X

Autómatos	192.168.X.X	30 Equipamentos de rede	X
SCC – Serviço de controlo de comunicações	192.168.X.X	30 Equipamentos de rede	X
Floresta DASC	192.168.X.X	Também ocupa a rede 85 (máximo 510 máquinas)	X
Rede Ethernet	192.168.X.X	254 Equipamentos de rede	X
Rede ADSL	192.168.X.X	Máximo de 32 operadores com 5 máquinas cada	
Alarmes (SADI)	192.168.X.X	62 Equipamentos de rede	X
Voz (gestão)	192.168.X.X	62 Equipamentos de rede	X
Controlo de Acessos (SACA)	192.168.X.X	126 Equipamentos de rede	X
Direcção Comercial	192.168.X.X	Rede a reconfigurar em sub-redes	X
UPS´s	192.168.X.X	254 Equipamentos de rede	X
Raio X/Pórticos	192.168.X.X	126 Equipamentos de rede	X
CAMS	192.168.X.X		
Livre	192.168.X.X		
APIS (Controlo das mangas/iluminação da pista)	192.168.X.X	254 Equipamentos de rede	X
BeaconMaster	192.168.X.X	AIRNET - Gestão dos APs	X
GroundNet	192.168.X.X	AIRNET - Servidores	X
AirNet	192.168. X.X	AIRNET - Móveis	X
Livre	192.168. X.X		
Livre	192.168. X.X		
Sistema de Informação Horária	192.168. X.X		X

Tabela 4. 1 - Estrutura de endereços IP da rede do ASC

O endereço 192.168.X.X está associado ao equipamento principal da Aerogare, *Matrix N7*, que possui 6 cartas de rede a partir do qual derivam as ligações para os restantes equipamentos [15].

Numa primeira fase foram monitorizadas as redes 192.168.X.X (rede operacional) e 192.168.X.X (rede do domínio do ASC - DASC) por serem as que mais problemas podiam apresentar. Verificou-se mais tarde, pelos alertas obtidos, que havia também a necessidade de implementar uma sonda IDS para monitorizar o tráfego do *router*.

A tabela 4.2 apresenta a estrutura da rede operacional 192.168.X.X. Esta rede não só fornece serviços administrativos como também inclui serviços a entidades externas à ANA.

Rede	*Endereços*	*VLAN ID*
SITATEX	192.168.X.X	X
Acesso ISDN	192.168.X.X	X
Atlantis	192.168.X.X	X
FIDS/SGO	192.168.X.X a 192.168.X.X	X
TAP	192.168.X.X a 192.168.X.X	X
SPdH	192.168.X.X a 192.168.X.X	X
Alf	192.168.X.X a 192.168.X.X	X
SEF	192.168.X.X a 192.168.X.X	X
PJ	192.168.X.X a 192.168.X.X	X
PROSEGUR	192.168.X.X a 192.168.X.X	X
PSP	192.168.X.X a 192.168.X.X	X
CLA	192.168.X.X a 192.168.X.X	X
GF1	192.168.X.X a 192.168.X.X	X

Tabela 4. 2 -Estrutura da rede de Acesso Operacional

A rede 192.168.X.X está associada ao acesso da *internet* (ADSL) por entidades externas à ANA, sendo que a ANA apenas necessita de fornecer as infra-estruturas e o equipamento activo.

Rede	*Endereços*	*VLAN ID*
Quiosque	192.168.X.X a 192.168.X.X	X
Informática	192.168.X.X a 192.168.X.X	X
Ibéria	192.168.X.X a 192.168.X.X	X

PJ	192.168.X.X a 192.168.X.X	X
SABA	192.168.X.X a 192.168.X.X	X
Climex	192.168.X.X a 192.168.X.X	X
Petrogal	192.168.X.X a 192.168.X.X	X
Iberlim	192.168.X.X a 192.168.X.X	X
Siemens	192.168.X.X a 192.168.X.X	X
LAS	192.168.X.X a 192.168.X.X	X
AirEvents	192.168.X.X a 192.168.X.X	X
Nippon	192.168.X.X a 192.168.X.X	X
Swiss	192.168.X.X a 192.168.X.X	X
Livre	192.168.X.X a 192.168.X.X	X
SGEL	192.168.X.X a 192.168.X.X	X
Internet Corners	192.168.X.X a 192.168.X.X	X
AeroNorte	192.168.X.X a 192.168.X.X	X
SafePort	192.168.X.X a 192.168.X.X	X

Tabela 4. 3 - Estrutura da rede ADSL

4.3 Redes Locais Virtuais (VLAN)

A rede do ASC encontra-se dividida em várias redes locais sobre uma rede local física que é composta por vários equipamentos activos, o que permite reduzir o tráfego de *broadcast* e aumentar a segurança, concedendo uma maior flexibilidade à rede e diminuindo a quantidade de informação irrelevante que circula na rede [15]. As VLAN podem ser

baseadas em agrupamentos de portas de comutadoras, endereços MAC, endereços de rede ou numa combinação das anteriores.

O agrupamento existente no ASC é baseado na associação e atribuição de VLAN´s a determinadas portas. No presente caso, um *switch* faz o *forward* das tramas apenas para as portas da mesma VLAN. Podem co-existir várias VLAN´s no mesmo comutador sendo o protocolo predominante o IEEE 802.1Q. Para cada porta do *switch* é definido o *Port VLAN Identification* (PVID), isto é, a porta associada a uma determinada VLAN. Assim os *switches* só encaminham tramas, incluindo o *broadcast*, entre portas pertencentes à mesma VLAN, conseguindo assim poupar a largura de banda [15].

4.4 Rede de Área Alargada

Uma *Wide Area Network* (WAN) é uma rede que está geograficamente distribuída por longas distâncias. A ANA utiliza uma rede WAN para conseguir interligar todas as redes dos diversos aeroportos onde também será incluindo o Novo Aeroporto, e assim conseguir optimizar os recursos oferecidos nos diversos aeroportos (figura 4.1).

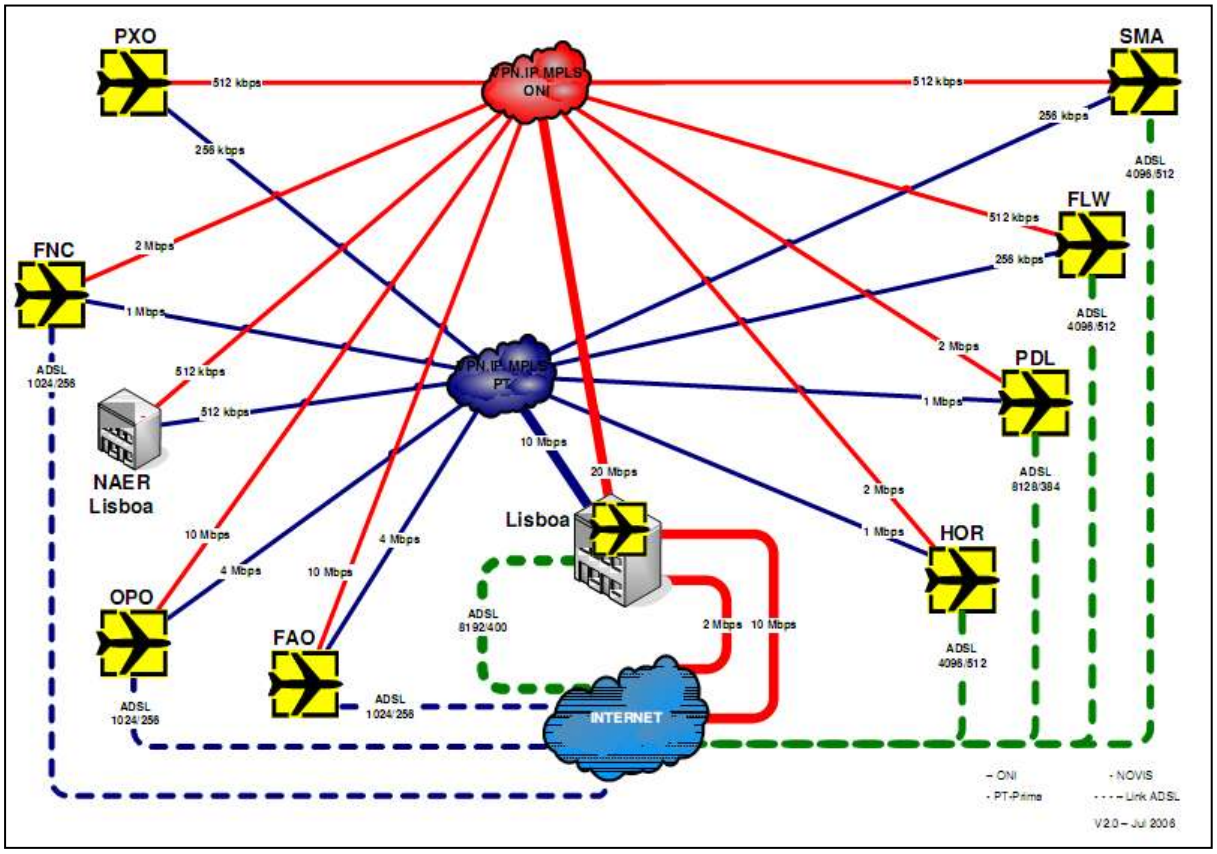

Figura 4. 1 - Rede WAN

Com a rede WAN, os serviços como o correio electrónico, sistemas de gestão e facturação e outros, ficam assegurados entre todos os locais da rede. Ficam ligados em rede

sistemas informáticos distribuídos por uma área vasta para exclusivo uso e administração da ANA.

A rede local do ASC liga-se a dois operadores de telecomunicações nacionais - a PT Prime e a ONI - através de encaminhadores à rede VPN.IP sobre a tecnologia MPLS. Nesta rede existem duas linhas dedicadas para a comunicação entre o ASC e o Aeroporto da Portela, o mesmo acontecendo no aeroporto de Faro e os arquipélagos. A única excepção é o aeroporto das Flores com o qual a ligação é feita via satélite [15].

A gestão e monitorização da WAN são efectuadas pelo Aeroporto de Lisboa, onde é utilizado a ferramenta da CA (*Computer Associates*) o *Spectrum* (figura 4.2).

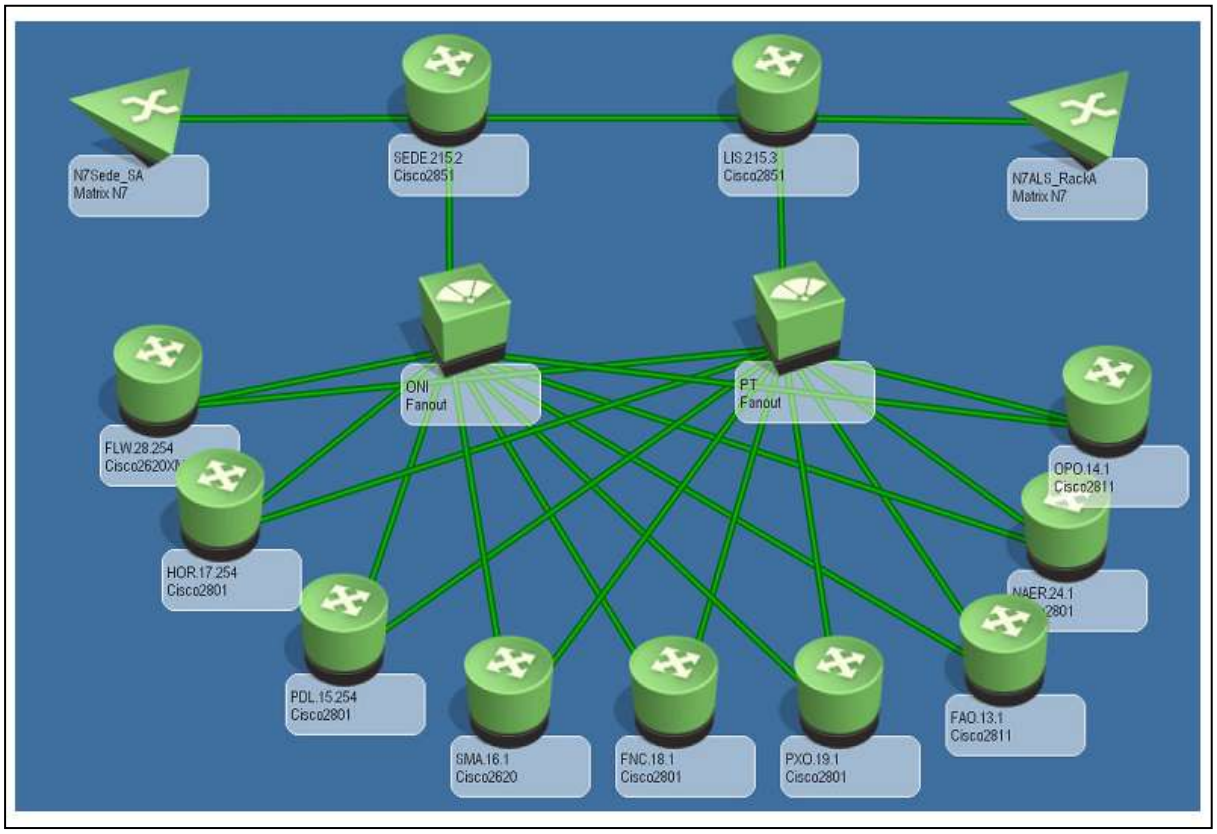

Figura 4. 2 - Rede WAN visualizado no *Spectrum* no Aeroporto da Portela

FLW - Aeroporto das Flores
HOR - Aeroporto da Horta
PDL - Aeroporto de Ponta Delgada
SMA - Serviço Móvel Aeronáutico
FNC - Aeroporto do Funchal
PXO - Aeroporto de Porto Santo
FAO - Aeroporto de Faro
NAER - Novo Aeroporto
OPO - Aeroporto do Porto

4.5 Tecnologias de Rede

Como já foi referido, o ASC possui como tecnologias de comunicação a *Ethernet* e o ADSL. A Ethernet está presente em duas variantes: *Fast Ethernet* e *Gigabit Ethernet*. Para a primeira é utilizada cablagem UTP 5 e 6 e também a fibra óptica e multimodo (100-Base-FX). O *backbone* da rede está presente em *Gigabit Ethernet* com fibra óptica multimodo (1000-Base-SX) e monomodo (1000-Base-SX) com débitos de 1Gbps.

A tecnologia de comunicação *Asymmetric Digital Subscriber Line* (ADSL) é utilizada pela ANA para uso interno e também para fornecer *Internet* às entidades externas situadas no Aeroporto. Existem actualmente 3 *links* ADSL da operadora nacional PT para o ASC. Dois são disponibilizados para uso interno da ANA a 6Mbit e são multiplexados para redundância, enquanto o terceiro fornece *Internet* aos clientes externos a 1Mbit. A rede de acesso à *Internet* para clientes externos é separada da outra, estando apenas a instalação e disponibilização dos equipamentos de rede a cargo da ANA.

Como a tecnologia de transmissão sem fios oferece simplicidade na sua instalação e gestão, tornou-se numa forma complementar à LAN já existente no ASC [15]. Há várias zonas de acesso à *Internet* sem fios no ASC, também conhecidos como *Hotspots:* a Aerogare, zona VIP, Rent a Car, SABA, serviços administrativos e o terminal de carga. Os pontos de acesso são assegurados por um *Aironet* 1100 da Cisco que respeita a norma IEEE 802.11g operando na faixa de frequência de 2,4 GHz.

4.6 Equipamentos da Rede

A actual rede do ASC é maioritariamente composta por equipamentos da linha *Enterasys,* como por exemplo o *Matrix N-Series* e *Matrix V2 Group Switch.* Estes equipamentos são dimensionados e configurados para garantir estabilidade e segurança á rede.

4.6.1 Enterasys DFE Platinum

Os módulos DFE Platinum são dos mais sofisticados comutadores Ethernet disponível no mercado, sendo um dos primeiros, na sua classe, a ter compatibilidade com *hardware* IPv6. Disponibilizam ferramentas de administração para facilmente classificar, dar prioridade e proteger as comunicações de voz, vídeo e dados. Nestes módulos destaca-se a política de multi-utilizador e multi-método de autenticação por porta, fornecendo um maior controlo de acessos à rede. Este módulo implementa uma arquitectura de fluxo para gerir as aplicações e os utilizadores individuais e não apenas portas e VLANs. Inclui também uma política de inspecção profunda de pacotes que detecta e responde automaticamente a ameaças de segurança, podendo assim melhorar a confiança sem prejudicar a qualidade do serviço, o que os torna ideias para *backbones.*

4.6.2 Chassis Enterasys Matrix N-series

As interfaces DFE podem ser instaladas em qualquer equipamento da serie *Matrix N*, nomeadamente a *Matrix N3, Matrix N5* e a *Matrix N7*.

As *Matrix N-Series* são inovadoras na sua concepção, oferecendo uma vasta gama de opções de conectividade e características avançadas, aumentando assim a capacidade de controlo sobre a rede. Com estes módulos o custo é drasticamente reduzido já que não há necessidade de compra posterior de componentes adicionais tais como módulos de supervisão, módulos de router ou módulos de gestão.

A escolha de qual o equipamento a utilizar é baseado na densidade, capacidade e desempenho necessário. Na rede actual do ASC estão a ser utilizadas as *Matrix N3* e *Matrix N7*, que podem ser encontrados no edifício A1 e A2.

4.6.3 Entrasys X-Pedition SSR 8000/8600

O *Smart Switch Router8000/8600* é um equipamento com funções de comutador e encaminhador, concebido para o tráfego gerado no *backbone* de pequenas ou médias redes. Oferece um elevado desempenho, mesmo com os filtros de segurança activos, podendo ser configurados para ser utilizados com *Ethernet* de diferentes velocidades, permitindo assim o uso de *Fast Ethernet* e *Gigabit Ethernet.* Este equipamento no ASC foi configurado para suportar grandes volumes de dados já que há diversas ligações do ASC a convergir para este equipamento, como exemplo o PT Norte/ Parque 4 e o complexo de carga.

4.6.4 Sun Microsystems Fire V210

O servidor *Fire V210* foi concebido para ter um alto nível de desempenho em serviços Web. Inclui processador de alta velocidade, pode ser configurados para ser utilizado com *Ethernet* de diferentes velocidades, permitindo assim o uso de *Fast Ethernet* e *Gigabit Ethernet*, o que o torna num excelente servidor para aplicações Web, DNS, autenticação, etc. Na rede do ASC este aparelho é utilizada para implementar um firewall.

4.6.5 Enterasys Matrix C2 (C2G124-48P)

O Matrix C2G124-48P oferece funcionalidades avançadas de comutação (camada 2) e encaminhamento (camada 3) com um vasto recurso incorporado de segurança. O C2G124-48P contém 48 portas, quatro das quais são para o *uplink* de receptores ópticos e permite o uso de *Fast Ethernet* e *Gigabit Ethernet*. Este equipamento encontra-se em uso em vários locais do ASC.

4.6.6 Enterasys Vertical Horizon VH-2402S2

Este *switch* possui 24 portas RJ45 de 10/100 Mbps com opção de dois *slots* de expansão que podem ser utilizados como ligações de *uplink* ao núcleo da rede por meio de 100-Base-FX ou 1000-Base-X, existe um *slot* dedicado à gestão para facilitar seu desempenho. A Vertical Horizon é adequada para suportar altos volumes de tráfego e aplicações intensivas da rede, é ideal para VLAN´s padrão IEEE 802.1Q. Quando implantado numa configuração empilhada pode ser gerida como uma única entidade.

4.6.7 Enterasys *Matrix* V2

O V2 é um *switch* que possui 24 portas de 10/100 BASE-TX com duas *slots* de expansão que podem ser utilizados para ligações de *uplink* ou para permitir aumentar o número de módulos. Podem ser geridos através de uma simples interface num servidor *Web*, telnet ou aplicações baseadas em SNMP. O *Matrix V2* tem vários tipos de protecção para prevenir contra o uso não autorizado, como ACL, 802.1X e a possibilidade de bloquear portas de específicos MAC.

4.7 Equipamentos para Teste e Monitorização

Por vezes é necessário monitorizar o tráfego duma rede para melhorar o seu desempenho e segurança mas sem prejudicar o seu funcionamento. Com o aumento da complexidade das redes, os equipamentos foram obrigados a encorporar tecnologia que permita a monitorização da rede através de *mirror/Span ports*, que é uma cópia do tráfego enviada para uma determinada porta. A utilização deste método não é isento de problemas, pois pode haver perda de pacotes caso não haja memória suficiente para a cópia além da necessidade de haver pelo menos uma porta livre. Os TAP permitem monitorizar uma rede sem que haja qualquer interrupção ou atraso no envio dos pacotes. Foram adquiridos pelo ASC dois TAP, um fixo e outro portátil, permitindo o último monitorizar temporariamente equipamentos sem alterações na gestão da rede.

4.7.1 FATAP-2000 BT/SX

Este equipamento tem capacidade para monitorizar duas redes simultaneamente em *full-duplex*, operando a 10/100/1000 Mbps com conexões em RJ-45 ou em fibra óptica. Existem duas portas de TAP por rede, o que possibilita a utilização de um IDS e uma ferramenta de monitorização em conjunto.

Caso haja uma falha na alimentação do TAP, as redes que estão a ser monitorizadas não serão afectadas, isto é, o tráfego continuará a ser transmitido. Este equipamento é facilmente configurável através do programa *FLOWControl*, o que permite não só escolher vários tipos

de filtros, como VLAN ou IP, mas também seleccionar a porta para onde se destina o envio dos dados.

4.7.2 ATAP-GIG BT-BT

O ATAP-GIG tem a particularidade de ser portátil, o que permite ao gestor de redes uma rápida intervenção sem grandes alterações à rede. Pode monitorizar uma rede em *full-duplex*, operando a 10/100/1000 Mbps mas apenas com portas RJ-45 e possui também duas portas TAP para permitir simultaneamente a utilização de duas ferramentas na monitorização dos dados.

5. Implementação de um Sistema de Segurança de Teste

5.1 Introdução

Dada a complexidade da rede do ASC e o perigo que podia representar para o seu funcionamento a instalação de novo *software,* optou-se numa primeira fase pela implementação de um sistema de segurança num ambiente mais restrito, de teste.

A escolha das ferramentas para a implementação do IDS foi baseada em programas *open source* por estas serem a custo zero e também por serem muito utilizadas o que facilita o apoio que se mostre necessário. Utilizando várias ferramentas *open source* pode obter-se um IDS de mais elevado nível. No início do estágio verificou-se que já havia muitos alertas gravados na base de dados do IDS actual e este não estava actualizado.

Neste capítulo vão descrever-se os vários sistemas implementados apresentando justificações para algumas opções tomadas.

5.2 O Sistema já implementado

No sistema já implementado constava um servidor IDS e dois sensores. No servidor estava instalado o Sistema Operativo *Suse* v10.0 com a versão 5.0.19 do *MySQL* e nos sensores o mesmo Sistema Operativo e a versão 2.4.4 do *Snort*. O primeiro sensor monitorizava o tráfego do servidor *November* enquanto o segundo monitorizava todo o tráfego no *router* que faz a ligação com Lisboa. Como o número de alertas era demasiado elevado para descobrir quais os que eram de risco real para a rede, foram todos apagados. Foi então decidida a implementação de um novo IDS.

5.3 A implementação do IDS de teste

Atendendo às características da rede, optou-se pela implementação de um NIDS utilizando o método de modelos de anomalias predefinidos. Desta forma pode-se fazer uma monitorização de todo o tráfego da rede e definir os modelos de comparação necessários para diminuir os falsos alertas.

Foi utilizada a ferramenta *Snort* que é constituída por cinco componentes: libpcap, descodificador de pacotes, pré-processador, motor de detecção e *output*. O tráfego, quando

chega à placa de rede (*Network Interface Card* - NIC) do IDS, segue o seguinte caminho (figura 5.1) [7] [11]:

- Libpcap - este software é um elemento a instalar além do *Snort*. É utilizado para ler e manipular os pacotes recebidos pelo NIC [7].
- Descodificador de pacotes - os pacotes são descodificados para determinar qual o protocolo utilizado. Nesta fase, verifica-se também se o comportamento dos dados é o normal para aquele tipo de protocolo. Este componente pode também alertar se existirem cabeçalhos danificados, pacotes demasiado longos ou opções de TCP erradas ou irregulares [11].
- Pré-processadores - são vários os *plugins* que permitem transformar os dados recebidos do descodificador para parcelas mais aceitáveis com que o *snort* possa trabalhar. Os pré-processadores são escolhidos através do ficheiro de configuração. Se for decidido não utilizar os pré-processadores, alguns dados maliciosos poderão não ser considerados pelo *Snort* visto não poder ser visualizados. Existem vários ataques possíveis para evitar o IDS, por exemplo, dividir os dados em vários pacotes para os depois reagrupar no sistema [7] [11].
- Motor de detecção - este é o componente do *Snort* que se considera mais importante, porque compara os dados vindos dos pré-processadores com as regras predefinidas [7].
- *Output* - quando um pré-processador ou um modelo é accionado por um pacote malicioso, é gerado um alerta e esses pacotes são armazenados pelo método estipulado no ficheiro de configuração [7].

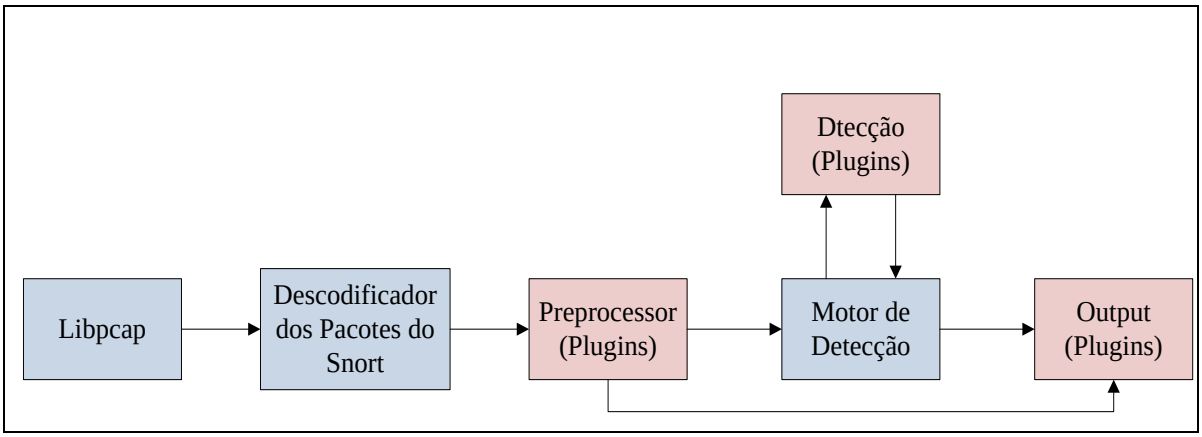

Figura 5. 1 - Processamento dos pacotes pelo *Snort* [11].

5.3.1 As regras do *Snort*

Uma regra do *snort* corresponde a um grupo de instruções, que são utilizadas para comparar pacotes de tráfego com os modelos armazenados, originando depois uma determinada acção [11]. O *Snort* utiliza uma linguagem simples para a descrição das regras estando estas divididas em duas partes: o cabeçalho e as opções (figura 5.2).
O cabeçalho da regra é constituído por quatro elementos [7]:

Acção - indica ao *snort* o que fazer quando um pacote que verifica a condição da regra é encontrado. Existem cinco opções disponíveis [16]:

- *alert* - gera um alerta utilizando o método seleccionado e de seguida armazena o pacote;
- *log* - armazena o pacote;
- *pass* - ignora o pacote. Esta opção é muito utilizada em falsos alarmes;
- *activate* - esta acção funciona com a acção *dynamic.* Alerta e executa uma regra *dynamic*;
- *dynamic* - permite, em certas circunstâncias, activar um segundo nível de processamento através da espera de um sinal de uma regra, por exemplo *activate,* para actuar.

  A utilização do par *activate/dynamic* é ideal para encontrar ataques mais complexos que podiam não ser descobertos. Este par não é utilizado habitualmente em configurações normais, mas apenas em configurações de IDS mais avançadas [7].

Protocolo - o *snort,* actualmente, só suporta quatro protocolos: IP, ICMP, TCP e UDP.

Endereço de origem - é o endereço de origem do pacote enviado. Pode ser um único endereço IP ou uma sub-rede, logo que a notação CIDR (Classless Inter-Domain Routing) seja cumprida [7].

Porta de origem - porta ou intervalo de portas, utilizado para o envio dos pacotes.

Endereço de destino - endereço ou grupo de endereços IP para onde o pacote é enviado.

Porta de destino - porta ou grupo de portas para onde os pacotes são enviados.

Operadores de direcção - existem dois operadores: o "->" que considera o endereço da esquerda a origem e o da direita o destino e também o operador "<>" que considera os pacotes bidireccionais onde a esquerda e a direita são consideradas destino e origem [16].

Conteúdo a procurar (content) - conteúdo do pacote a comparar com os modelos de ataque estabelecidos.

Mensagem de alerta (msg) - mensagem que será armazenada com o pacote ou com o alerta.

Na descrição anterior só foram exemplificadas algumas opções, mas existem outras que podem ser utilizadas nas regras do *snort*, como logto, ttl, sid, etc. [7] [16]

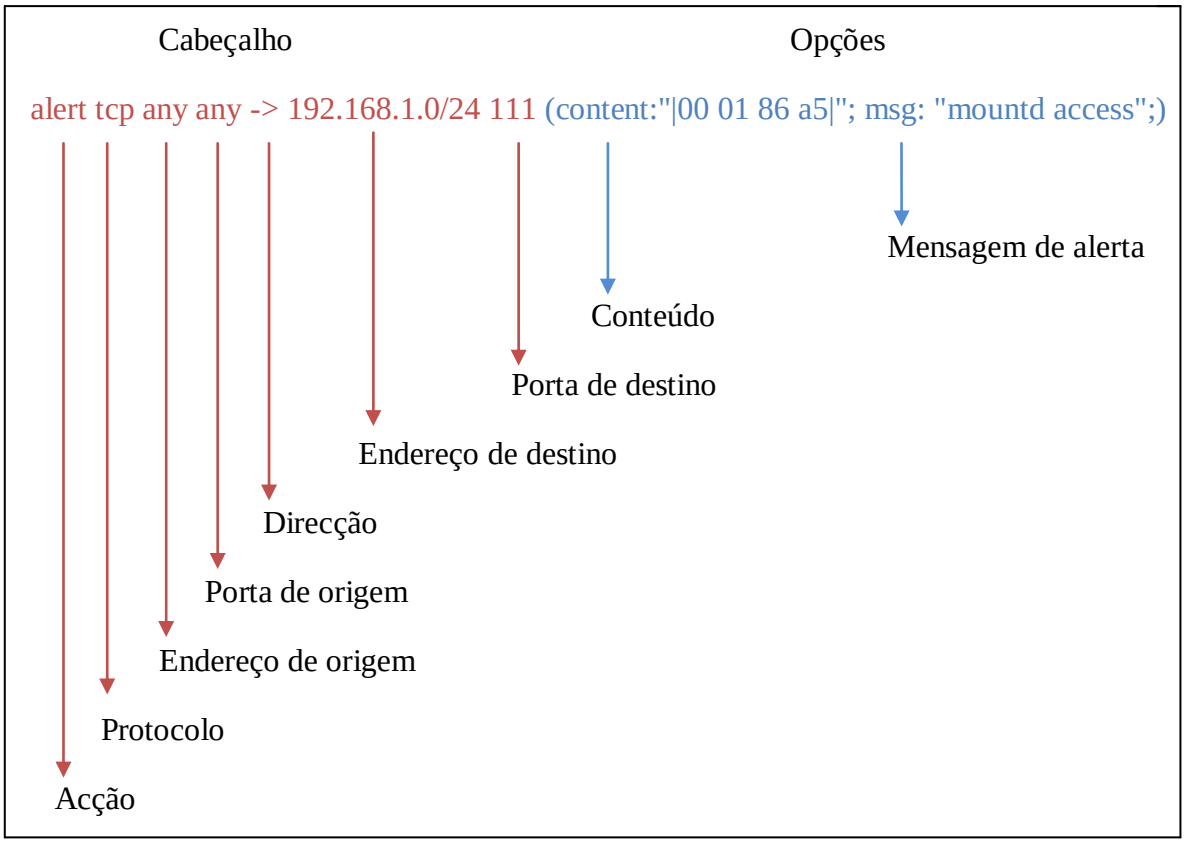

Figura 5. 2 - Exemplo das Regra de *Snort*

5.4 A instalação dos componentes do IDS de teste

Um dos objectivos iniciais desta tese era fazer uma actualização e melhoramento ao sistema actual de IDS da rede. Depois de um período de estudo sobre a rede e seus sistemas de segurança, optou-se por implementar um IDS com um servidor/sensor (*Snort* e *MySQL*) e um sensor. O equipamento utilizado para servir de IDS, tanto servidor/sensor como sensor, foi um computador Pentium III que como equipamento de teste funcionou bastante bem, considerando a quantidade de informação que tinha de processar para a monitorização e o armazenamento dos alertas. O servidor/sensor foi colocado a monitorizar a rede 84 enquanto o sensor ficou a monitorizar o tráfego da rede 14.

5.4.1 O servidor /sensor IDS de teste

Nesta máquina foi instalado o Sistema Operativo Suse v10.3 com a componente gráfica. Mesmo sabendo que o IDS podia ficar mais vulnerável, foi necessário instalar a componente gráfica para ser possível usar a ferramenta BASE. Para que o sistema ficasse o mais actualizado possível, as ferramentas necessárias não foram instaladas com o sistema operativo tendo-o sido posteriormente nas suas versões mais actualizadas. Para o servidor/sensor ficar a funcionar bastou fazer os *downloads* das versões mais recentes e seguidamente proceder à sua instalação.

As ferramentas instaladas foram:

- Mysql v5.0.51
- Apache v2.2.8
- PHP v5.2.6
- ADODB v4.98

Nota: Como os servidores do Apache e Mysql têm de ser instalados antes do PHP, é recomendado fazê-lo na sequência acima indicada. Tanto o servidor Mysql como o Apache foram preparados para se auto reiniciar com o ligar da máquina.

- Libpcap v0.98
- Pcre v7.7
- Snort v2.8.1
- Snort rules current

Nota: Como as regras do Snort já não são distribuídas com o mesmo, foi necessário fazer um registo para ser possível o seu *download*. Depois é necessário copiar também os novos ficheiros *mapping* e *signatures* obtidos com as novas regras para a pasta do Snort.

- Barnyard v0.2.0

Nota: Para possibilitar ao Snort maior eficácia na sua função, foi utilizado o Barnyard para as funções de envio de dados para o armazenamento no Mysql. Esta função teve de ser activada no ficheiro de configuração do Snort. Foi utilizado um script para fazer com que o Snort e o Barnyard pudessem começar automaticamente com o reiniciar [Apêndice A]. Ter as ferramentas necessárias com a possibilidade de se auto iniciar com o reiniciar da máquina é muito importante porque se, por exemplo, houver um corte na corrente eléctrica o IDS pode recomeçar sem ajuda humana.

- BASE v1.4.0
- Image Graph v0.7.2
- Numbers Roman v1.2.0

- Numbers Words v0.15.0

Nota: Para se poder visualizar e obter uma melhor compreensão dos alertas e dos registos armazenados é necessário utilizar um GUI. Para tal foi decida a utilização do BASE que permite visualizar e analisar os alertas recebidas. As restantes componentes são bibliotecas que permitem a utilização da função gráfica da BASE.

As ferramentas instaladas no servidor/sensor e no segundo sensor foram-no através de compilação de *tarballs* e não do uso de RPM pré-compilados, o que permite verificar os erros da compilação e igualmente a instalação fica mais apropriada a cada máquina.

5.4.2 O sensor de teste

No segundo sensor instalou-se o sistema operativo Suse v8.3 (foi instalada uma versão inferior à da primeira máquina porque o *hardware* assim obrigou) e como será apenas utilizada como sensor não será necessária a instalação das componentes gráficas. O *software* instalado foi:

- Mysql v5.0.51
- Libpcap v0.98
- Snort v2.8.1
- Barnyard v0.2.0

O Mysql foi instalado neste sensor por duas razões: em primeiro lugar porque é necessário ter o código fonte instalado na máquina para que o servidor /sensor consiga aceder remotamente ao IDS e em segundo lugar é de boa filosofia ter uma base de dados no sensor para armazenar os alertas caso o servidor seja comprometido. Infelizmente quando se está a armazenar os alertas em dois locais, o processo torna-se muito intensivo para a máquina podendo haver perda de recursos para a monitorização do tráfego. Para que esta situação não se verifique, foi instalado o Barnyard e escolhido o *output* do *Snort* em binário.

Foi também necessária a colocação de um NIC adicional para se poderem enviar os dados para o servidor.

5.4.3 Localização

A localização do IDS é de vital importância. Se colocado antes de um *firewall* visualiza todos os ataques à rede, mas esta situação torna-se bastante fatigante para o IDS e para administrador, porque recebe muitos alertas e não há necessidade de desperdiçar recursos a provar que a Internet tem, como já se sabe, tráfego maléfico. Como o verdadeiro custo da implementação de um IDS está mais no tempo de trabalho dispendido pelo administrador (todas as ferramentas necessárias são *open source*), seria ideal não desperdiçar muito tempo a verificar alertas falsas da rede. Assim sendo, se o IDS estiver colocado depois do *firewall*,

este já pode bloquear algum do tráfego maléfico, o que simplifica o trabalho do IDS e do administrador permitindo maior atenção à rede interna.

Na situação em análise foi decidido monitorizar a rede 84 através de um *span port* com o servidor/sensor. Esta decisão foi tomada por várias razões: primeiro, porque era necessário analisar a rede e em segundo lugar, como o IDS ainda estava em fase experimental, esta rede possibilitava um acesso fácil caso houvesse necessidade de fazer alterações no IDS. Além disso foi uma oportunidade para estudar o tipo de tráfego que nela circula, os alertas obtidos e suas possíveis consequências. Foi ainda possível criar e testar regras específicas para a mesma.

Verificado o sucesso no servidor/sensor, foi implementado um novo IDS para funcionar como um sensor e monitorizar o tráfego do servidor *November* através de um *span port.*

5.5 A implementação do *honeypot* de teste

Para a implementação do *honeypot* tiveram de ser considerados vários factores importantes, tais como o equipamento disponível e o nível de interactividade do *honeypot*. Com base nesses factores foi decidido utilizar a ferramenta *honeyd,* por ser uma ferramenta *open source* e de baixa interactividade evitando assim qualquer possibilidade de um intruso vir a utilizar o *honeypot* para seu uso pessoal. O *honeyd* permite configurar uma rede virtual com o equipamento e o sistema operativo que se desejar sendo também possível emular serviços como por exemplo *backdoors* e *telnet.* A interacção dos *crackers* com o *honeypot* está limitada à camada de rede. Deste modo, em vez de se ter um sistema operativo completo, é somente necessário simular a camada de rede. Este aspecto tem a desvantagem de o *cracker* nunca conseguir obter o acesso total ao sistema e a informação obtida ser limitada. Esta ferramenta consegue simular serviços de TCP e UDP e também responde correctamente às mensagens ICMP.

Para poder simular redes reais, o *honeyd* cria redes virtuais que são topologias arbitrárias de *routing* incluindo alguns dos factores configuráveis que podem ser encontrados em redes reais como a latência e a perda de pacotes. Quando são utilizadas ferramentas de sondagem na rede virtual, estas apenas poderão visualizar as topologias da rede simulada.

5.5.1 Funcionamento do *honeyd*

A arquitectura do *honeyd* (figura 5.3 [14]) é constituída pelos seguintes componentes:

- Base dados da configuração;
- Expedidor de pacotes;
- Manipulador de protocolos;
- Motor de personalidades;
- Topologia de *routing.*

Os pacotes recebidos são processados pelo expedidor de pacotes que verifica o tamanho e a existência de qualquer erro. O programa reconhece três tipos de protocolos: TCP, UDP e ICMP. Qualquer outro tipo de protocolo é armazenado e rejeitado. Antes de efectuar o processamento de qualquer pacote, o expedidor de pacotes questiona a base de dados da configuração sobre a existência de um *honeypot* que corresponda ao endereço IP em questão. Se ele existir, o pacote será encaminhado para um manipulador específico desse protocolo. No caso do protocolo ICMP, o manipulador de protocolos suporta a maioria dos pedidos conhecidos. Por defeito, todas as configurações respondem aos pedidos *echo* com a mensagem *destination unreachable.* A resposta a outros pedidos depende das configurações. No caso do TCP e UDP pode haver ligações a serviços suplementares. Estes serviços são aplicações externas, como por exemplo a simulação de um *backdoor* do vírus *myDoom.* Eles recebem os dados no *stdin* e enviam um *output* pelo *stdout*. O comportamento do serviço está dependente do tipo da aplicação. Caso seja recebido um pacote UDP direccionado a uma porta fechada, o *honeyd* enviará uma mensagem em ICMP de *port unreachable.* Antes do pacote ser enviado à rede, é processado pelo motor de personalidades. O motor de personalidades faz com que a pilha protocolar do *honeypot* actue como a personalidade configurada, introduzindo alterações nos cabeçalhos dos pacotes para que correspondam à personalidade. Este ajusta o conteúdo do pacote para que simule um pacote do sistema operativo configurado. É utilizada a base de dados do *Nmap* e *Xprobe* como referência para as personalidades dos pacotes do TCP e ICMP, respectivamente.

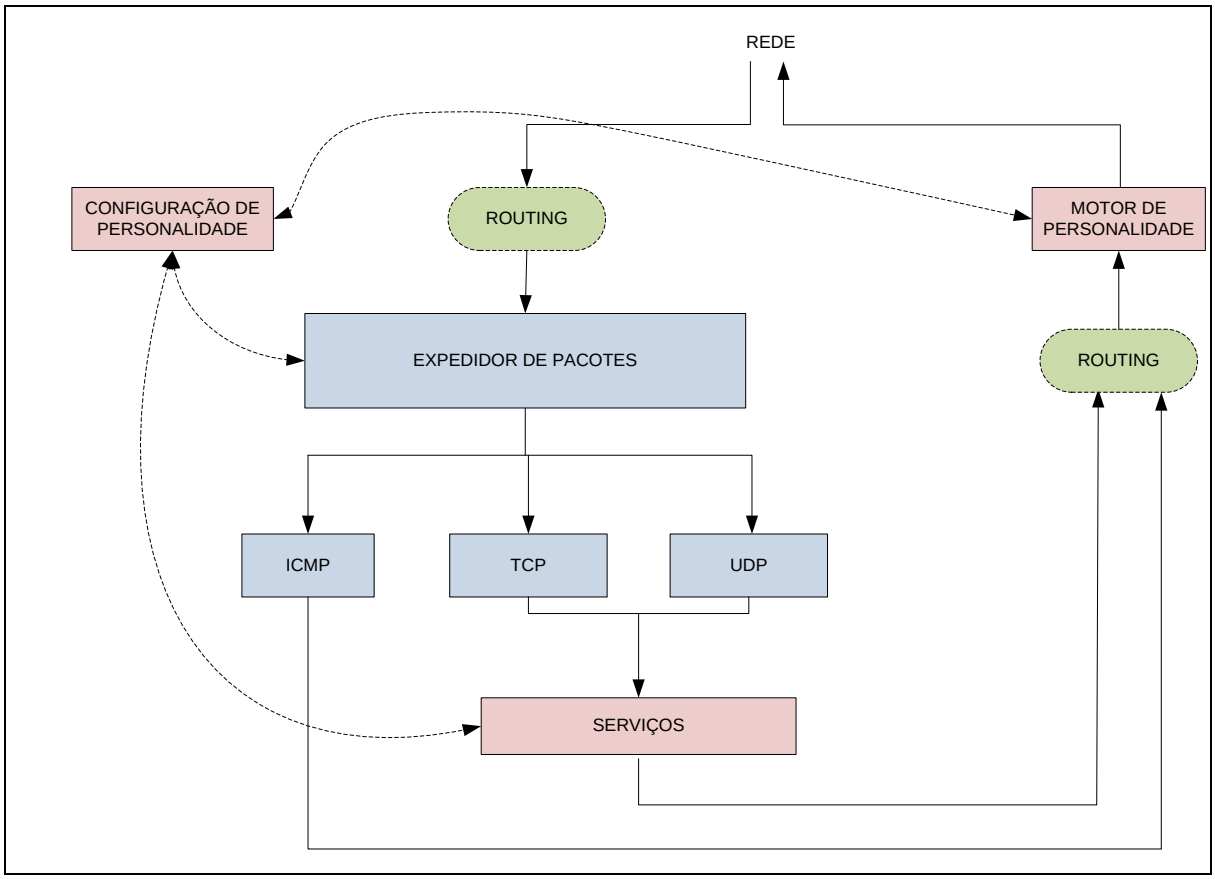

Figura 5. 3 - Arquitectura do honeyd [14]

Para a configuração do *honeyd* foi utilizada uma terceira máquina onde se instalou o Sistema Operativo Suse v8.3 com a componente gráfica. Foi Igualmente necessário instalar as dependências:

- libevent v1.4.6;
- libdnet v1.11;
- libpcap v0.98.

De seguida instalou-se a ferramenta com os serviços:

- Honeyd v1.5c;
- iisemultor v0.95 - emulador do produto IIS (*Internet Information Server*) da Microsoft;
- Telnet - emulador do telnet nos sistemas operativos Linux, Solaris e Windows;
- Ftp - emulador de um servidor ftp;
- PoP - emulador de um servidor POP3;
- Kuang2 ;
- Mydoom;
  Os emuladores *kuang2* e *Mydoom* conseguem simular os *backdoors* instalados pelos vírus e armazenam também quaisquer comandos utilizados em tentativas de uso do *backdoor;*
- Arpd v0.2.7.
  O Arpd é um daemon que está à escuta dos pedidos ARP e responde por qualquer IP não atribuído e em conjunto com o *honeyd* aloca os endereços de IP's não utilizados para o *honeypot* virtual.

Nota: tal como no IDS, todas as dependências e ferramentas foram instaladas através de compilação de *tarballs* e não com o uso dos RPM pré-compilados. É aconselhável desligar o IP *forwarding* onde o *honeyd* reside, evitando-se a duplicação de pacotes e possíveis avalanches dos mesmos na rede.

5.5.2 A rede do *honeypot*

A rede implementada do *honeyd* é constituída pelos sistemas operativos e serviços apresentados na tabela 5.1.

IP	**Sistema Operativo**	**Serviços Emulados**		**Portas abertas**	
		Tipo	**Porta TCP**	**TCP**	**UDP**
192.168.164.51 **192.168.165.51**	Microsoft Windows 2000 SP2	iisemul mydoom	80 1080	137 139	135 137

192.168.164.52 **192.168.165.52**	Linux v2.2.13	Apache kuang	80 4444	137 139	135 137
192.168.164.53 **192.168.165.53**	Microsoft Windows XP SP2 (com firewall)	iisemul mydoom	80 1080	137 139	135 137
192.168.164.54 **192.168.165.54**	Microsoft Windows XP Pro	iisemul mydoom pop3	80 1080 110	137 139	135 137
Router 192.168.84.47 **Router 192.168.165.2**	Cisco IOS v11.3 – 12.0	Telnet	23		

Tabela 5. 1 - Sistemas operativos e serviços utilizados no honeypot.

A topologia da rede do *honeypot*, figura 5.4, é constituída pelas redes 164 e 165, que realmente existem no ASC e não estavam a ser utilizadas de momento. Uma das desvantagens do *honeyd* é necessitar de IPs reais para a rede virtual funcionar. Como as redes reais não são livres de erros e atrasos nos pacotes, foi necessário aproximar o mais possível a rede virtual da realidade através da configuração (Apêndice B). Para tal foi utilizado no *latency* um valor de 50 ms, para simular o possível atraso existente em cada salto, no *loss* um valor de 0,1%, para simular a possível perda de pacotes e no *bandwidth* um valor de 1 Mbps para simular a largura de banda.

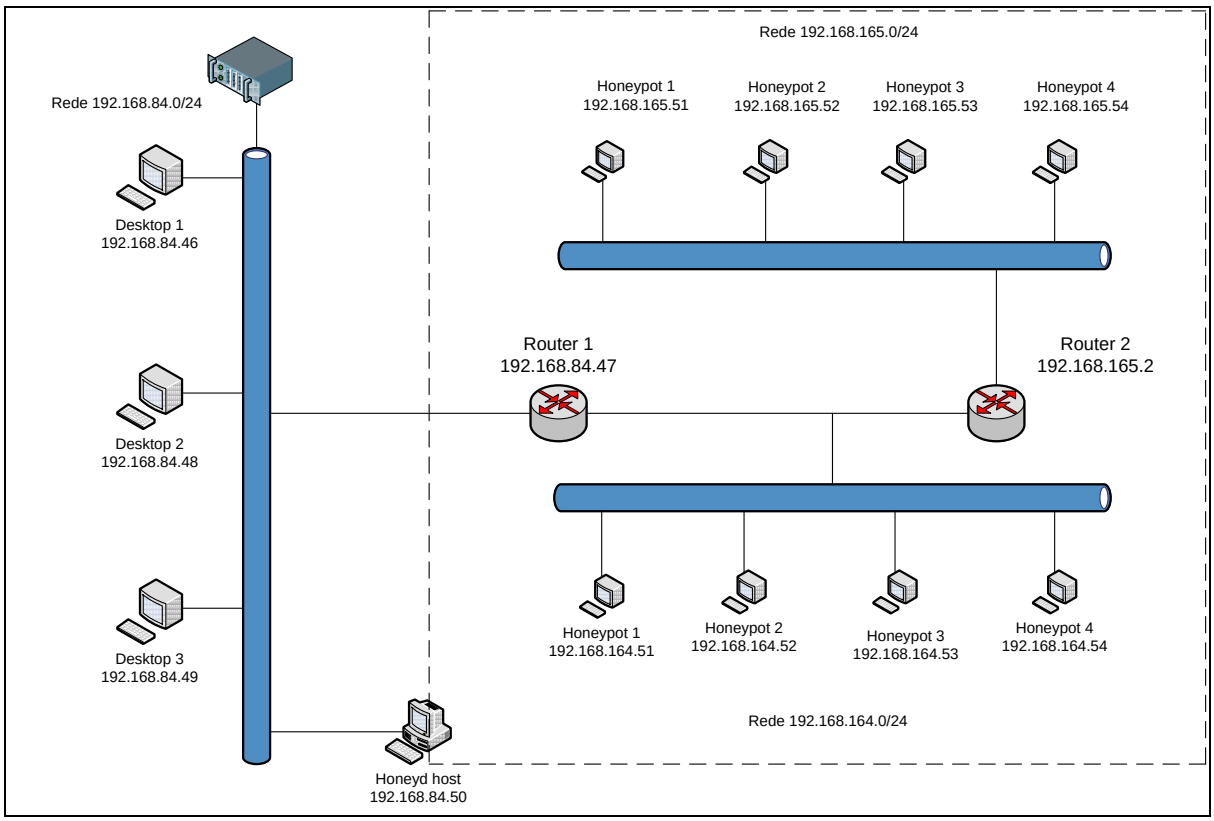

Figura 5. 4 - O *Honeypot* criado através do *honeyd*.

5.6 TAP passiva

Uma TAP é uma porta de teste de monitorização do tráfego não invasiva. Existem basicamente dois tipos de TAP: o primeiro utiliza uma tecnologia parecida com a de um *switch* fazendo uma cópia do tráfego e depois regenerando-o para o IDS para este o poder monitorizar. Este equipamento tem a desvantagem do seu preço ser elevado. O segundo, faz uma divisão do sinal eléctrico não sendo necessário equipamento complexo, pelo que é uma solução mais económica.

Como o ASC ainda não tinha disponível as TAP´s, decidiu-se construir uma de divisão do sinal eléctrico. Para tal bastaram 4 interfaces RJ-45 e ligar os pares de fios entrelaçados, como mostra a figura 5.5. Para evitar erros é necessário ter um cuidado extremo com os pares de fios entrelaçados, os quais se devem levar o mais longe possível unidos e fazer as ligações entre interfaces tão curtas quanto possível. Com a utilização deste tipo de TAP é necessário ter em mente que o tráfego da TAP A e da TAP B será em *half-duplex* e, daí, o IDS precisar de duas NIC´s para poder visualizar todo o tráfego da rede. Será também necessária a utilização de uma ferramenta extra para fazer a reunião de todo o tráfego e se tal não for possível, basta colocar o IDS a monitorizar os dois NIC´s. O produto final é constituído por quatro interfaces: a primeira e a última (HOST 1 e HOST 2) correspondem à entrada e saída do tráfego de rede enquanto que as TAP A e B são interfaces para a monitorização do tráfego. Como só estão ligados às interfaces da TAP A e TAP B os fios de recepção, o IDS não será visível por qualquer máquina da rede. Esta TAP não teve qualquer influência no tráfego da rede não tendo provocado atrasos ou perdas de pacotes. A sua utilização mostrou ser de grande utilidade para um administrador que necessite de verificar qualquer ponto da rede.

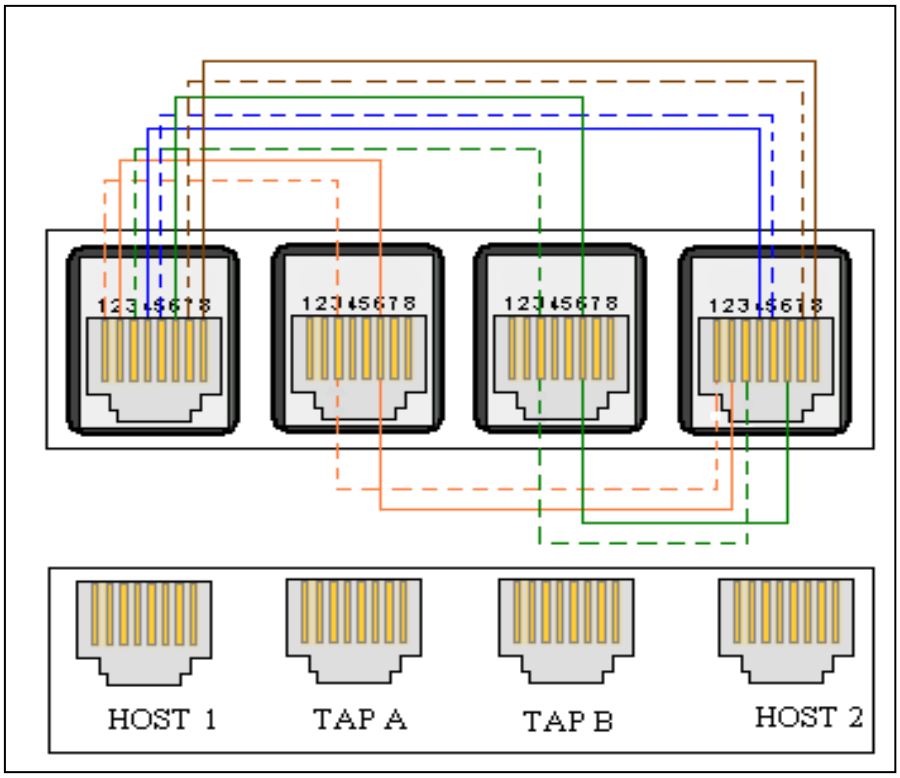

Figura 5. 5 - TAP passiva de Ethernet

5.7 Testes realizados

Depois de se ter instalado e configurado o IDS e o *honeyd* era necessário verificar o seu correcto funcionamento. No caso dos IDS´s havia que verificar o funcionamento das regras, especialmente as criadas, sendo também preciso testar se haveria queda de pacotes em excesso e se o endurecimento dos IDS´s era suficiente. Para o *honeyd,* era necessário verificar se a rede virtual estava em funcionamento com todos os serviços que foram instalados.

5.7.1 Testes dos IDS´s

Para a realização destes testes foram utilizados dois programas: *Nmap* v4.68 e *Nessus* v3.2.0. Pode argumentar-se que os programas são equivalentes. No entanto durante os testes verificaram-se pequenas diferenças nos resultados destes programas. Em todos os testes realizados aos IDS´s nunca foi opção utilizar testes que pudessem colocar a rede em perigo, tais como DoS. Como o *snort* conseguio detectar possível tráfego maléfico da rede e, também se verificou alertas dos vários testes realizados pelo *Nessus* e *nmap*, concluiu-se que as regras dos IDS´s estavam a funcionar correctamente, não havendo necessidade de utilizar qualquer programa extra, como *mucus* ou *sneeze,* para a verificação das mesmas. A maioria dos ataques detectados pelo snort era de origem TCP e UDP. Houve também vários ataques registados à porta 80 e muitos *scans* realizados às portas. Verificou-se que o servidor/sensor obteve mais registos que o sensor por aquele ter mais ferramentas em funcionamento.

Nota: antes de realizar os testes houve o cuidado de endurecer no melhor possível o IDS.

5.7.2 Teste do *honeypot*

Na realização do teste ao honeypot verificou-se o seu correcto funcionamento incluindo os serviços adicionais, como as portas abertas pelos vírus. O teste foi realizado a toda a rede virtual, a figura 5.6 mostra apenas o realizado ao IP 192.168.165.51. Detectaram-se cinco portas abertas de baixo risco (3) sendo uma relativa à porta 80 e as quatro seguintes às portas 137 TCP, 139 TCP, 135 UDP e 137 UDP (2). Foi igualmente detectada uma situação que apresentava um risco elevado (1), correspondendo à porta simulada pelo *mydoom.* O *traceroute* (4) apresentou os saltos correctos, mas infelizmente o *Nessus* não conseguiu a informação certa sobre o sistema operativo, erro devido à base de dados para as personalidades do *honeyd* que não é totalmente compatível com o *Nessus.* O teste feito posteriormente pelo *Nmap* revelou melhores resultados a respeito do sistema operativo.

Figura 5. 6 - Resultados obtidos pelo Nessus no teste de uma máquina da rede virtual.

A figura 5.7 mostra um pequeno exerto do teste realizado ao *router* com o IP 192.168.165.2. As portas simuladas foram correctamente visualizadas pelo *Nessus* assim como também a simulação do serviço telnet (5). Também o *traceroute* mostrou estar correcto (6), mas neste teste o sistema operativo utilizado no *router* (7) foi identificado correctamente pelo *Nessus* e o *Nmap.*

Figura 5. 7- Resultados obtidos pelo Nessus no teste ao router virtual.

5.7.3 Teste da TAP

Para verificar o correcto funcionamento da TAP, utilizou-se primeiramente o *Wireshark* (versão 1.0) e depois confirmou-se se realmente ela estava oculta na rede. Como para a utilização da TAP eram necessárias duas NIC´s para fazer a captura de todos os pacotes e, como o *Wireshark* não pode fazer essa captura nas duas interfaces simultaneamente, foi necessário executar duas sessões e no final reunir os pacotes recebidos e constatar se houve alguma perda de pacotes. A ocultação do IDS através da TAP na rede revelou-se funcional não só por ser impossível fazer um *ping* ao IDS mas também porque as ferramentas de teste não tiveram sucesso na realização de qualquer tipo de teste.

6. O Sistema de Segurança implementado para a rede do ASC

6.1 O IDS implementado para a rede do ASC

Depois de se ter implementado e testado com sucesso o IDS e o *honeypot* de teste, decidiu-se fazer algumas alterações ao sistema inicial e implementar um sistema de segurança mais adequado à rede do ASC. Continuou a utilizar-se um sistema baseado num servidor/sensor e num sensor a monitorizar as redes, em locais distintos. A instalação do servidor/sensor foi feita num servidor *HP ProLiant DL380 G4 High Performance - Xeon 3.8 GHz*, foi utilizado o Suse v11.0 e as mesmas ferramentas do teste, simplesmente mais actualizadas. Em relação ao sensor, utilizou-se um computador *Pentium III* também com o sistema operativo Suse v11.0. Apenas houve alterações na quantidade de redes a monitorizar pelo IDS. O servidor/sensor, por ser um bom equipamento, tanto em processamento como em armazenamento, foi utilizado para monitorizar as redes 14 e 84. A implementação foi feita neste equipamento para que, no final deste estágio, o sensor/servidor pudesse continuar em função. O sensor foi implementado para detectar eventuais anomalias no tráfego de Lisboa e na rede do ISA e, desse modo, facilitou a resolução de alertas verificados nas referidas redes.

6.2 Localizaçao do sistema implementado

Para melhor alcançar os objectivos propostos, manteve-se a monitorização da rede principal (rede 14) e da rede operacional (rede 84) através do servidor/sensor localizado nos Serviços Administrativos e Técnicos (SAT), onde mais tarde o *honeypot* também veio a ser colocado. O sensor ficou localizado dentro da Aerogare e fez a monitorização do router (tráfego entre Porto e Lisboa), mas por mera precaução acabou monitorizando também o servidor *Internet Security and Acceleration* (ISA). A figura 6.1 mostra os pontos de ligação utilizados para o IDS.

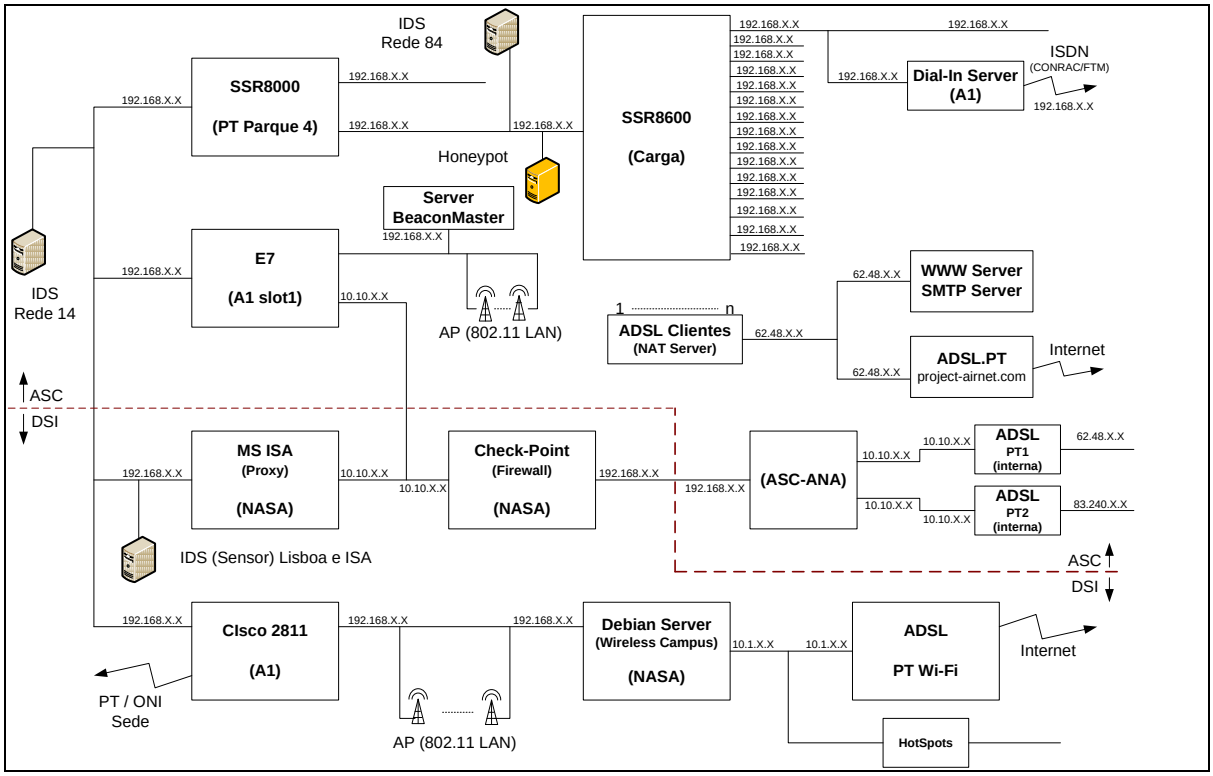

Figura 6. 1 - Localizaçao do IDS

6.3 Utilizaçao da TAP

Durante o estágio realizado no âmbito desta tese, o ASC adquiriu duas TAP´s: um portátil (ATAP GIG BT-BT) e outro fixo (FATAP-2000 BT/SX). Assim acresecentou-se aos objectivos iniciais a configuração e teste destes equipamentos. Nos testes iniciais, necessários para eliminar qualquer dúvida sobre o seu funcionamento, houve o cuidado de verificar se os *tags* das VLAN´s eram reconhecidos pelo equipamento. O TAP foi configurado e colocado a monitorizar temporariamente a VLAN 14 e a VLAN 84. A figura 6.2 mostra o programa necessário para configurar o FATAP-2000 BT/SX.

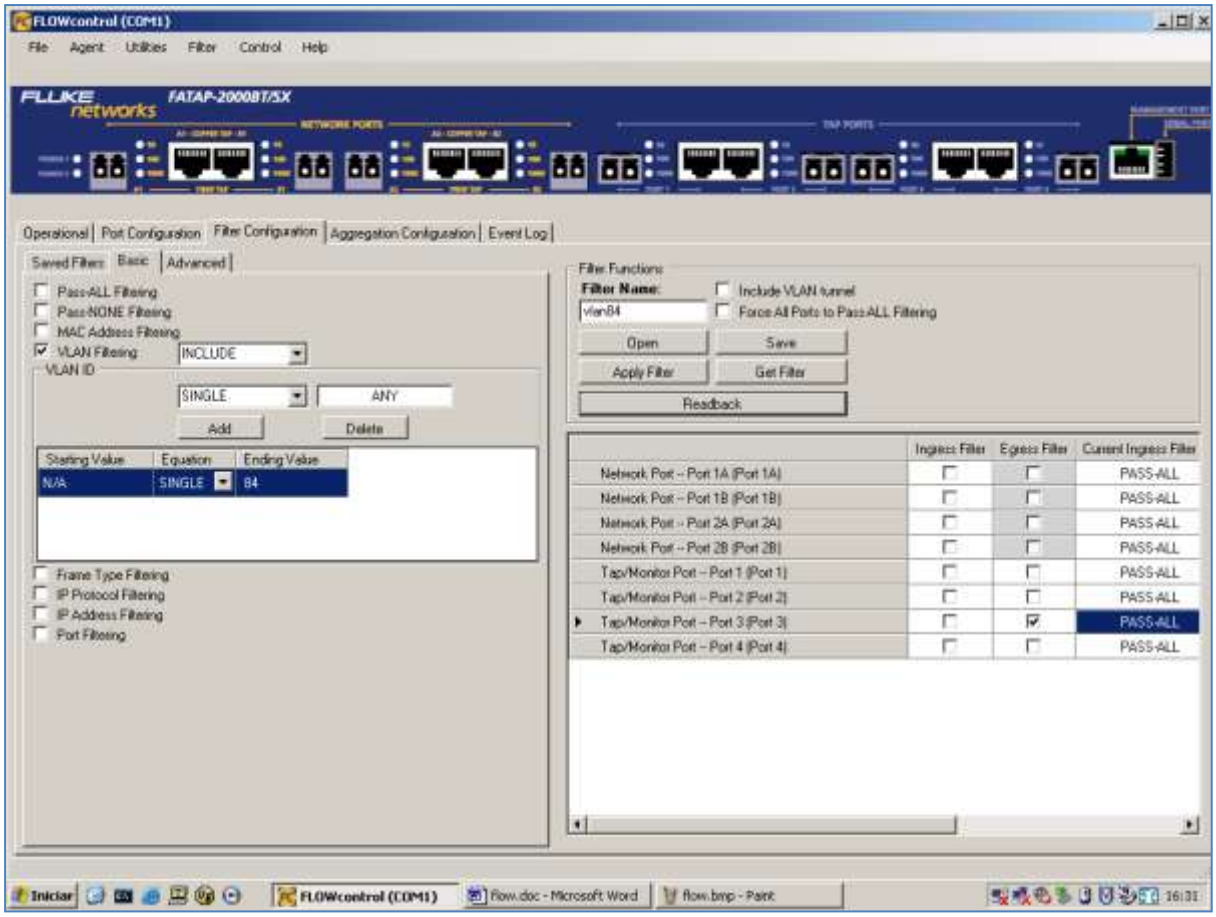

Figura 6. 2 - O FLOWcontrol - programa para a configuração da TAP

Foram escolhidas as redes 14 e 84 por ser nelas que se centrou o trabalho de estágio. A utilização da TAP permitiu obter melhor informação sobre todo o tráfego nestas redes, visto os pacotes serem analisados antes do *switch* o que permite detectar qualquer erro no pacote antes de ser descartado por este.

As ligações entre os *switchs* e a TAP foram feitas por fibra óptica em *link aggregation* e, entre o IDS e a TAP, utilizou-se um cabo CAT-5e. Houve a preocupação de saber a distância entre os *switchs* e a TAP, porque as ligações de *Gigabit interface converter* (GBIC) entre eles, eram SX (fibra multimodo) não sendo recomendadas para distâncias superior a 550 m.

Configurou-se a TAP com dois filtros, um para a rede 14 e outro para a rede 84. Estes funcionavam apenas para o tráfego copiado para as portas de monitorização, ficando o restante tráfego com entrada e saída livre. A figura 6.3 mostra os filtros activos nas portas de monitorização 1 e 3.

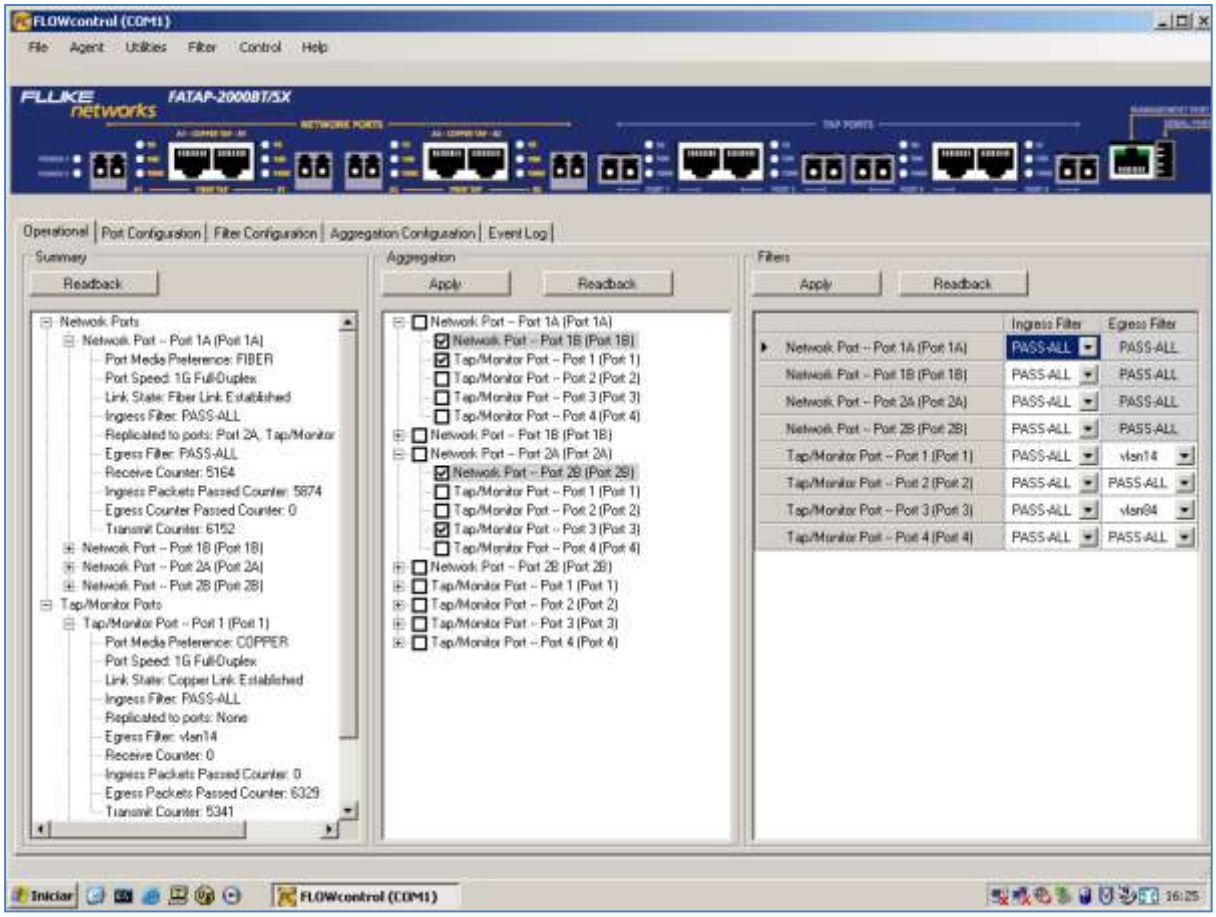

Figura 6. 3 - A configuração da TAP

6.4 Os alertas obtidos

A instalação de um IDS exige, inicialmente, um trabalho muito cuidado para eliminar o máximo de falsos alertas e libertar o administrador que deve concentrar a sua atenção nos alertas verdadeiros. Existem várias maneiras de conseguir reduzir os falsos alertas sendo a solução mais simples desactivar qualquer regra desnecessária para a rede. Uma boa configuração do IDS e a actualização de programas é a melhor segurança podendo reduzir muitos alertas. O administrador necessita sempre de saber a localização e o estado de todos os componentes da rede, tanto os novos como os velhos, visto que um equipamento esquecido, mas activo, poder provocar muitos problemas.

Quando se inicializou a fase da resolução dos alertas já havia aproximadamente 750 000 alertas das 4 redes, isto é do tráfego de Lisboa, rede ISA, VLAN 14, VLAN 84 e também da sua monitorização através da TAP. Uma vez que a VLAN 14 e a VLAN 84 já estavam ser monitorizadas há mais tempo que as outras, existem mais alertas desta proveniência como mostra o gráfico 6.4.

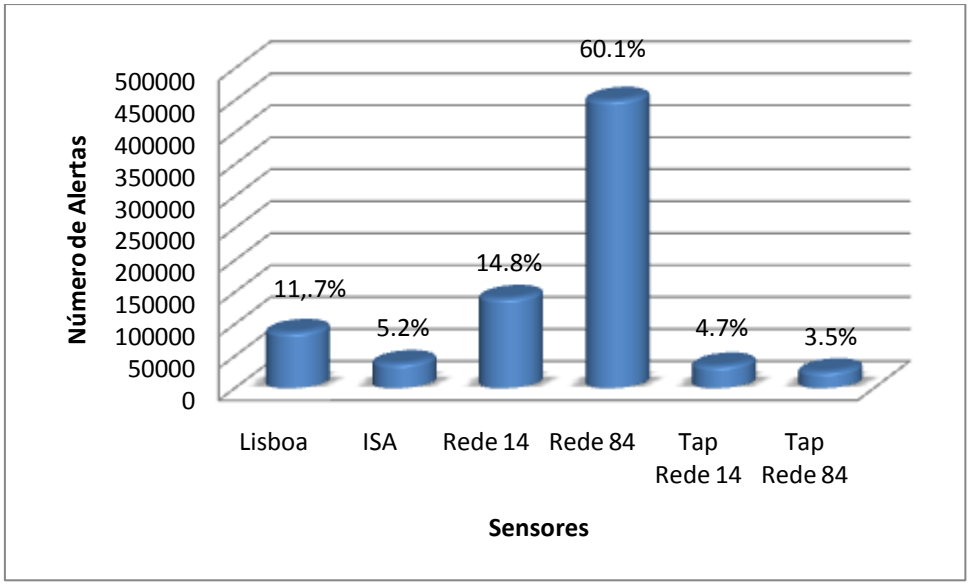

Figura 6. 4 - Número de alertas obtido nas redes

Como se pode ver pelo gráfico, a VLAN 84 é a que apresenta a maior quantidade de alertas, aproximadamente 450 000, mas apenas 16 eventos diferentes, sendo esta a que também obrigou a um maior esforço para os solucionar. A VLAN 14 tinha aproximadamente uns 180 000 alertas com 18 eventos diferentes. Todos os resultados acima referidos foram obtidos directamente do GUI do *BASE*, como mostra a figura 6.5.

Nota: A imagem 6.5 foi obtida depois de já se ter resolvido alguns alertas.

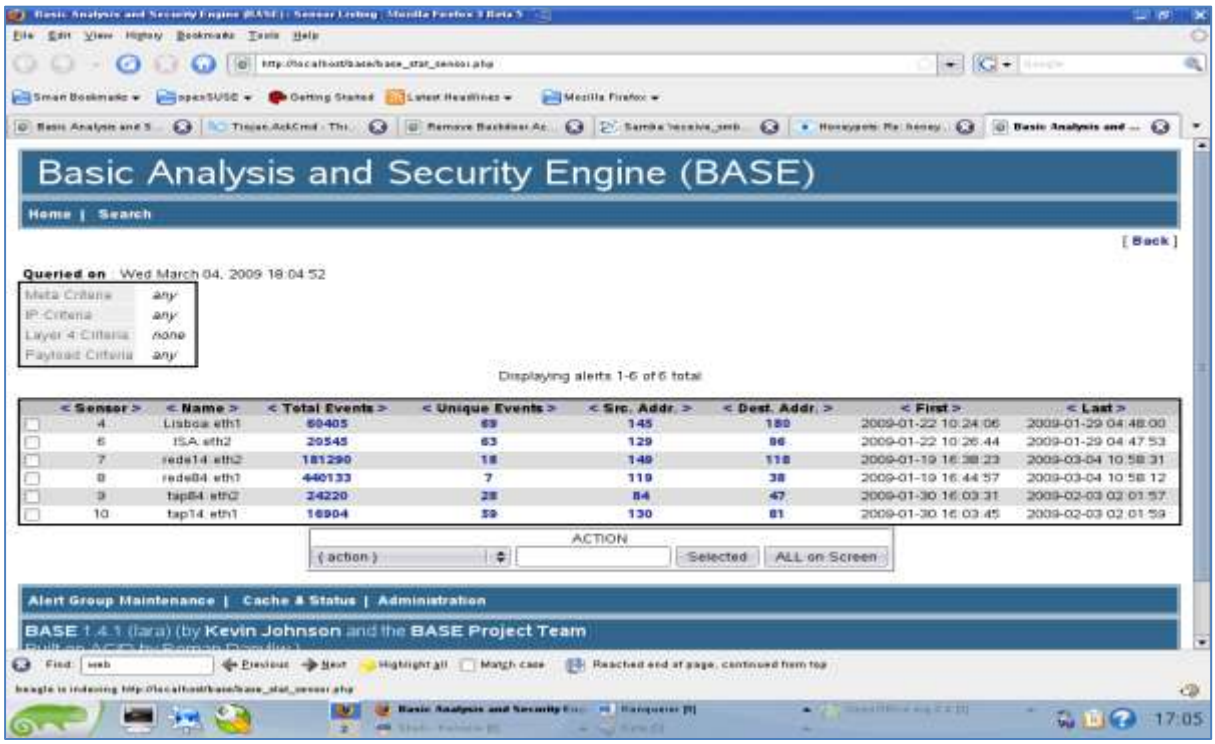

Figura 6. 5 - GUI do *BASE*

Os alertas recebidos estavam divididos em 13 categorias. O reconhecimento activo foi o que mais alertas obteve, como mostra o gráfico 6.6.

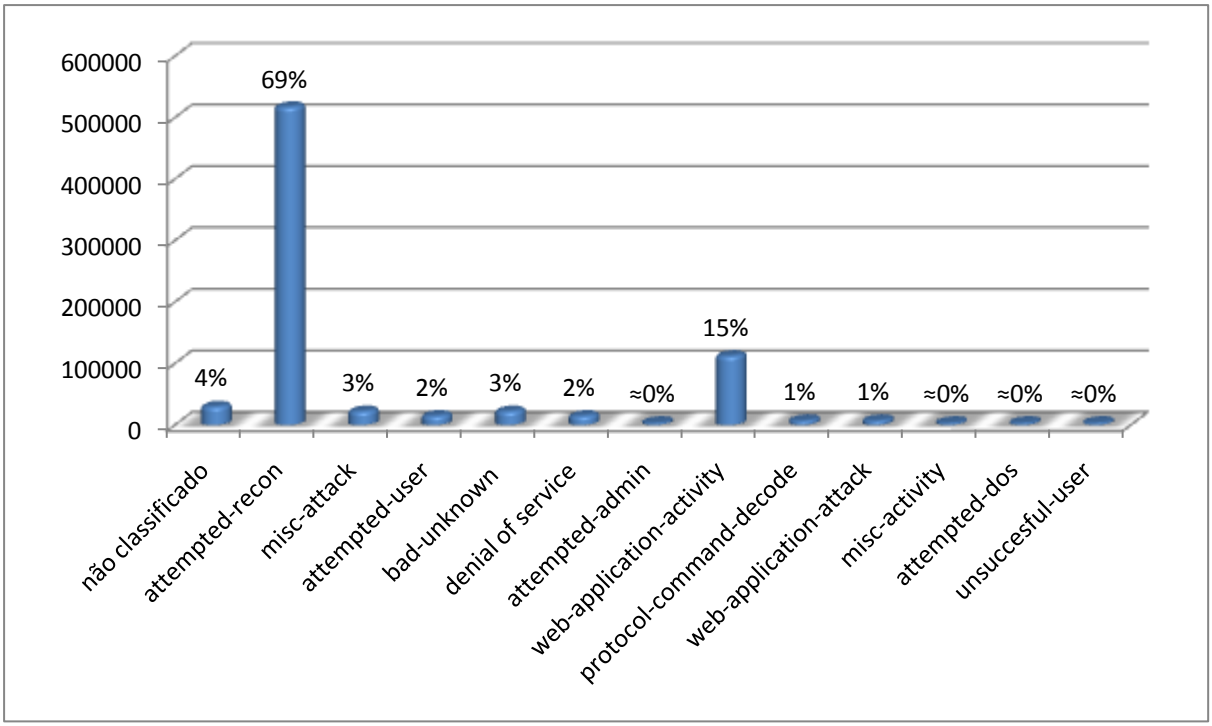

Figura 6. 6 - Número de alertas vs tipos de alertas

O IDS foi iniciado no dia 19 de Janeiro e, como se pode visualizar no gráfico 6.7, houve um aumento contínuo dos alertas. Aos fins-de-semana como era de esperar porque o número de utilizadores sofria um decréscimo os alertas diminuíam consideravelmente, como se pode verificar nos dias 24 e 25 de Janeiro.

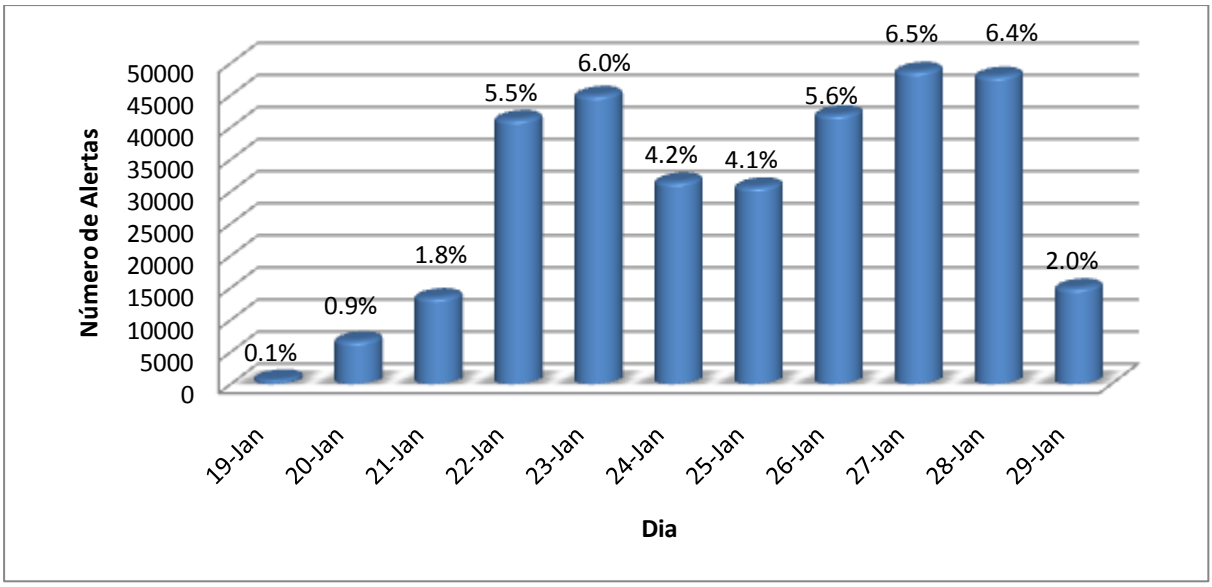

Figura 6. 7 - Número de alertas vs tempo (dias)

Para ajudar na administração dos alertas obtidos foi utilizado a ferramenta *cerebus*. A figura 6.8 mostra todos os alertas recebidos, a sua quantidade e também o seu nível de prioridade sendo possível verificar que o *Ping Nmap* e o *L3retriever Ping* são os que estão em maior quantidade. Antes de começar com o solucionamento dos alertas é necessário decidir quais a resolver primeiro. Neste caso foi decidido começar com os mais graves, já que podia haver uma tentativa de camuflagem das intrusões graves através de um bombardeamento das menos graves.

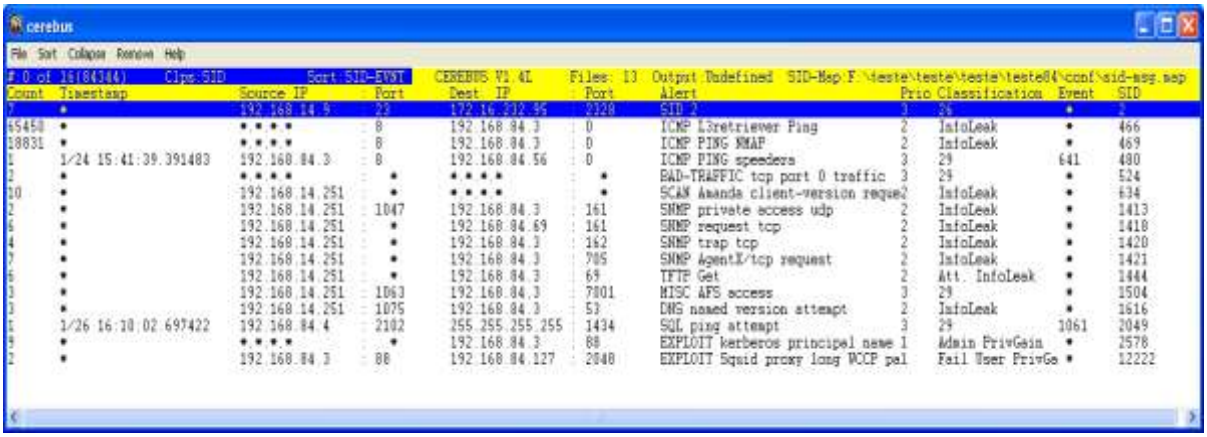

Figura 6. 8 - Visualizaçao dos diferentes tipos de alertas pelo *Cerebus*

Depois de se ter iniciado a fase de resolução dos alertas, houve uma clara diminuição dos mesmos como mostra o gráfico 6.9. Infelizmente a VLAN 84 representou-se como um enorme desafio isto porque esta rede apresentava o maior número de alertas de reconhecimento activo.

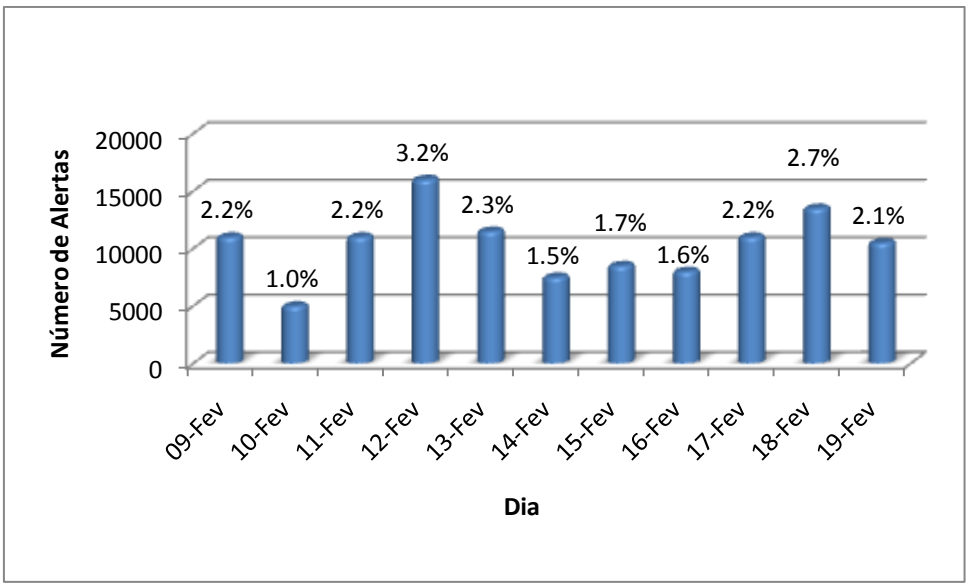

Figura 6. 9 - Número de alertas vs tempo (dias)

Como nenhuma rede está suficientemente segura, o IDS continuo a detectar vários alertas e será sempre necessário analisá-los e investigá-los todos para avaliar os possíveis impactos e riscos para a rede. O tempo, em alguns casos, é inimigo do administrador já que uma actuação rápida e apropriada pode evitar danos maiores.

A seguir listam-se alguns dos alertas mais frequentes da VLAN 14 e VLAN 84, acrescentando uma breve explicação dos mesmos e das acções tomadas.

6.4.1 VLAN 14

Depois de se ter iniciado o servidor/sensor e enquanto se colocava o sensor na rede e também se fazia testes no TAP, o IDS recebeu vários eventos muitos dos quais apenas só com um alerta. Como eram muitos eventos mas poucos alertas, decidiu-se verificar se eram prejudicial para a rede e depois simplesmente eliminá-los, para depois verificar se voltavam a aparecer.

Alerta: `SMTP ClamAV recipient command injection attempt`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	25

Este alerta foi excluído por várias razões: primeiro porque o endereço de origem corresponde a um servidor de Lisboa o que se considera como uma fonte fidedigna; segundo porque foram muito poucos os alertas e também muito esporádicos e, terceiro, por não existir nenhum Clam anti-vírus instalado no servidor virtual, o endereço de destino. Como não representava perigo, o tráfego de origem da rede 1 foi filtrado nesta regra.

Alerta: `NetBios-DS repeated logon failure`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	445	192.168.X.X	X
192.168.X.X	445	192.168.X.X	X

Este alerta é gerado pelo servidor quando uma máquina falha o *login*, o que pode significar que um utilizador não autorizado esteja a fazer tentativas de várias palavras-chave para obter acesso. No entanto como o acesso às máquinas é limitado (seguranças à entrada dos edifícios) e como os utilizadores são obrigados a alterar as palavras-chave com periodicidade, considerou-se filtrar esta regra apenas para o tráfego da rede 14. Mas como existe sempre a possibilidade de haver ataques internos, optou-se, por manter a regra e apenas continuar atento à evolução dos alertas.

Alerta: Netbios SMB C$ Unicode share access

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	X
192.168.X.X	X	192.168.X.X	X
192.168.X.X	X	192.168.X.X 192.168.X.X	X

e

Alerta: Netbios SMB-DS C$ Unicode share access

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	X

Estes alertas são gerados quando há uma tentativa de acesso de recursos privados, neste caso a do ASCMAIL. Estes IP´s tinham permissão para aceder aos recursos, sejam eles da rede 1 de Lisboa ou no caso da rede 14 o 56 que corresponde à máquina da administração remoto da rede. Como tal a regra dos alertas foi alterada para filtrar os IP´s 192.168.X.X e também o 192.168.X.X e o 192.168.X.X. Verificou-se que depois de se ter aplicado a alteração à regra os alertas cessaram.

Alerta: MISC UPnP malformed advertisement

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X **192.168.X.X** **192.168.X.X** **192.168.X.X** **192.168.X.X** **192.168.X.X** **192.168.X.X** **192.168.X.X**	X	239.255.255.250	1900

O *Universal Plug and Play* (UPnP) é um serviço que permite às máquinas de uma rede local localizar e utilizar dispositivos na rede. Este serviço é normalmente utilizado com interfaces que escutam em *broadcast* o que permite ao *cracker* atacar vários sistemas sem ter

necessidade de conhecer os IP´s. Os sistemas vulneráveis a este tipo de ataque são o Microsoft Windows 98, 98 SE, ME e o XP. Estes alertas foram resolvidos desligando o serviço nos equipamentos. Neste caso os alertas foram detectados vindo de impressoras e de portáteis.

Alertas envolvendo ICMP´s

Se for comparado os alertas de ICMP recebidos nas duas VLAN´s, é possível verificar que a VLAN 84 obteve um número muito mais elevado. A figura 6.10 mostra os pacotes ICMP capturados pelo *wireshark* da VLAN 14 com a utilização de uma TAP durante um dia normal de utilização. Como era de esperar o servidor (192.168.X.X) estava muito mais activo que as máquinas. Nas máquinas veio a verificar-se registos ICMP´s com um intervalo de tempo de aproximadamente 20 minutos.

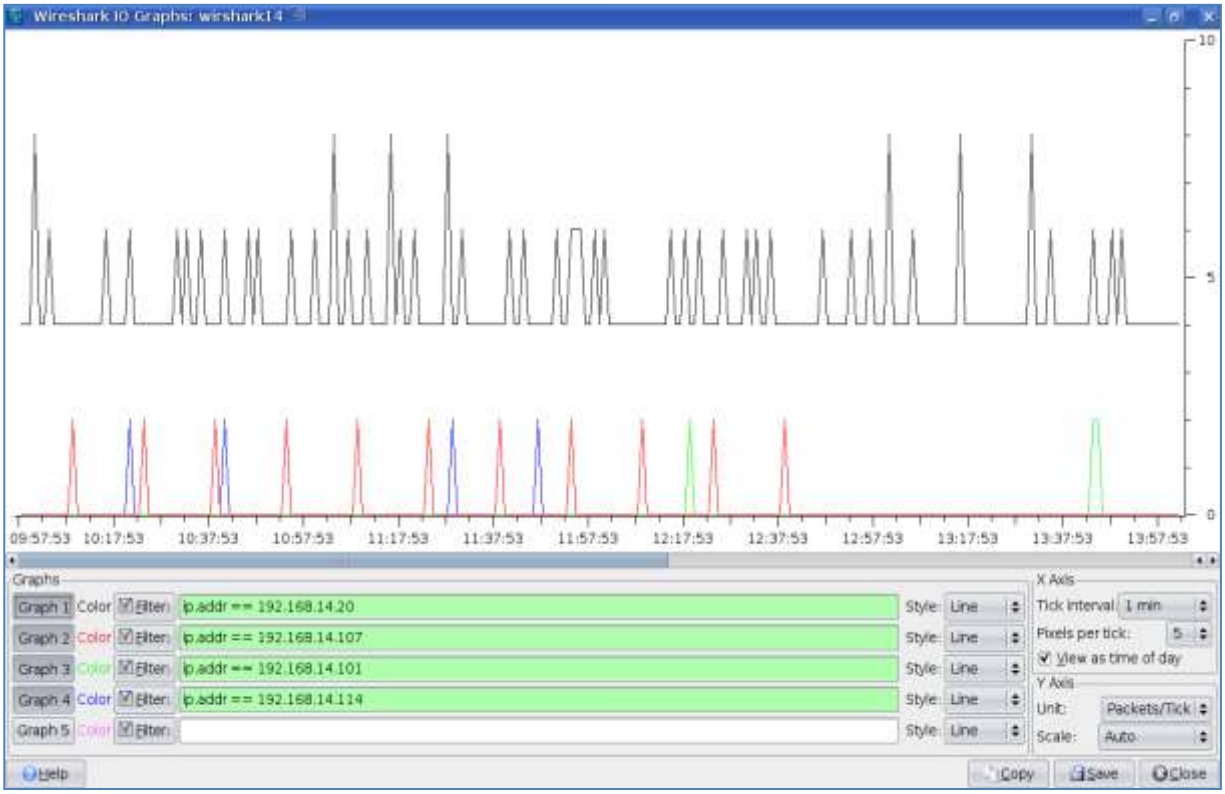

Figura 6. 10 - Número ICMP´s vs tempo (horas)

Alertas: `ICMP PING NMAP`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	8	192.168.X.X	0

A existência destes alertas implica que pode ter havido um *ping* gerado pelo *Nmap*, o que pode traduzir numa possível *scan* à rede. O tamanho dos dados de um *ping* do *Nmap* é de

valor zero sendo esta a única característica actual que o identifica. Infelizmente também há outras ferramentas a utilizar *pings* de ICMP de tamanho zero, como por exemplo o *Avast antivírus*. Este produz *pings* ICMP quando se está a ligar com os seus servidores para fazer uma actualização (que pode ocorrer cada 40 segundos caso não obtenha nenhuma resposta).

Como todos os IP´s de origem eram de máquinas fidedignas e também como a quantidade de ICMP´s por máquina não era elevada, optou-se por não alterar a regra. Isto porque este alerta pode anteceder um ataque. No entanto caso haja alertas do tipo *portsweeps* ou *portscans* após um alerta *ping Nmap,* exigia um melhor estudo dos IP´s envolvidos para verificar se houve um ataque.

Alerta: `ICMP L3retriever Ping`

Origem	***Porta***	***Destino***	***Porta***
192.168.X.X	8	192.168.X.X 192.168.X.X 192.168.X.X	0

Os alertas *L3retriever Ping* indicam que foi utilizado o *L3 Network security retriever* para fazer um *scan* à rede. O *scan* podia ter sido feito por um *cracker* para obter uma melhor percepção da rede, ou pela Symantec que adquiriu este *software* e utiliza-o em alguns dos seus produtos. Também se tem verificado que este alerta é gerado por sistemas que utilizam o Windows 2000 e XP, pois estes enviam demasiadas mensagens aos servidores.

Como não existe nenhum *L3 Network security retriever* instalado nas máquinas do ASC optou-se por utilizar o *wireshark* para visualizar os pacotes ICMP´s que podiam estar a gerar os alertas. Foi verificado que os pacotes em grande parte tinham uma periocidade de aproximadamente 20 minutos por máquina e que correspondiam a ARP *reply´s,* como a tabela de ARP é renovada com a mesma periocidade, decidiu-se manter a regra mas excluir a rede 14 e continuar atento à evolução dos alertas. Visto que foram observados muitos alertas deste tipo na VLAN 84 e, foram estes que lançaram a suspeita sobre possíveis problemas.

Alerta: `ICMP Redirect net`

Origem	***Porta***	***Destino***	***Porta***
192.168.X.X	5	192.168.X.X	0

Estes alertas verificam-se por existirem mensagens enviadas de um *router* para uma máquina, informando-a sobre o melhor percurso para o envio de pacotes através do mesmo. A informação é temporária e será apenas utilizada enquanto existir na cache. Esta funcionalidade existe para evitar que haja *routing tables* muito extensas. O endereço IP 192.168.X.X corresponde a um *router* que faz a ligação com Lisboa. Verificaram-se as definições do *gateway* e confirmou-se que a melhor solução é a utilização do IP 192.168.X.X

como *gateway*. Infelizmente esta solução não resolveu todos os alertas e foi necessário filtrar a rede 192.168.X.X na regra.

Alertas envolvendo o preprocessador http_inspect

O http_inspect é um preprocessador que faz a descodificação de http para as aplicações, normalizando e descobrindo os campos http num *buffer* de dados. Este preprocessador foi concebido para evitar que o IDS seja iludido por técnicas de evasão.

Por defeito, as configurações deste preprocessador fazem com que qualquer servidor seja monitorizado e que accione um alerta por tamanho excessivo sempre que qualquer caminho tenha mais que 500 bytes. Estes alertas, em excesso, significam que a máquina tem algum problema ou está a gerar ataques, verificou-se que os alertas eram gerados apenas na VLAN 14.

Alertas: `(http inspect) DOUBLE DECODING ATTACK`

Origem	***Porta***	***Destino***	***Porta***
192.168.X.X	X	192.168.X.X	80

Num ataque de *double encoding*, o *cracker* envia um pedido URI que tem caracteres que foram codificados duas vezes. O IDS detecta a primeira codificação e consegue normalizá-la, no entanto o *output* resulta numa *string* de caracteres ASCII ainda codificados. Normalmente este tipo de ataque tem como objectivo iludir um IDS e atacar um servidor *Web*, onde depois poderá lançar mais ataques sem ser notado. O pré-processador *http_inspect* do *snort* tem a opção de *double_encode* que, se escolhido, identifica este tipo de ataque. Os sistemas *Microsoft IIS Servers* são os únicos que são vulneráveis a este tipo de ataque.

Alerta: `(http_inspect) BARE BYTE UNICODE ENCODING`

Origem	***Porta***	***Destino***	***Porta***
192.168.X.X	X	192.168.X.X	80

e

Alerta: (http inspect) IIS UNICODE CODEPOINT ENCODING

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X **192.168.X.X** **192.168.X.X**	X	192.168.X.X	X
192.168.X.X	X	192.168.X.X	X

Estes alertas são parecidos com o anterior. O pré-processador *http_inspect* detecta tráfego que pode provocar ataques ou iludir um IDS por ter caracteres mal identificados. Os únicos sistemas afectados são os *Microsoft IIS Servers*. Quando o *snort* é instalado, é configurado, por defeito, para suportar uma página de código *unicode* com o valor 1252 que corresponde à linguagem americana e inglesa. Este valor funciona bem no *snort* com a língua inglesa, mas como existem línguas com caracteres não reconhecidos pelo inglês, quando as páginas com estes caracteres são processadas pelo *snort* geram alertas, que por vezes podem ser em grande quantidade e parecer um ataque. Este problema resolve-se em parte alterando o valor da configuração naquela página do código *unicode* para os valores próximos dessas línguas. Não é aconselhável eliminar por completo esta opção porque pode criar lacunas na segurança.

Alerta: (http_inspect) OVERSIZE REQUEST-URI

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	80

Estes alertas foram gerados porque o pré-processador *http_inspect* detectou um URL maior que o valor especificado no ficheiro de configuração. Pode ser mais um método utilizado para evadir um IDS. Uma solução possível é aumentar o valor especificado no ficheiro de configuração.

Para solucionar a maioria destes alertas foi suficiente melhorar as duas linhas de código no ficheiro de configuração do preprocessador. A linha *http_inspect* afecta o funcionamento global do preprocessador. O *iis_unicode.map*, corresponde à página de *unicode* a ser utilizado pelos servidores IIS, que neste caso, para a língua inglesa (Americano) tem um valor de 1252. Existem outros valores dependendo sempre do número de caracteres do abecedário de uma língua.

```
preprocessor http_inspect: global \
    iis_unicode_map unicode.map 1252
```

A segunda linha a *http_inspect_server* define os alertas e o comportamento de normalização do tráfego dos servidores.

```
preprocessor http_inspect_server: server default \
    profile all ports { 80 8080 8180 } oversize_dir_length 500
```

Considerando as configurações acima referidas foram feitas as seguintes alterações ao ficheiro de configuração do snort, para assim poder melhor afinar o preprocessador. A primeira linha ficou inalterada, no entanto na segunda foi alterada o valor dos *profiles* de ALL para o IIS, já que deste modo ficava mais específico aos servidores a monitorizar. O tamanho do valor do caminho foi aumentado para 600 bytes para melhor se adequar ao *software*. Optou-se numa primeira fase não especificar todos os servidores e apenas usar o valor *default.*

```
preprocessor http_inspect_server: server default \
       profile iis ports { 80 8080 8180 } oversize_dir_length 600
```

Todas as correcções dos alertas de *http-inspect* apenas possibilitaram obter um melhor desempenho do IDS. Era possível optar por não receber este tipo de alertas, mas corria-se o risco de existirem falsos negativos.

6.4.2 VLAN 84

A solução dos alertas da rede 84 foram, em alguns casos, mais difícil de resolver. A seguir listam-se alguns dos alertas principais da rede 84 assim como uma breve explicação dos mesmos e das acções a tomar.

Alerta: `SMB-DS ADMIN $ Unicode share access`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	0
192.168.X.X	8	192.168.X.X	0

Segundo este alerta houve uma tentativa de aceder à parte administrativa de uma máquina. Considerando que estas máquinas são servidores essa possibilidade existe. O que tornou estes alertas excepcionais foi o facto de estarem espaçados por 12 horas, isto é o primeiro alerta era às 12:57:x e o seguinte era às 00:57:x, havendo apenas uma diferença de segundos entre eles. Fez-se uma analíse pelo antivírus e não se detectou nada fora do normal. Como tal será necessário fazer uma verificação aos serviços para ter melhor conhecimento dos processos a correr e verificar se existe algum processo que pode ser desactivado.

Alerta: `Sql Ping Attempt`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X **192.168.X.X** **192.168.X.X** **192.168.X.X**	X	255.255.255.255	1434

Este alerta indica que houve um ping ao SQL que permite informar sobre a versão do programa em uso. Havia a possibilidade de a rede estar a ser analisada pelo *Nessus* mas esta hipótese foi logo eliminada, porque os endereços IP´s eram fidedignos e correspondiam a utilizadores que necessitavam do SQL para o seu trabalho. Neste caso apenas se alterou a regra do IDS para excluir estes endereços IP´s durante a comparação dos modelos de anomalias.

Alerta: `Misc MS Terminal server request`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	X
192.168.X.X	X	192.168.X.X	X

e

Alerta: `MISC MS Terminal Server no encryption session initiation attempt`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	1448

Os alertas acima são recebidos quando se efectuam pedidos ao *Microsoft Terminal Service*. Estes serviços correspondem aos serviços remotos fornecidos pelo *Windows Server* e possibilitam o acesso a aplicações e dados de um computador. Se existirem muitos pedidos pode provocar um DoS. Para se poder solucionar este problema é aconselhado ter o *software* sempre actualizado. No entanto estes alertas não correspondiam a essa situação, tendo surgido depois de ter havido testes de coordenação ao *Video Hall*. Assim foi considerado que vieram de uma fonte fidedigna e não houve necessidade de mais acções.

Alerta: `Portsweep portscan`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	213.199.149.8 192.168.X.X 192.168.X.X 207.46.250.153 207.46.211.250 207.46.20.252 65.55.25.125 65.55.52.84	X

Segundo este alerta houve um *portsweep* do *file server* às restantes maquínas. Apesar de se poder considerar este acto normal para o endereço de IP 192.168.X.X. Para as restantes obrigaram a uma análise cuidada. Felizmente veio a verificar-se que todos os outros IP´s, excepto um, pertencia à rede da Microsoft (American e também da European Data Center) e o *portsweep* era necessário para haver a actualização dos programas da Microsoft no servidor, sendo assim foi então criado uma excepção na regra no IDS para estes IP´s. No entanto o endereço IP 192.168.X.X não só tinha um elevado número de alertas como também era responsável por outros alertas do tipo *portscans*, pelo que foi solicitado ao utilizador uma análise do seu sistema através do antivírus. Esta acção permitiu a descoberta de alguns vírus e *trojans* que depois de eliminados os alertas pararam.

Alerta: `Netbios Nimda RICHED20.DLL`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X **192.168.X.X**	X	192.168.X.X	139
192.168.X.X **192.168.X.X** **192.168.X.X**	X	192.168.X.X	139

Estes alertas avisam sobre uma possível existência da *worm nimda*. Nas máquinas infectadas foi inicialmente actualizado o antivírus para se depois poder realizar uma análise do computador. No entanto verificou-se que nas máquinas com o sistema operativo Windows 2000 instalado, quando tentou-se actualizar o antivírus obtinha-se um erro, o que consequentemente impossibilitava a actualização e a análise da máquina.

A solução para estes casos foi, instalar o sistema operativo Windows XP. Esta tarefa já estava planeada para os computadores com o sistema operativo Windows 2000, apenas foi antecipada para estas máquinas, visto haver uma possível ameaça.

Depois do novo sistema operativo estar instalado, por prevenção optou-se fazer uma análise aos computadores através do antivírus, onde não se detectou a presença da *worm*. Em relação a estas máquinas o IDS não voltou a receber este tipo de alertas.

No entanto havia máquinas que já tinha o sistema operativo Windows XP instalado e onde se continuava a receber alertas no IDS.

Nestas máquinas também se fez uma actualização e análise com antivírus onde se veio a verificar que a alerta era falsa. O IDS para detectar a presença da *worm nimda* verifica se existe um ficheiro riched20.dll num pacote do tráfego, infelizmente este ficheiro também se encontra em aplicações como mspaint, notepad e o CAD. Foi o que se veio a verificar com as máquinas que tinham o CAD instalado.

Como a *worm nimda* propaga-se por ficheiros infectados, correio electrónico ou através da partilha de ficheiros foi decidido continuar com esta regra activa, porque: primeiro pode vir a acontecer que esta alerta não venha a ser falsa e segundo porque os alertas recebidos eram de uma pequena quantidade que facilmente se podia investigar e vir a evitar problemas na rede.

Alerta: `NETBIOS SMB-DS Trans unicode Max Param/Count DOS attempt`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	1971	192.168.X.X	445

e

Alerta: `Netbios SMD DS-IS arpc`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	4090	192.168.X.X	445

e

Alerta: `ICMP Ping speedera`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	192.168.X.X	X

Os alertas acima referidos foram obtidos no momento que se inicializou a instalação do sistema operativo para Windows XP. Não houve por isso nenhuma alteração ao IDS ou acções tomadas às máquinas em causa, sei que para a resolução destes alertas era

aconselhável fazer uma actualização ao sistema operativo, o que estava a decorrer na altura. As regras continuaram activas por precaução.

Alerta: `Telnet traffic encrypted`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	X	172.16.232.95	X

A este alerta foi dada prioridade máxima porque correspondia a um servidor virtual de serviços de facturação (SIFACT) que estava a comunicar com um IP que não pertencia à rede do ASC, sendo obvio que se um *cracker* conseguisse obter informação desse servidor, podia provocar alguns danos à empresa. Depois de se ter verificado o pacote capturado e de se ter analisado os endereços MAC, veio a confirmar-se que este correspondia a um computador portátil da ANA utilizado para testes.

Alertas: `ICMP L3retriever Ping`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	8	192.168.X.X	0
192.168.X.X	8	192.168.X.X	0

e

Alertas: `ICMP PING NMAP`

Origem	*Porta*	*Destino*	*Porta*
192.168.X.X	8	192.168.X.X	0

Estes alertas foram gerados aos milhares por dia sendo na figura 6.4 bem visível a diferença dos alertas entre as VLAN´s. Numa rede em condições normais estes alertas são também gerados mas em quantidade muito menor. Haviam várias hipóteses para esta ocorrência como, por exemplo, uma má configuração ou, considerando uma situação mais grave, um *trojan.*

Uma das primeiras acções foi verificar se a quantidade de tráfego de entrada era parecido, com a quantidade de saída e para tal foi utilizada a ferramenta *PRTG Network Monitor* (figura 6.11). Este programa possibilita verificar a quantidade de tráfego ao longo do tempo assim como também os recursos disponíveis.

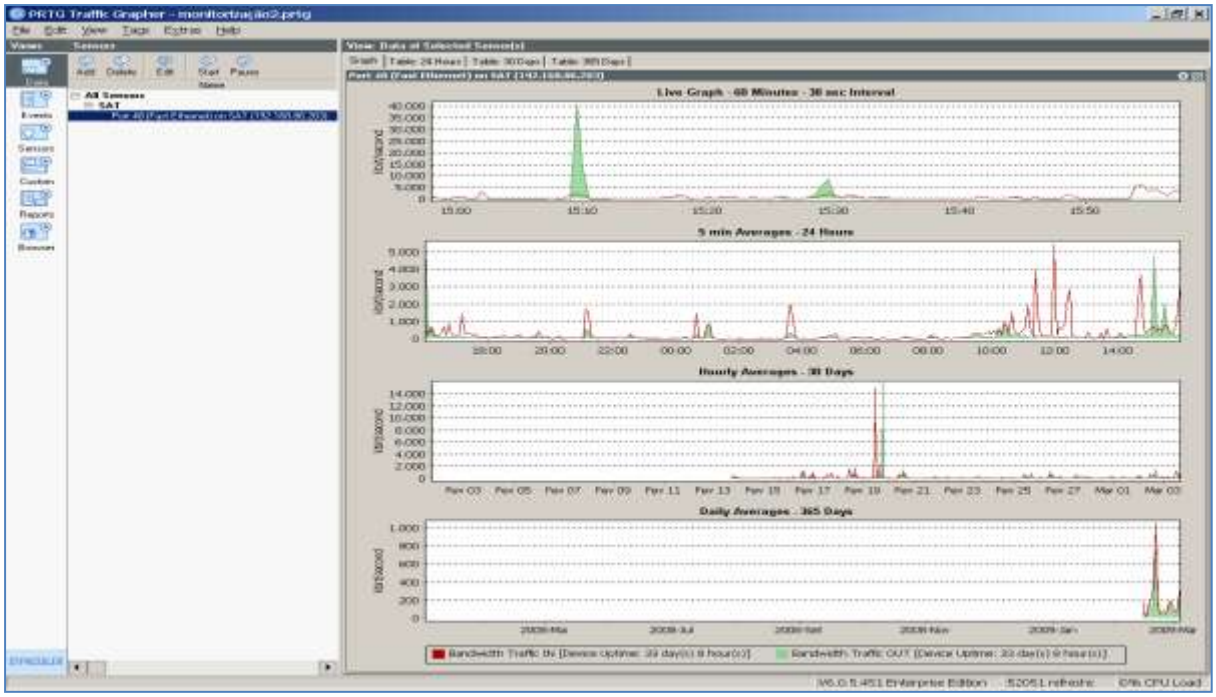

Figura 6. 11 - PRTG Network Monitor

A figura 6.12 mostra os resultados obtidos onde se pode verificar alguns picos, mas nada que possa confirmar actividade fora do normal na rede.

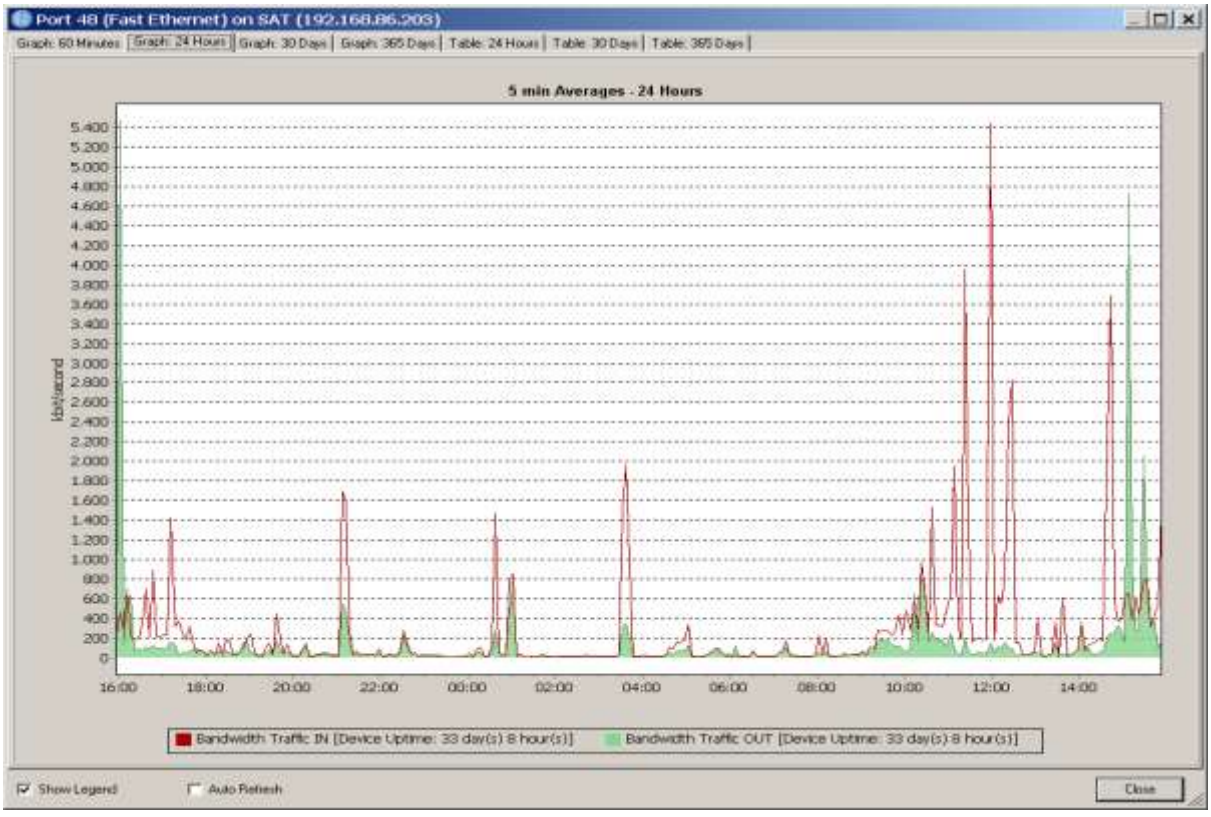

Figura 6. 12 - PRTG Network Monitor – tráfego de um dia

Como os alertas eram maioritariamente em relação ao endereço IP 192.168.X.X (*file server*), foi decidido comparar os serviços activos com o servidor 192.168.X.X, já que estes deviam ser semelhantes e também porque o segundo servidor não apresentava problemas. Para executar esta tarefa utilizou-se o *Nessus* e verificou-se que o *file server* tinha 5 portas adicionais abertas:

- port 1161 – health polling
- port 2515 – msrpc
- port 3548 – interworld
- port 1054 – brvread

De todos, apenas a porta 1054 apresentava a possibilidade de ser um *trojan* sendo neste caso o AckCmd. Foi de imediato feito uma análise ao servidor com o antivírus, mas os resultados não foram os desejáveis: realmente foi encontrado um *trojan* mas não relacionado com o que se pretendia e, depois de eliminado, os alertas continuavam e com a mesma quantidade. Para fazer uma análise mais detalhada foi colocado o *Wireshark* a monitorizar o tráfego ICMP com uma TAP. Os resultados foram curiosos: havia *reply* mas muito poucos *request* e os ICMP deveriam ser sempre aos pares (*request* e *reply*) como mostra a figura 6.13.

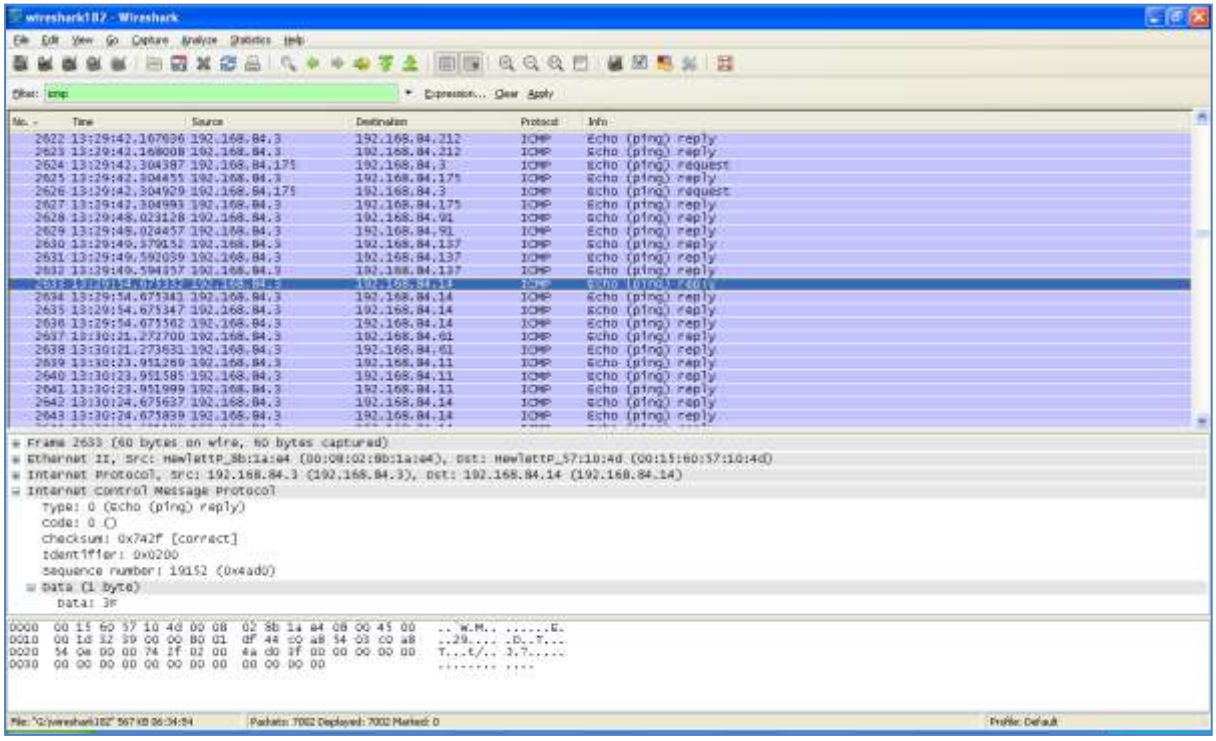

Figura 6. 13 - ICMP obtidos pelo Wireshark

Foram depois escolhidos alguns endereços IP´s e fez-se uma comparação da actividade das máquinas com os ICMP´s. As figuras 6.14 e 6.15 mostram os resultados obtidos.

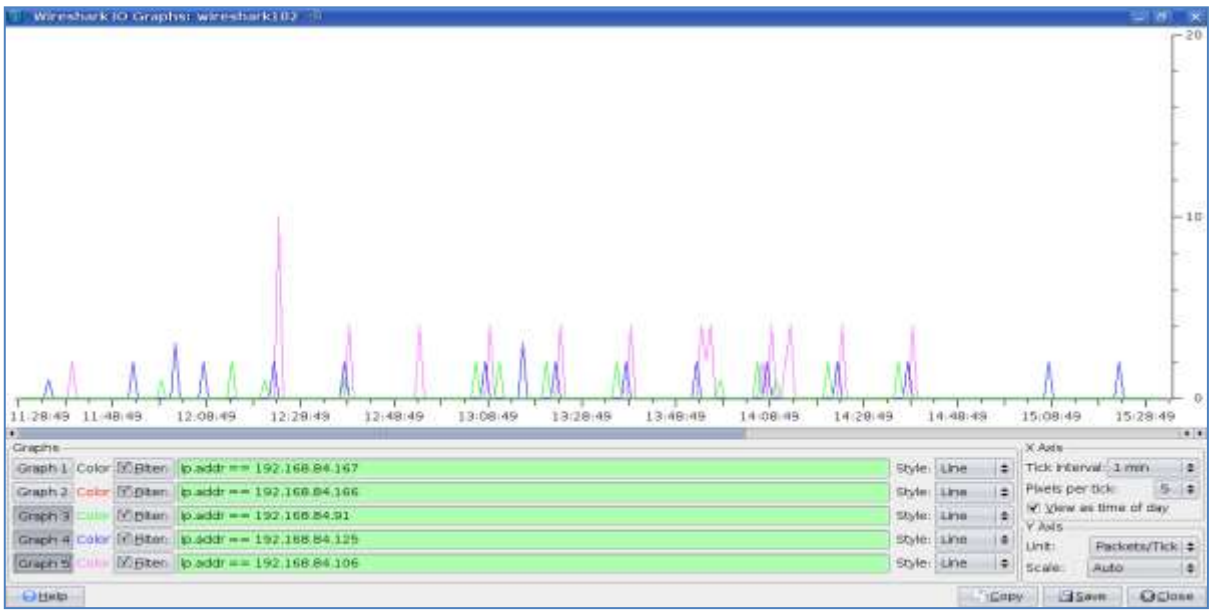

Figura 6. 14 - Resultados gráficos do *Wireshark*

Na figura 6.14 podem visualizar-se os ICMP de aproximadamente 20 em 20 minutos o que corresponde à renovação da tabela de ARP, como já se tinha visto na VLAN 14, mas na figura 6.15 verifica-se que existem algumas máquinas mais activas.

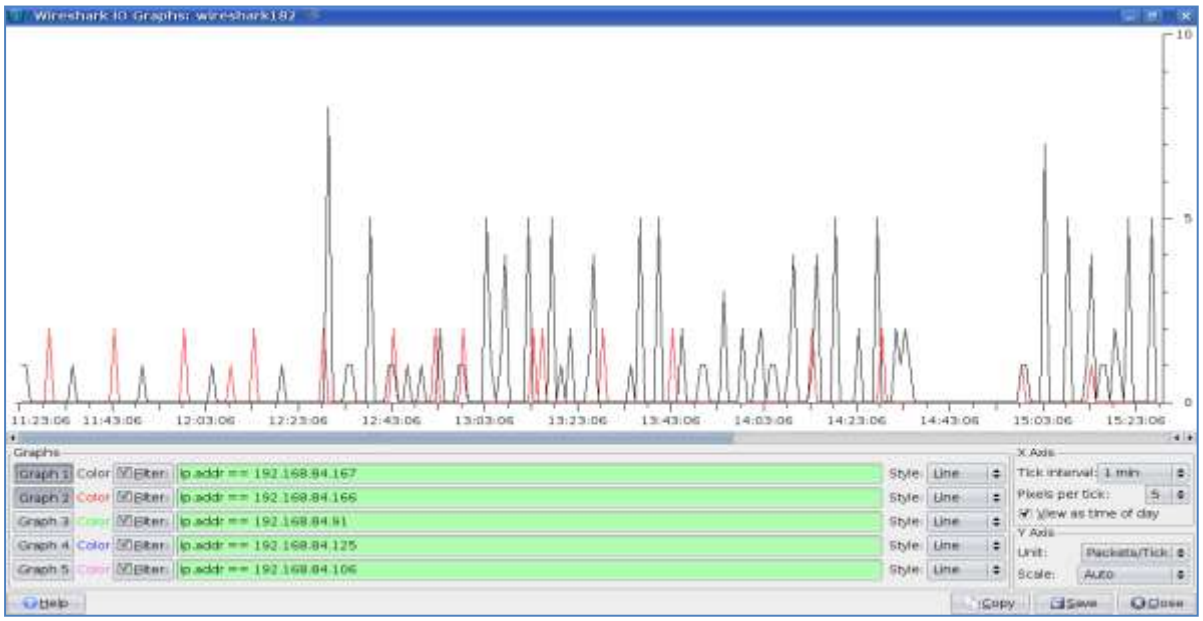

Figura 6. 15 - Resultados gráficos do *Wireshark*

A próxima hipótese a verificar foi a existência de serviços das máquinas mais activas não instalados mas menos activas. Para fazer essa análise foi utilizada a ferramenta *Anvir TaskManager Pro* (figura 6.16) e concluiu-se que não havia diferenças significativas entre as máquinas para explicar os alertas.

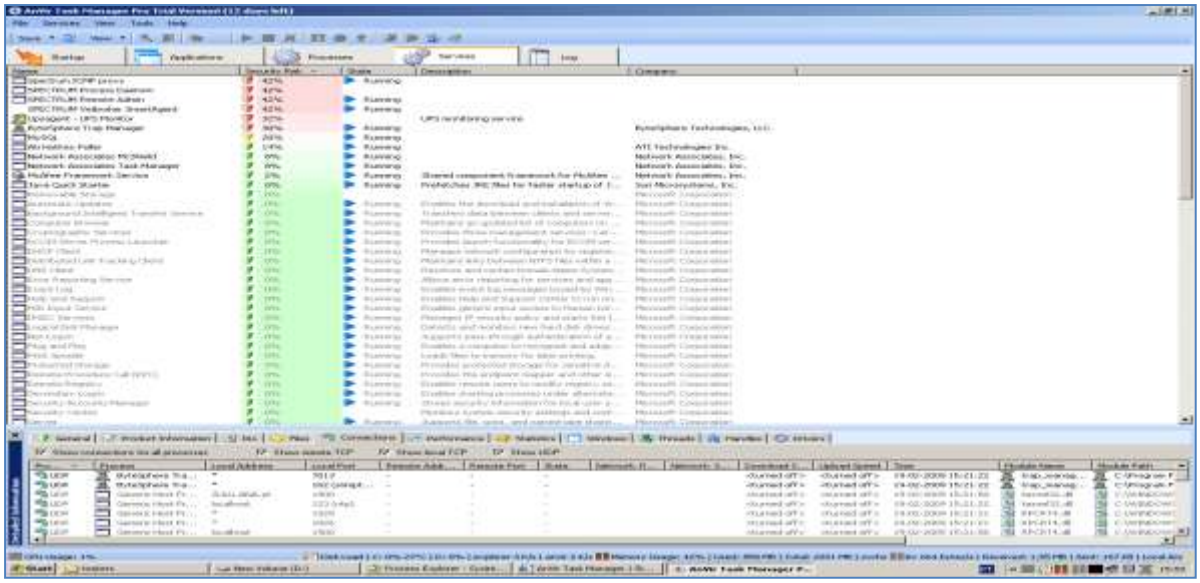

Figura 6. 16 - A ferramenta *Anvir TaskManager Pro*

Depois de se ter verificado as várias hipóteses, pode-se concluir que as máquinas com o sistema operativo Windows 2000 foram as mais activas. Decidiu-se manter a regra mas excluir a rede 84 porque: uma parte dos alertas *L3retriever Ping* provém da renovação da tabela ARP, estes alertas também eram verificados quando havia manutenção remota nas máquinas pelo administrador e o *software Network security retriever* não é utilizado pelo ASC.

Depois de haver as alterações referidas às redes 84 e 14, conseguisse diminuir a quantidade de alertas para aproximadamente 1800 por dia (figura 6.17). Enquanto este valor pode parecer elevado não o é, porque muitos alertas provinham de *portsweeps*, *portscans* e de ICMP do *Netbios* e, não é aconselhável eliminar estas regras por serem sinais de possíveis vírus, *trojans*, ou tentativas de intrusões nas máquinas.

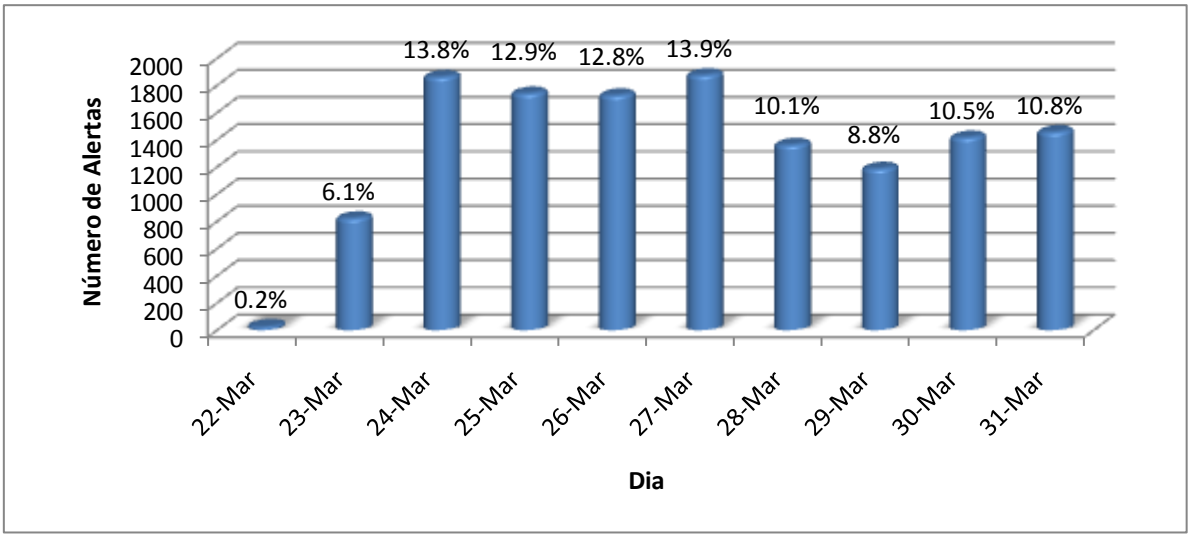

Figura 6. 17 - Número de alertas finais vs tempo (dias)

6.5 O *honeypot* implementado

A *honeynet* de teste foi criada para usar os endereços de IP livres podendo criar-se uma rede completa com *routers*, vários terminais e impressoras tentando iludir ao máximo o *cracker*. Infelizmente, numa rede como a do ASC, em constante evolução e alteração, é difícil haver muitos endereços IP livres. Por isso foi decidido implementar apenas um *honeypot* tentando emular a maior quantidade de serviços possível para permitir uma melhor interacção com o *cracker*.

Nesta implementação foram utilizados os mesmos programas do *honeynet* de teste, ou seja, o *honeyd* com vários emuladores como o FTP, o *mydoom* e o telnet. O *honeypot* foi instalado no computador que serviu de sensor para o IDS. Não é aconselhável implementar um *honeypot* e um IDS na mesma máquina, visto os objectivos serem muito diferentes, isto é, o *honeypot* é implementado para ser comprometido, enquanto com o IDS uma intrusão pode comprometer os dados e colocar a rede em risco.

O *honeyd* necessita de uma gama de endereços IP's para funcionar como uma rede virtual. Caso esta situação não se verifique e apenas exista um endereço IP ou se não for possível utilizar o programa *arpd*, torna-se necessário fazer uso dos *iptables* para redireccionar os pacotes. Como para a rede do ASC se decidiu implementar um *honeypot* com apenas um IP, isto é, um IP para o *honeywall* e o outro para o *honeypot* (figura 6.18).

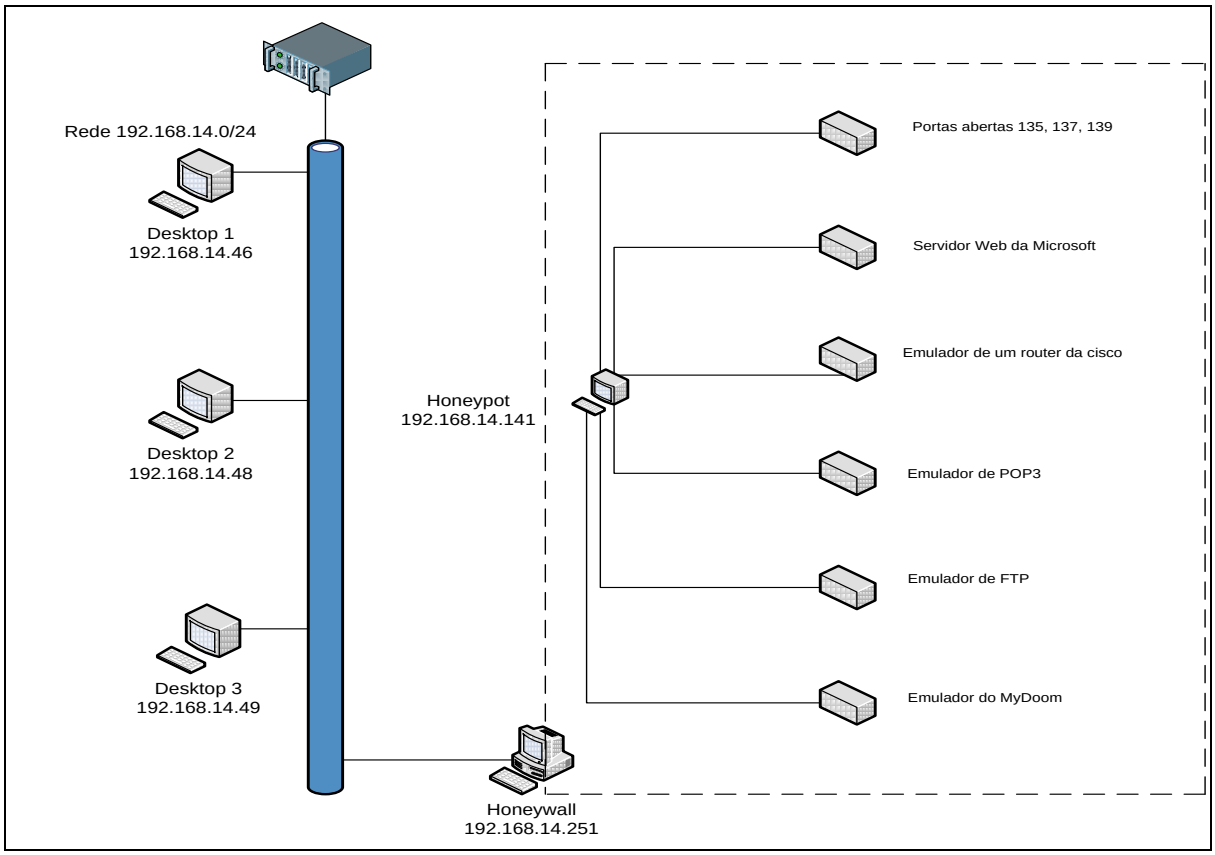

Figura 6. 18 - *Honeypot* implementado no ASC

O *honeywall* ficou a funcionar com o IP da máquina (192.168.14.251), e para o *honeypot* foi necessário utilizar um IP livre da rede 14 que foi criado virtualmente na máquina (192.168.14.141). Optou-se por simular o Windows 2000 SP2 com as portas 135, 137 e 139 abertas. Estas portas foram escolhidas por possibilitarem um *hacker* executar código na máquina, daí serem mais procuradas. Como se utilizou um ambiente Windows teve de se utilizar um emulador do Web *server* do Windows, caso contrário o *hacker* podia desconfiar que era um *honeypot*. Foi também emulado os serviços POP3, FTP, um *router* da Cisco e uma porta aberta pelo vírus MyDoom.

O *honeypot* foi colocado e executado na rede 14 no SAT e foi utilizado o programa *arpd* para permitir o envio dos pacotes para o mesmo (figura 6.19).

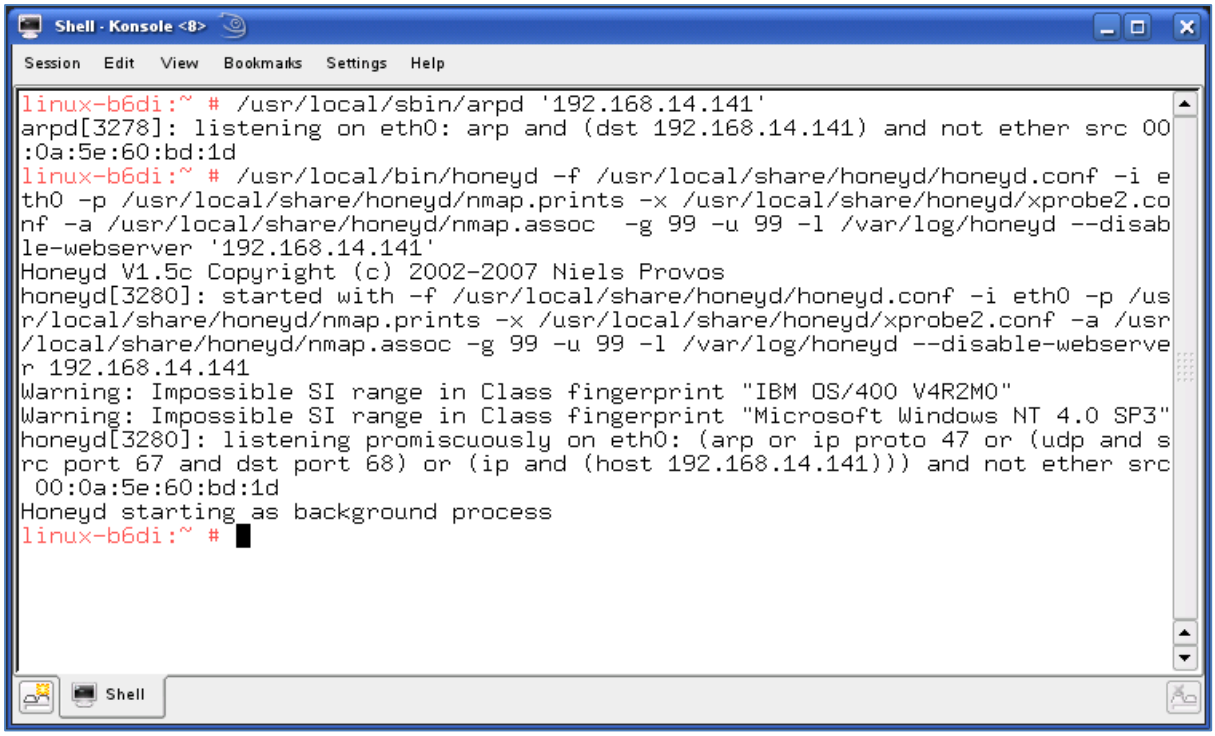

Figura 6. 19 - Inicialização do *honeypot*

Foram feitos alguns testes para verificar se o funcionamento do *honeypot* estava correcto, primeiro através de um ping (figura 6.20) e depois fazendo uso do *Nessus*. Verificou-se que todas as portas dos *scripts* emulados se apresentavam nas condições desejadas.

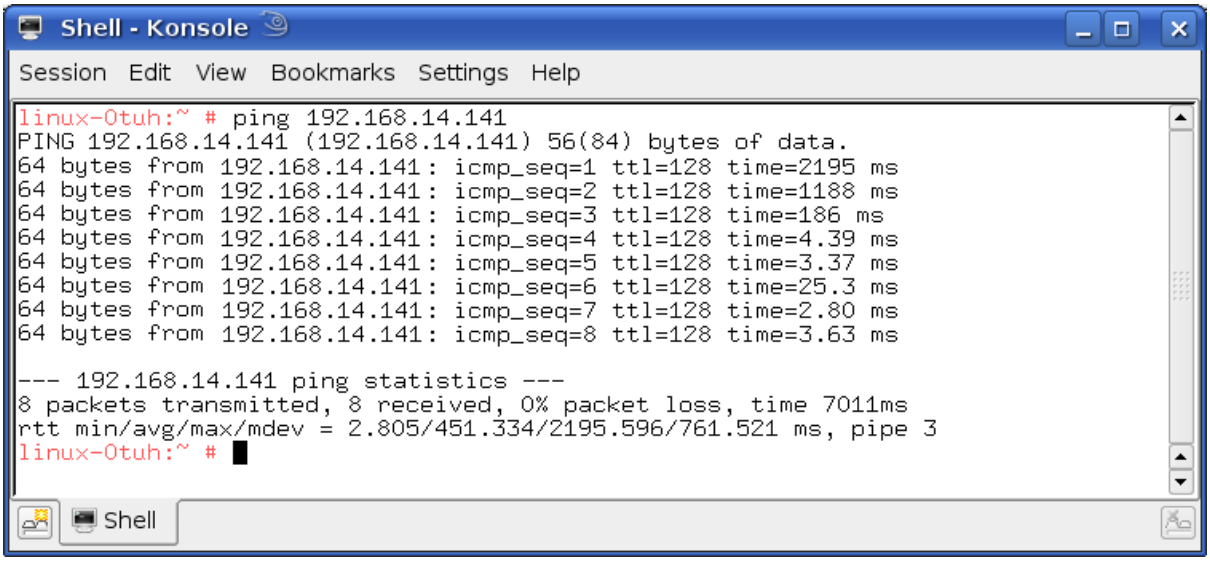

Figura 6. 20 - Teste realizado ao *honeypot*

O serviço de emulação do servidor POP3 consegue emular alguns dos comandos do protocolo RFC 1939 do servidor, mostrando as acções que normalmente se verificam. Pode até assinalar mensagens como eliminadas, mas após o *cracker* sair do programa, como não houve qualquer actuação relativamente a essas mensagens (figura 6.22), fica tudo como estava originalmente, à espera do próximo intruso. A figura 6.21 apresenta a *shell* de comandos do servidor POP3. Na situação presente, o utilizador é "USER" e a palavra-chave é "jo", caso o *hacker* não consiga inserir os dados correctos após 3 tentativas, tem de voltar a estabelecer contacto. Como os emuladores são pequenos programas facilmente se consegue alterar o código para parecer mais real.

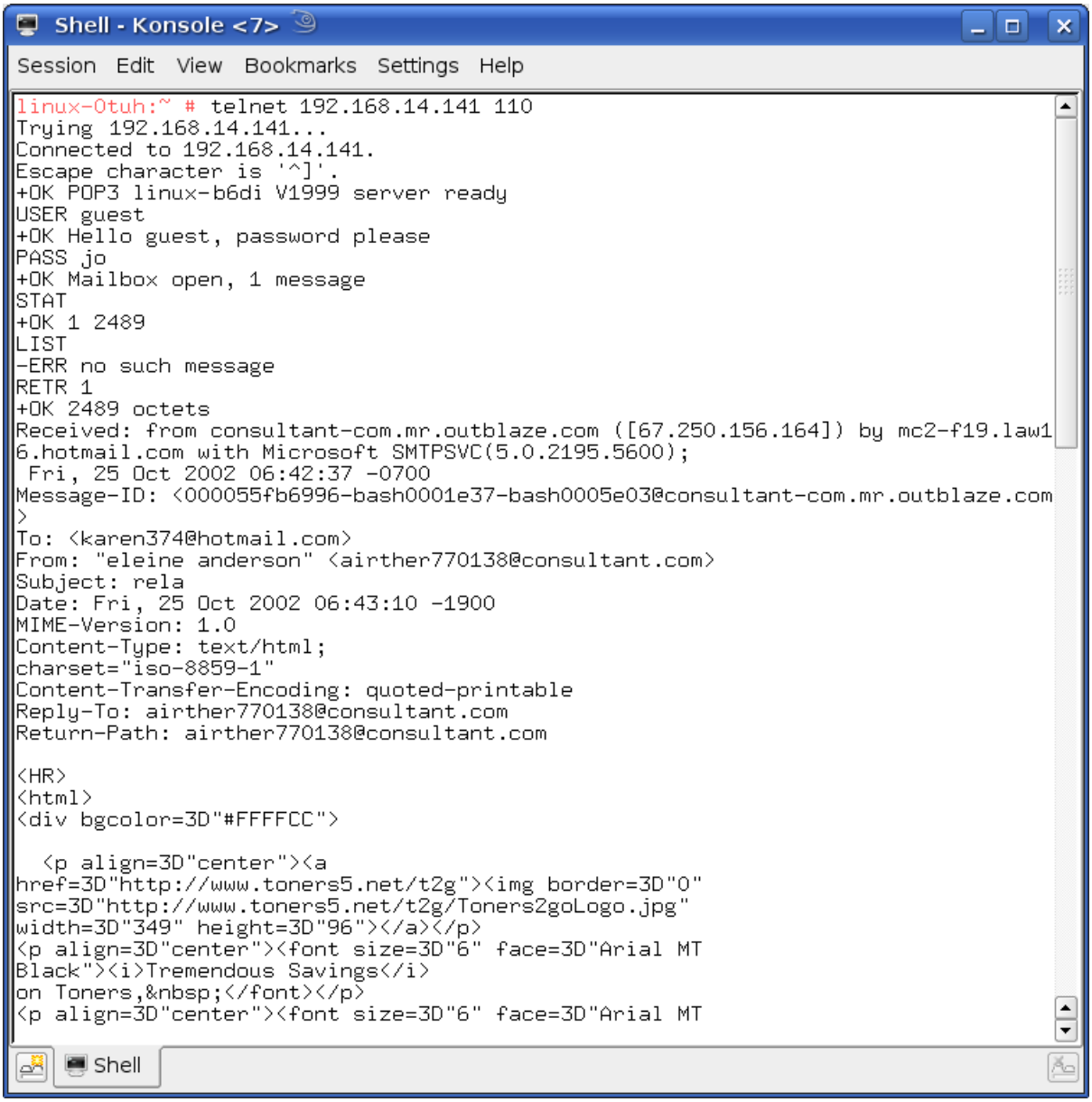

Figura 6. 21 - A *shell* do serviço do servidor POP3

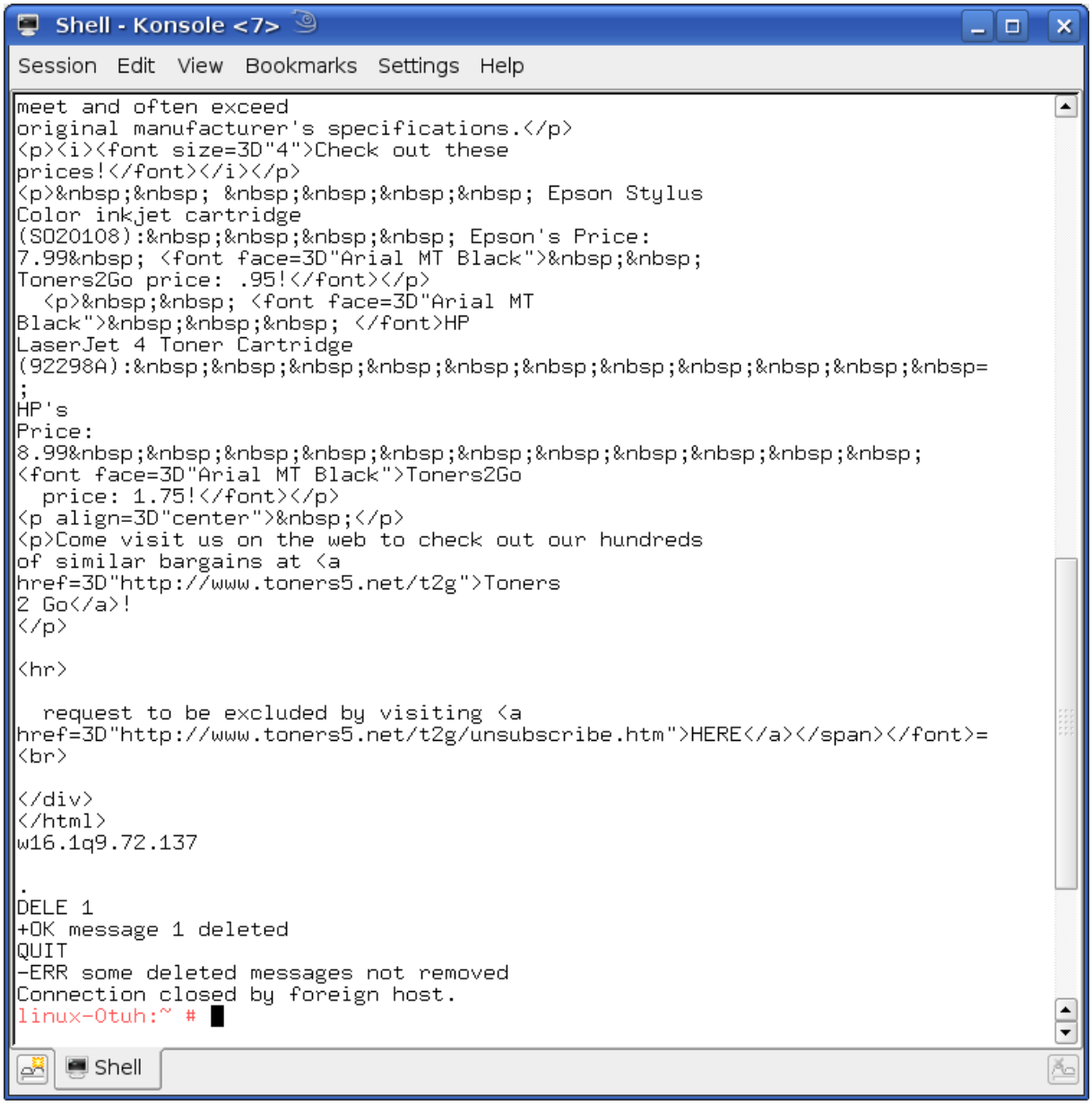

Figura 6. 22 - A *shell* do serviço do servidor POP3

A simulação do serviço do *router* Cisco através de telnet é bastante mais simples, como se pode ver na figura 6.23. Apenas se pode visualizar a página inicial onde o pedido do utilizador e da palavra-chave é feito. O emulador permite três tentativas para obter o acesso, que nunca irá acontecer. Caso não se verifiquem as três tentativas e após um tempo determinado, o *prompt* volta a ser visualizado.

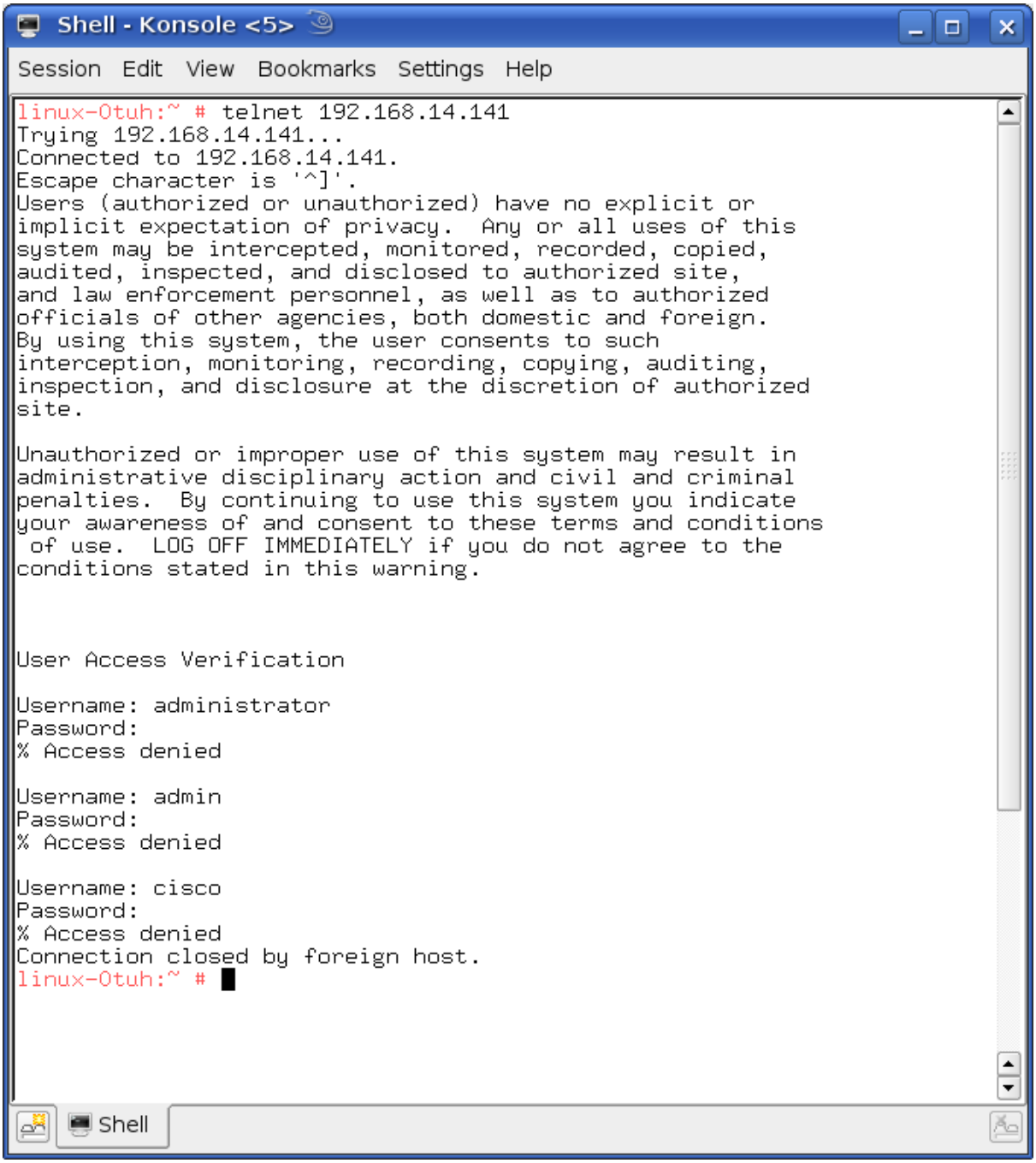

Figura 6. 23 - A *shell* do serviço telnet

Foi também simulado um serviço FTP (figura 6.24), este como os anteriores também só permite três tentativas para inserir o utilizador e palavra-chave correcta. Infelizmente não é uma aplicação muito evoluída apenas permite três comandos do FTP o MKD, CWD e o NOOP sendo que para os outros responde com "command not implemented", o que depois de algumas tentativas torna-se duvidoso e o *hacker* pode vir a perceber que não está a interagir com um sistema real.

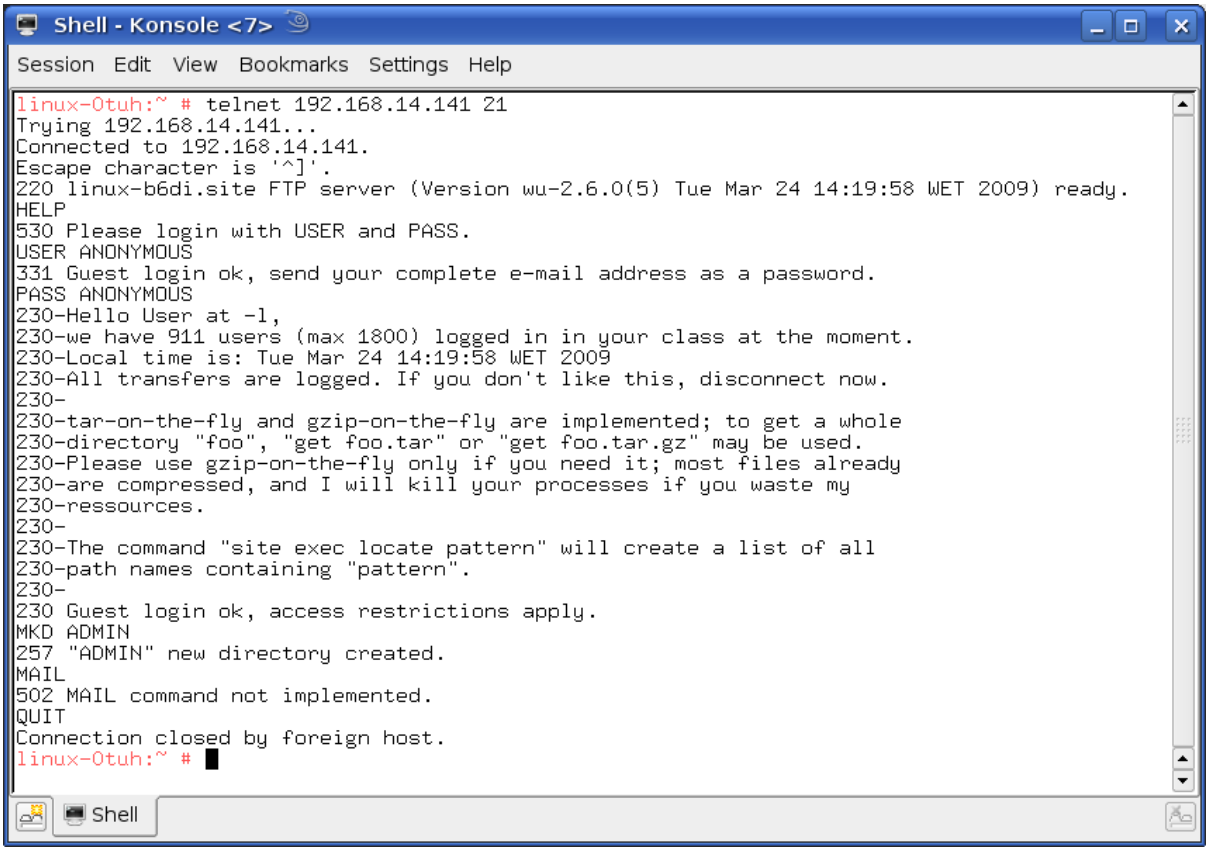

Figura 6. 24 - A *shell* do serviço do servidor FTP

É possível obter alguma informação no registo do *honeypot*, como mostra a figura 6.25, onde se vê uma pequena amostra obtida após um teste. À esquerda temos a data e a hora da tentativa de intrusão e logo a seguir o tipo de protocolo utilizado. O *honeyd* consegue identificar tanto os protocolos ICMP, UDP, TCP como os protocolos mais raros utilizados apenas por programas de intrusões. A coluna seguinte corresponde ao tipo de conexão: o "S" (*start*) indica o ínicio, o "E" (*end*) o fim da mesma, enquanto o " - " informa que o pacote não pertence a nenhuma ligação. As colunas seguintes informam sobre os IP´s e as portas utilizadas na origem e destino. A última coluna apresenta informação sobre o tipo do sistema operativo utilizado pelo emissor e também a quantidade de bytes recebidos depois de fechada a conexão.

```
2009-03-30-16:13:22.0575 honeyd log started ------
2009-03-30-16:13:22.0580 udp(17) - 192.168.14.56 1086 192.168.14.141 161: 106
2009-03-30-16:13:42.2334 udp(17) - 192.168.14.56 1086 192.168.14.141 161: 106
2009-03-30-16:13:48.2334 udp(17) - 192.168.14.56 1086 192.168.14.141 161: 106
2009-03-30-16:13:54.2335 udp(17) - 192.168.14.56 1086 192.168.14.141 161: 106
2009-03-30-16:14:12.5794 icmp(1) - 192.168.14.114 192.168.14.141: 8(0): 60
2009-03-30-16:14:13.5702 icmp(1) - 192.168.14.114 192.168.14.141: 8(0): 60
2009-03-30-16:14:14.5701 icmp(1) - 192.168.14.114 192.168.14.141: 8(0): 60
2009-03-30-16:14:26.2761 tcp(6) S 192.168.14.114 1817 192.168.14.141 23 [Windows XP SP1]
2009-03-30-16:14:34.2090 tcp(6) S 192.168.14.114 1818 192.168.14.141 110 [Windows XP SP1]
2009-03-30-16:14:39.9805 tcp(6) S 192.168.14.114 1819 192.168.14.141 110 [Windows XP SP1]
2009-03-30-16:15:29.2952 tcp(6) E 192.168.14.114 1817 192.168.14.141 23: 0 0
2009-03-30-16:15:37.2300 tcp(6) E 192.168.14.114 1818 192.168.14.141 110: 0 0
2009-03-30-16:15:43.0078 tcp(6) E 192.168.14.114 1819 192.168.14.141 110: 0 0
2009-03-30-16:47:57.6344 honeyd log stopped ------
2009-03-31-12:32:57.0990 honeyd log started ------
2009-03-31-12:34:52.3790 tcp(6) S 192.168.14.114 2087 192.168.14.141 23 [Windows XP SP1]
2009-03-31-12:34:53.1789 udp(17) - 192.168.14.56 1086 192.168.14.141 161: 106
2009-03-31-12:34:59.1796 udp(17) - 192.168.14.56 1086 192.168.14.141 161: 106
2009-03-31-12:35:02.7726 tcp(6) S 192.168.14.114 2088 192.168.14.141 110 [Windows XP SP1]
2009-03-31-12:35:55.4079 tcp(6) E 192.168.14.114 2087 192.168.14.141 23: 0 0
2009-03-31-12:36:05.7925 tcp(6) E 192.168.14.114 2088 192.168.14.141 110: 0 0
2009-03-31-13:27:10.8106 tcp(6) S 192.168.14.114 26200 192.168.14.141 110 [Linux 2.6 .1-7]
2009-03-31-13:32:27.4006 tcp(6) E 192.168.14.114 26200 192.168.14.141 110: 21 100
2009-03-31-13:34:49.4767 tcp(6) - 192.168.14.114 26200 192.168.14.141 110: 66 PA
2009-03-31-13:34:49.4769 tcp(6) - 192.168.14.114 26200 192.168.14.141 110: 66 PA
2009-03-31-13:51:34.6593 tcp(6) S 192.168.14.114 7724 192.168.14.141 21 [Linux 2.6 .1-7]
2009-03-31-14:05:14.6122 tcp(6) - 192.168.14.114 27595 192.168.14.141 110: 52 FA
2009-03-31-14:05:14.8813 tcp(6) E 192.168.14.114 7724 192.168.14.141 21: 148 1324
```

Figura 6. 25 - Resultados obtidos dos registos do *honeyd*

Felizmente não se verificou qualquer intrusão no *honeypot.* Todos os registos verificados foram de computadores fidedignos que foram utilizados para fazer os testes ao *honeypot* implementado.

6.6 A utilização do *aircrack-ng*

Foi decidido utilizar a ferramenta *Aircrack-ng* na verificação da segurança das palavras-chave de uma rede sem fios. Esta ferramenta foi escolhida não só por ser *open source* mas também porque, em comparação com outras ferramentas do mesmo tipo, consegue recuperar palavras-chave tanto na encriptação WEP como na WPA. Foi realizado um teste com um PDA da ANA. Nessa altura, o PDA estava a ser submetido a testes para a utilização na rede sem fios da aerogare. No entanto á partida era quase impossível obter a palavra-chave do PDA, por falta de capacidade de processamento e tempo. Optou-se então por fazer uso de uma rede sem fios externa à do ASC. Esta rede permitiu não só a utilização do *Aircrack* como também a colocação de um IDS na rede, através de uma porta de acesso para testes (Test Access Port -TAP).

6.7.1 *Aircrack-ng*

A capacidade do *Aircrack-ng* em recuperar as palavras-chave de uma rede sem fios está limitada ao tipo de encriptação utilizado.

O *Aircrack-ng v1.0* foi instalado numa máquina com o sistema operativo *Suse v10.3* com componente gráfica tendo sido também necessária a instalação da dependência *zlib v1.2.3.* Por razões já referidas todas as ferramentas e dependências foram instaladas através da compilação (*tarballs*).

Depois de executado o *airodump-ng* é possível visualizar os pontos de acesso com os clientes associados (figura 6.26). A primeira linha mostra o canal actual, o tempo decorrido, a data e se foi detectado um *handshake.* A seguir são apresentados dois blocos: os pontos de acesso e os clientes.

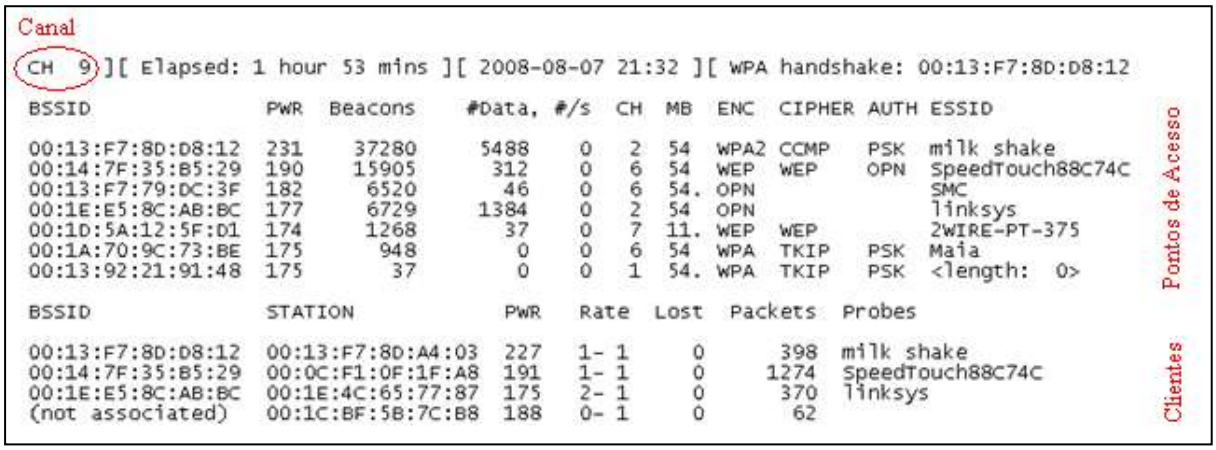

Figura 6. 26 - Exemplo dos AP´s e clientes captados por uma placa de rede.

No bloco de pontos de acesso obtém-se a seguinte informação:

- BSSID (*Basic Service Set Identifier*) - endereço MAC dos pontos de acesso;
- PWR - intensidade do sinal;
- Beacons - quantidade de pacotes *beacon* recebidos (um pacote *beacon* corresponde a um pacote de gestão que permite às estações estabelecer e manter uma comunicação);
- Data - quantidade de pacotes de dados recebidos;
- CH - canal no qual o AP está a funcionar;
- MB - velocidade de máxima de transferência suportada pelo AP, valor 11 significa que a norma 802.11b está em uso enquanto o valor 54 corresponde à norma 802.11g;
- ENC - tipo de encriptação, OPN significa que nenhuma encriptação está presente; no caso de WEP ou WPA/WPA2 é a encriptação correspondente;
- CIPHER - cifra detectada;
- AUTH - autenticação do protocolo;
- ESSID - nome da rede (esta informação por vezes não está disponível).

O bloco dos clientes fornece a seguinte informação:

- BSSID - o MAC do AP ao qual o cliente está associado;
- STATION - o MAC do cliente;
- PWR - intensidade do sinal;
- LOST - número de pacotes perdidos nos últimos 10 segundos;
- Packets - número de pacotes recebidos;
- Probes - redes a que o cliente se quer ligar, se já não estiver ligado.

A figura 6.27 mostra as terminais que era possível visualizar do SAT, mas na aerogare estavam disponíveis mais. Tentou-se obter a palavra-chave de um PDA da ANA mas como eram necessários bastantes pacotes capturados para conseguir fazer uma descodificação da palavra-chave, os obtidos do PDA na aerogare não eram suficientes. Sabendo também que o PDA da ANA utilizava a encriptação WPA com a cifra TKIP, dificilmente se conseguiria obter a palavra-chave pelo método de desencriptação do *Aircrack-ng*. A dificuldade está na palavra ser constituída por 28 caracteres com letras maiúsculas, letras minúsculas, símbolos de pontuação e numeração. O tempo estipulado para obter a palavra-chave com uma máquina a processar 500 000 palavras por segundo é de aproximadamente 1.55e24 anos [40], o que se pode considerar como impossível.

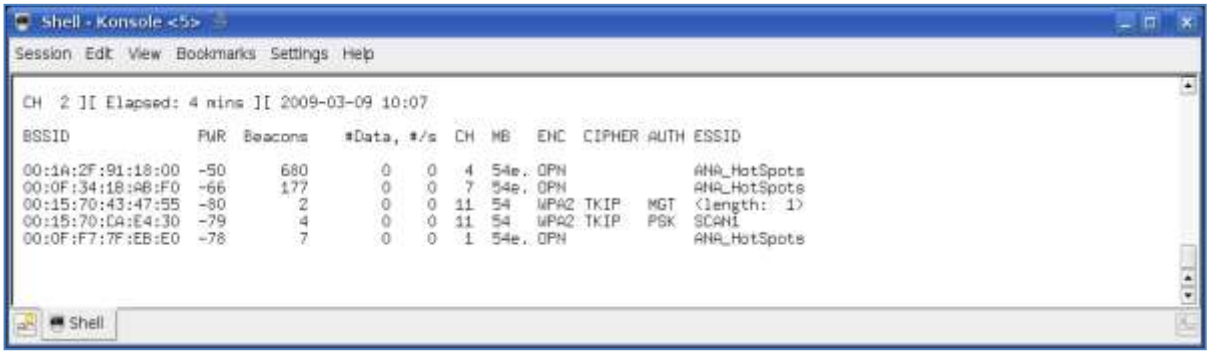

Figura 6. 27 - Terminais visualizados no SAT pelo *aircrack*

Para se poder visualizar bem o funcionamento do *aircrack-ng* optou-se por testá-lo numa rede externa à do ASC. A figura 6.28 mostra os terminais captados na altura pela ferramenta.

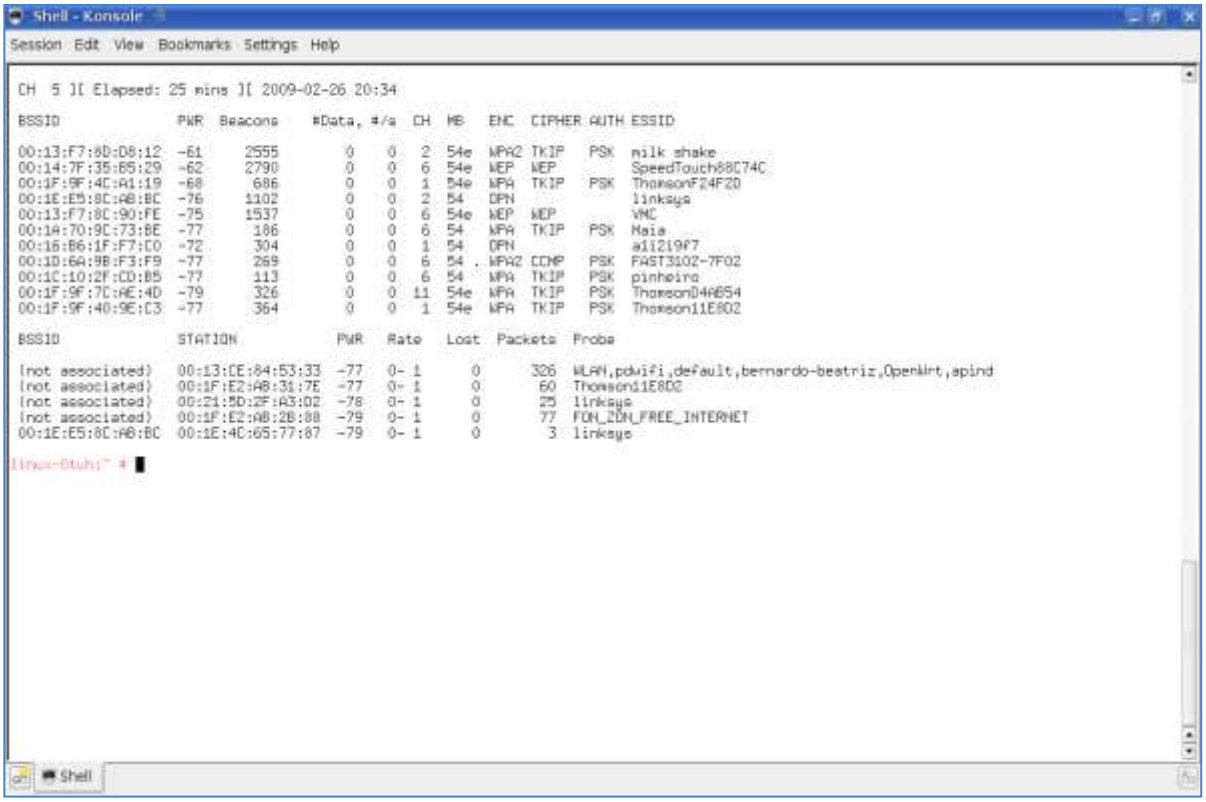

Figura 6. 28 - Redes sem fios

Para obter melhores resultados é aconselhável escolher um AP que já tem um cliente associado e também aquele que tem maior intensidade no sinal. Como mostra a figura 6.28, havia vários candidatos para o teste: o *linksys* tinha um cliente associado mas estava em sinal aberto; os dois primeiros *milk shake* e *SpeedTouch88C74C* tinham o sinal mais forte, o *milk shake* estava a utilizar a cifra *Temporal Key Integrity Protocol* (TKIP), enquanto que o *SpeedTouch88C74C* estava a utilizar a cifra *wired equivalent privacy* (WEP). Foi decidido

fazer dois testes, o primeiro ao protocolo WAP fazendo uso do *milk shake* mesmo que este aparentasse ser o mais difícil de obter a palavra-chave, e o segundo ao protcolo WEP utilizando o *SpeedTouch88C74C*.

Depois de a ferramenta estar várias horas a fazer captura de pacotes conseguiu-se informação suficiente para obter um bom resultado. A figura 6.29 mostra os resultados da ferramenta *aircrack-ng* utilizando o método do dicionário para obter a palavra-chave. O tempo de espera depende da velocidade de processamento da máquina e do tamanho do dicionário. Neste caso foi utilizado o dicionário *Jack the Ripper*.

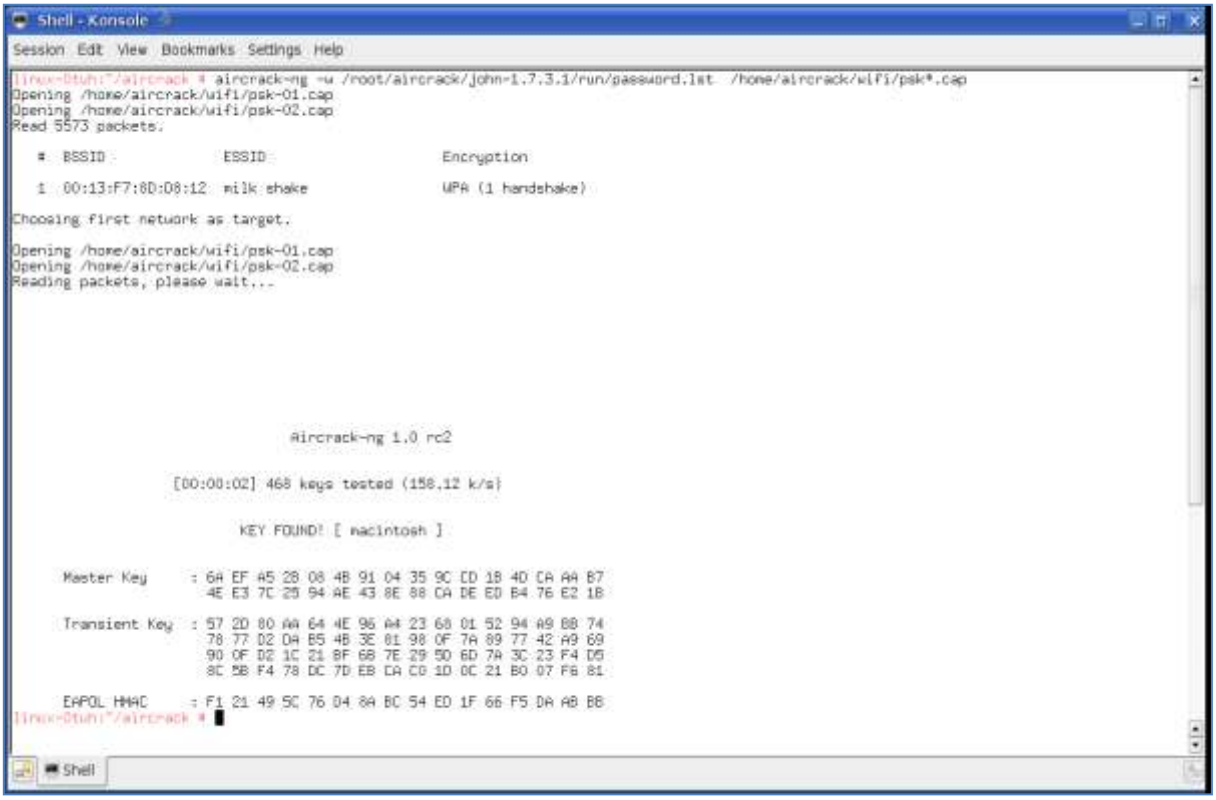

Figura 6. 29 - Resultado obtido pelo Aircrack-ng no WPA

Apesar de a ferramenta ter demorado bastante tempo para obter a palavra-chave, conseguiu descodificá-la por ela ser vulgar. Neste caso este AP apresenta um elevado risco de segurança, por falta de uma palavra-chave robusta. Para se conseguir maior segurança, as palavras-chave deviam ser constituídas por números, símbolos de pontuação, letras minúsculas e maiúsculas e não por palavras vulgares. Esta falta de segurança na palavra-chave pode ter consequências graves para o responsável da rede.

Como no protocolo WEP já existem grandes lacunas na segurança sendo possível obter facilmente as palavras-chave, não se esperava muita dificuldade na obtenção da mesma. Foi exactamente o que se verificou, com poucos pacotes capturados rapidamente a palavra-chave foi descoberta, como mostra a figura 6.30.

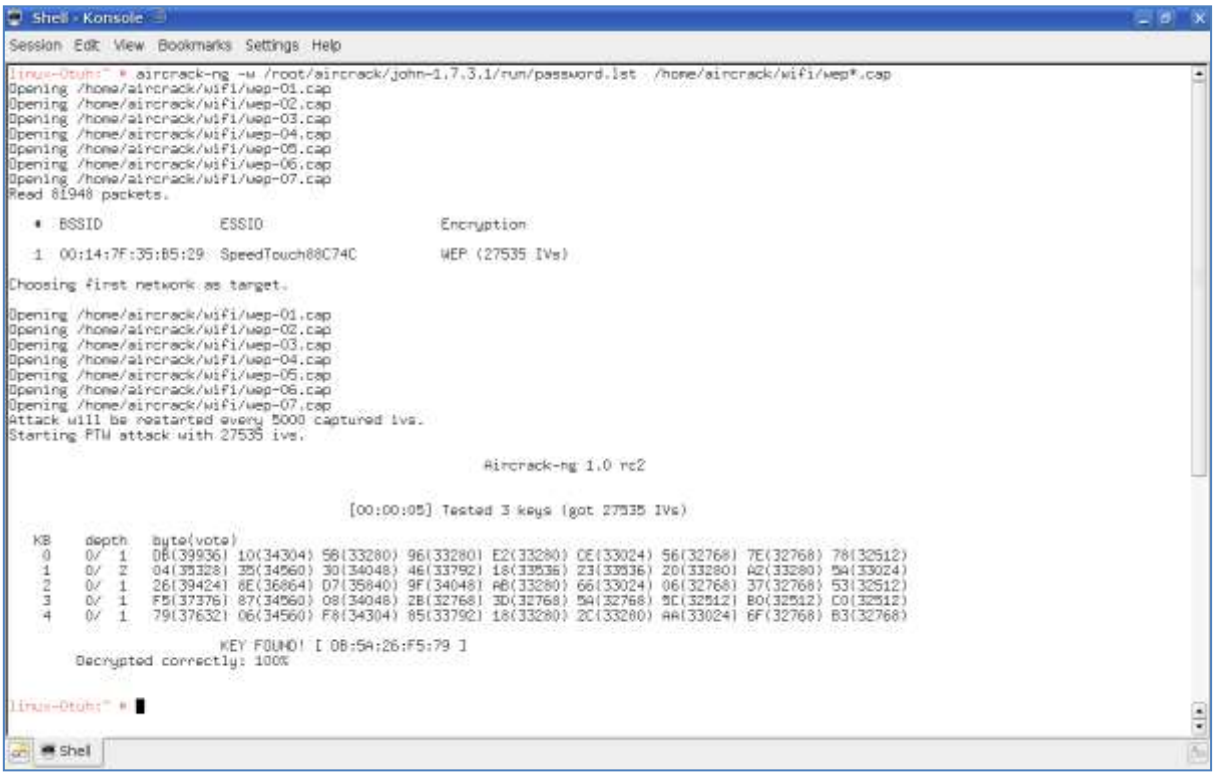

Figura 6. 30 - Resultado obtido pelo Aircrack-ng no WEP

7. Conclusões

A implementação de um sistema de segurança nunca pode ser considerada concluída, pois tem que ser uma preocupação constante. Novos tipos de ataque são desenvolvidos todos os dias podendo provocar graves danos às empresas. Daí a necessidade de o administrador de redes estar constantemente a actualizar e a melhorar o seu sistema de protecção e detecção. Para uma boa protecção é necessário eliminar o máximo possível de falsos alarmes positivos de forma que, em caso de um alerta verdadeiro, o administrador de redes possa facilmente detectar o intruso e agir conviventemente.

7.1 Objectivos realizados

Ao longo do desenvolvimento desta dissertação houve algumas alterações aos objectivos inicias tendo-se, por exemplo, optado por implementar um *honeypot* e investigar mais sobre o equipamento físico para a segurança das redes. A importância de alguns objectivos iniciais foi também redefinida para que se pudessem implementar novas funcionalidades. Assim sendo, desvalorizou-se a colocação de mais sensores em pontos estratégico e a monitorização da rede sem fios. Em termos gerais houve uma actualização do *software* e a verificação do seu funcionamento através das ferramentas correctas. Posteriormente fizeram-se as correcções e alterações necessárias para obter uma melhoria do desempenho do IDS e o seu endurecimento para melhor o proteger. Fez-se uso de TAP´s para obter um melhor conhecimento do tráfego da rede e facilitar o solucionamento dos problemas verificados pelo IDS. Foram também realizados testes de vulnerabilidades a computadores que tinham apresentado possíveis sinais de intrusões. Foi depois implementado um *honeypot* de baixa interactividade com vários serviços emulados. Tanto os IDS como o *honeypot* foram testados para verificar o seu correcto funcionamento e por sinais de vulnerabilidades. Através da ferramenta *AirCrack* testou-se a segurança das palavras-chave dos PDA existentes. Foram também realizados testes numa rede com a implementação de um IDS fazendo uso de uma TAP construída.

7.2 Resumos dos trabalhos efectuados

O IDS permitiu detectar vulnerabilidades e comportamentos suspeitos do tráfego da rede do ASC (VLAN 14, VLAN 84, Lisboa e ISA). Através dos alertas obtidos conseguiu-se corrigir algumas vulnerabilidades o que permitiu obter uma rede mais segura e com melhor desempenho. A maioria dos alertas foram corrigidos através da desactivação de alguns serviços, instalação de *patches* como também a actualização dos sistemas operativos e do antivírus. Houve a necessidade de analisar algumas máquinas, não só com o antivírus mas também com ferramentas que permitiam visualizar os serviços activos. Em casos extremos os resultados obtidos obrigaram a uma nova instalação de um sistema operativo. Foram também analisados alguns dos servidores com o antivírus e o *Nessus*, mas neste caso foi apenas

necessário uma actualização e eliminação dos vírus. Nem todos os alertas foram resolvidos, isto porque houve um maior esforço para tentar resolver os alertas de ICMP recebidos da VLAN 84 e também porque optou-se por utilizar equipamentos e ferramentas de segurança diferentes ao IDS. O IDS ficou instalado e configurado para as VLAN 14 e VLAN 84, está disponível para ser utilizado pela equipa de gestão de redes da ANA.

Os TAPS recentemente adquiridos foram estudados, testados e colocados na rede, com o IDS contribuíram para uma melhoria da segurança da rede do ASC.

Foi também implementado um *honeypot* de baixa interactividade que simula vários serviços como *telnet* e POP3.

As vantagens de um documento como a presente dissertação ficar na posse da ANA são: de haver um estudo sobre várias ferramentas de segurança de redes, como também os resultados das implementações do IDS, *honeypot* e dos vários testes efectuados à rede do ASC.

7.3 Trabalho futuro

A implementação de sistemas de segurança em redes de informática é um serviço que nunca pode ser considerado terminado, visto necessitar sempre de alterações e actualizações ao sistema implementado. Existem vários caminhos para melhorar o sistema criado, como, por exemplo, colocando mais sensores em pontos estratégicos como nos servidores. As actualizações de todos os componentes do IDS podiam ser feitos automaticamente incluindo as regras do *snort*, libertando o administrador de redes do trabalho. Aconselha-se a que as actualizações do anti-vírus seja feito automaticamente a todos computadores e, também, que o *scan* do antivírus aos computadores seja diário com uma hora predefinida, evitando que sejam os utilizadores a fazê-lo.

A utilização de um sistema só de detecção já não é suficiente, aconselhando-se a avançar para um NSM. Um NSM é basicamente a união de um IDS com um IPS. Permite uma maior quantidade e melhor qualidade de dados para análise na detecção de intrusões. Assim auxilia o administrador de redes a responder com mais eficácia à intrusão e ajuda a encontrar os estragos provocados pela mesma. No entanto na sua configuração actual nem todos os equipamento do ASC permitem o uso de um IPS.

Em vez da utilização de um *span port* para os sensores pode utilizar-se uma TAP, que permite ao administrador de redes colocar o sensor onde desejar sem haver interrupções do fluxo do tráfego ou alterações adicionais na rede. Permite também uma análise do tráfego antes do *switch* o que possibilita verificar erros na transmissão de pacotes na rede.

Foi utilizado neste trabalho um *honeypot* de baixa interactividade. Para se poder obter mais informação dos intrusos e dos seus métodos, era aconselhável a utilização de um *honeypot* de alta interactividade. Existem também *honeypots* que fazem contra-ataques mas eticamente ainda são muito controversos e pouco aconselháveis, por se poder estar a atacar uma máquina controlada por um *hacker* e não a própria máquina do *hacker*.

7.4 Apreciação final

Este estágio revelou-se um enorme desafio porque, apesar dos conhecimentos obtidos no ISEP, uma rede desta escala apresenta sempre mais dificuldade que qualquer experiência feita em laboratório. Também me permitiu aprofundar os conhecimentos sobre redes e sua segurança. Tive ainda a oportunidade não só de estar em contacto com equipamento complexo, algo que até ao início do estágio só conhecia através da leitura, mas também aplicar e melhorar os meus conhecimentos teóricos. Foi uma experiência muito enriquecedora a nível pessoal e profissional.

Referências

[1] COLE, Eric – *Hackers Beware,* 1ª Ed. New Riders Publishing; 2004.

[2] BEJTLICH, Richard – *The Tao of Network Security*, 1ª Ed. Addison Wesley, 2004.

[3] S&C Enterprises - *Hacking Secrets Revealed.*

[4] *WICHERSKI,* Georg, *Medium Interaction Honeypots*, April 7, 2006.

[5] *Symantec Global Internet Security Threat Report, Trends for July-December 07,*vol. XIII, April 2008.

[6] *Symantec Government Internet Security Threat Report, Trends for January-June 07,*vol. XII, Setembro 2007.

[7] SCOTT, Charlie, WOLFE, Paul, HAYES, Bert, *Snort for Dummies*, Wiley Publishing, 2004.

[8] YUEBIN, Bai, HIDETSUNE, Kobayashi, *Intrusion Detection Systems*, Technology Development, BeiHang University.

[9] SPITZNER, Lance, *Tracking Hackers,* Addison Wesley, 2002.

[10] SPITZNER, Lance, *Honeypots Definitions and Value of Honeypots,*2003.

[11] CASWELL, Brian, BEALE, Jay, *Snort 2.1 Intrusion Detection*, Second Edition, 2004.

[12] Technology Overview: TAPS and Span Ports, NetOptics.

[13] DERAISON, Renaud, MEER, Haroon; TEMMINGH, Roelof; van der WALT, Charl, ALDER, Raven; ALDERSON, Jimmy; JOHNSTON, Andy; THEALL, George - *Nessus Network Auditing,* 1ª Ed. Syngress, 2004.

[14] PROVOS, Niels, *A Virtual Honeypot Framework,*Center for Information Technology Integration, University of Michigan, 2004.

[15] LOUREIRO, Marco - *Gestão e Monitorização da LAN do Aeroporto Sá Carneiro*, pg. 39-43,46-48.

[16] ROESCH, Martin, *Writing Snort Rules, How To write Snort rules and keep your sanity,* versão 1.3.1.2, 2000.

[17] BIDWELL,Teri; CROSS, Michael; RUSELL, Ryan; - *Hack Proofing your Identity in the Information Age,* 1ª Ed. Syngress, 2002.

[18] POINTON, Adam, *One-Way Cable for IDS Deployment*, Sentinel Data Security, versão 0.2, 2004.

[19] CHANDRAN, Roshen; PAKALA, Sangita; *Simulating Networks with Honeyd,* Ed. 0.5, 2003.

[20] McHUGH, John, CHRISTIE, Alan, ALLEN, Julia, *Defending Yourself: The Role of Intrusion Detection Systems*, Software Engineering Institute, CERT Coordination Center.

[21] Nessus version 4.68, http://www.nessus.org/

[22] Nmap version 3.2.0, http://nmap.org/

[23] GFI Security Software, http://www.gfi.com/

[24] Top 100 Network Security Tools, http://sectools.org/

[25] Wireshark version 1.0, http://www.wireshark.org/

[26] Tcpdump version 3.6.8, http://www.tcpdump.org/

[27] Snort version 2.4.4, http://www.snort.org/

[28] Ossec hids, http://www.ossec.net

[29] Sguil, http://sguil.sourceforge.net/

[30] Kismet, http://www.kismetwireless.net/

[31] Aircrack-ng version 1.0, http://www.aircrack-ng.org/doku.php

[32] Airsnort, http://airsnort.shmoo.com/

[33] Honeyd version 1.5c, http://www.honeyd.org/

[34] Nepenthes, http://nepenthes.mwcollect.org/

[35] Symantec Internet Security Threat Report, http://www.symantec.com/business/theme.jsp?themeid=threatreport

[36] Enterasys Security Networks, http://www.enterasys.com/

[37] Sunfire Server, http://www.sun.com/servers/entry/v210/

[38] Passive Tap Documentation, http://www.cubinlab.ee.unimelb.edu.au/probing/passive_tap.php#generate_copy

[39] Password Calculator, http://lastbit.com/pswcalc.asp

[40] Brute Force Attack, http://lastbit.com/rm_bruteforce.asp

[41] One way cable white papers; http://www.sentinelsecurity.net/whitepapers/OneWayCable-original.pdf

Apêndice

A - Script para o auto-arranque do *Snort* e *Barnyard*

```
#!/bin/bash
#  -c ficheiro de configuração
#  -d detalhes do log
#  -i interface
#  -l log directório
#  -u utilizador
#  -q modo silencioso
#  -D correr em modo daemon
#  -dev

# variáveis
LOGDIR=/home/snortuser/log
# em caso seja necessario ter as horas
DATE=`date +%Y%m%d`
INTERFACE="eth0"
SNORT=/usr/local/bin/snort
SNORTCFG=/etc/snort/snort.conf

# Barnyard variables
BARNYARD=/usr/local/bin/barnyard
BYCONF=/etc/snort/barnyard.conf
BYMSG=/etc/snort/sid-msg.map
BYGEN=/etc/snort/gen-msg.map
BYCLASS=/etc/snort/classification.config
SNORTFILE=snort-unified.log
BYPID=/var/run/by.pid
```

```
WALDO=/home/snortuser/log/bylog.waldo


startsnort() {
  echo ""
  echo "Começar o Snort em eth0"
  echo ""
  $SNORT -dev -u root -q -c $SNORTCFG -i $INTERFACE -l $LOGDIR
  echo ""
  echo "Espera 5 seg antes de começar o Snort..."
  sleep 5
  echo "Atribuir as permissões..."
  chown -R root /home/snortuser/log
  echo ""
  echo "Atribuir o ficheiro ao utilizador..."
  if [ ! -e $BYPID ]; then
    touch $BYPID
    #chown root.snort $BYPID
    chown root $BYPID
  else
    #if [[ $(find $BYPID -user root -group snort) != $BYPID ]]; then
if [[ $(find $BYPID -user root ) != $BYPID ]]; then
     # chown root.snort $BYPID
 chown root $BYPID
fi
  fi
  echo ""
  echo "Starting Barnyard"
   su - root -c "$BARNYARD -c $BYCONF -s $BYMSG -g $BYGEN -p $BYCLASS -d
$LOGDIRG -f $SNORTFILE -X $BYPID -w $WALDO -n -v -v -v -v -v -v -D"
}
stopsnort() {
```

```
  SNORTPID=$(pidof -x snort)
  BARNYARDPID=$(pidof -x barnyard)
  if [ $BARNYARDPID ]; then
    echo ""
    echo "Stopping Barnyard: $BARNYARDPID"
    echo ""
    kill -9 $BARNYARDPID
  else
    echo "Barnyard não está a correr."
  fi
  if [ $SNORTPID ]; then
    # Running - kill it
    echo ""
    echo "A parar o Snort: $SNORTPID"
    echo ""
    kill -9 $SNORTPID
  else
    echo "Snort não está a correr."
  fi
}
case "$1" in
 'start')
   startsnort
   ;;
 'stop')
   stopsnort
   ;;
 'restart')
   stopsnort
   sleep 2
```

```
    startsnort
    ;;
 *)
 # start por defeito
    startsnort
esac
```

B - Ficheiro de configuração do honeypot teste

```
### Para criar um router é necessário ter um ponto de entrada,

### utiliza-se para isso o comando route entry e especifica-se

### o endereço IP do router e a rede acessível através do router.

route entry 192.168.84.47 network 192.168.164.0/24


### Para especificar os endereços IP directamente acessíveis

### pelo router(não necessita de hops), utiliza-se o route link.

route 192.168.84.47 link 192.168.164.0/24


### Para adicionar um novo router ao já existente

### na rede é necessário utilizar route add net.

### Especificar a rede acessível através do novo router e o IP do novo
router.


### Para a rede ter um aspecto mais real,

### utiliza-se o latency (ms) para aparecer que há atrasos nos pacotes
chegar ao destino,

### utiliza-se o loss (%) para modelar a perda de pacotes

### utiliza-se o bandwidth (Kbps, Mbps ou Gbps) como a largura de banda

### neste caso estas especificações serão verificadas entre os dois routers
(192.168.84.47 e 192.168.165.2)

route 192.168.84.47 add net 192.168.165.0/24  192.168.165.2 latency 50ms
loss 0.1 bandwidth 1Mbps


### Endereços de IP são atribuídos a hosts virtuais que serão depois

### simulados com o Honeyd utilizando o bind.

### Depois seraõ ligadas a um modelo configurado

### Modelo Windows 2000 com SP2

### portas abertas 80/tcp,139/tcp,137/tcp,137/udp,135/udp,1080/udp

### emuladores - iis, mydoom abre a porta do virus
```

```
### as outras portas estão fechadas e se houver ligação em caso de

### tcp será enviado um RST em caso de

### udp será enviado uma mensagem Unreachable

create windows

set windows personality "Microsoft Windows 2000 SP2"

add windows tcp port 80 "perl /home/honey/s.installed/iisemulator-
0.95/iisemul8.pl"

add windows tcp port 139 open

add windows tcp port 137 open

add windows udp port 137 open

add windows udp port 135 open

add windows tcp port 1080 "perl /home/honey/s.installed/honeyd-
1.5c/scripts/mydoom.pl -l /var/log/honeyd/windows.log"

set windows default tcp action reset

set windows default udp action reset

bind 192.168.164.51 windows

bind 192.168.165.51 windows


### Modelo linux versão 2.2.1.3

### portas abertas 80/tcp,139/tcp,137/tcp,137/udp,135/udp,4444/tcp

### emuladores - apache script, kuang 2 abre a porta do vírus

### as outras portas estão fechadas e se houver ligação em caso de

### tcp será enviado um RST

### udp será enviado uma mensagem Unreachable

create linux

set linux personality "Linux 2.2.13"

add linux tcp port 139 open

add linux tcp port 80 "sh /home/honey/s.installed/honeyd-
1.5c/scripts/apache.sh"

add linux tcp port 137 open

add linux udp port 137 open
```

```
add linux udp port 135 open

add linux tcp port 4444 "perl /home/honey/s.installed/honeyd-
1.5c/scripts/kuang2.pl -l /var/log/honeyd/linux.log"

set linux default tcp action reset

set linux default udp action reset

bind 192.168.164.52 linux

bind 192.168.165.52 linux


### Modelo Microsoft XP SP firewall enabled

### portas abertas 80/tcp,139/tcp,137/tcp,1080/tcp,137/udp,135/udp

### emuladores - iis, mydoom abre a porta do vírus

### as outras portas estão fechadas e se houver ligação em caso de

### tcp será enviado um RST

### udp será enviado uma mensagem Unreachable

create xp2

set xp2 personality "Microsoft Windows XP SP2 (firewall enabled)"

add xp2 tcp port 80 "perl /home/honey/s.installed/iisemulator-
0.95/iisemul8.pl"

add xp2 tcp port 139 open

add xp2 tcp port 137 open

add xp2 udp port 137 open

add xp2 udp port 135 open

add xp2 tcp port 1080 "perl /home/honey/s.installed/honeyd-
1.5c/scripts/mydoom.pl -l /var/log/honeyd/windows.log"

set xp2 default tcp action reset

set xp2 default udp action reset

bind 192.168.164.53 xp2

bind 192.168.165.53 xp2

### Microsoft Windows XP Pro

### portas abertas 80/tcp,139/tcp,137/tcp,1080/tcp,110/tcp,137/udp,135/udp

### emuladores - iis, pop3, mydoom abre a porta do vírus
```

```
### as outras portas estão fechadas e se houver ligação em caso de

### tcp será enviado um RST

### udp será enviado uma mensagem Unreachable

create pro

set pro personality "Microsoft Windows XP Pro"

add pro tcp port 80 "perl /home/honey/s.installed/iisemulator-
0.95/iisemul8.pl"

add pro tcp port 139 open

add pro tcp port 137 open

add pro tcp port 110 "sh /home/honey/s.installed/honeyd-
1.5c/scripts/pop3.sh"

add pro tcp port 1080 "perl /home/honey/s.installed/honeyd-
1.5c/scripts/mydoom.pl -l /var/log/honeyd/windows.log"

add xp2 udp port 137 open

add xp2 udp port 135 open

set pro default tcp action reset

set pro default udp action reset

bind 192.168.164.54 pro

bind 192.168.165.54 pro


### Para o modelo dos routers foi utilizado o Cisco IOS

### com o emulador de telnet na porta 23

create router

set router personality "Cisco IOS 11.3 - 12.0(11)"

set router default tcp action reset

set router default udp action reset

add router tcp port 23 "/usr/bin/perl /home/honey/s.installed/honeyd-
1.5c/scripts/router-telnet.pl"

set router uid 32767 gid 32767

set router uptime 1327650

bind 192.168.84.47 router

bind 192.168.165.2 router
```

B.1 - Ficheiro de configuração do honeypot

```
### Endereços de IP são atribuídos a hosts virtuais que serão depois

### simulados com o Honeyd utilizando o bind.

### Depois seraõ ligadas a um modelo configurado

### Modelo Windows 2000 com SP2

### portas abertas 21/tcp, 23/tcp, 80/tcp, 110/tcp, 139/tcp, 137/tcp,
137/udp, 135/udp, 1080/udp

### emuladores - iis, mydoom, telnet do router cisco, pop3, ftp.

### as outras portas estão fechadas e se houver ligação em caso de

### tcp será enviado um RST em caso de

### udp será enviado uma mensagem Unreachable

create windows

set windows personality "Microsoft Windows 2000 SP2"

add windows tcp port 80 "perl /usr/local/share/honeyd/scripts/iisemulator-
0.95/iisemul8.pl"

add windows tcp port 139 open

add windows tcp port 137 open

add windows udp port 137 open

add windows udp port 135 open

add windows tcp port 1080 "/usr/bin/perl
/usr/local/share/honeyd/scripts/proxy.pl -l /var/log/honeyd.log"

add windows tcp port 23 "/usr/bin/perl
/usr/local/share/honeyd/scripts/router-telnet.pl -l /var/log/honeyd.log"

add windows tcp port 110 "/usr/bin/sh
/usr/local/share/honeyd/scripts/emulate-pop3.sh -l / var/log/honeyd.log "

add windows tcp port 21 "/usr/bin/sh /usr/local/share/honeyd/scripts/ftp.sh
-l /var/log/honeyd.log "

set windows default tcp action reset

set windows default udp action reset

bind 192.168.14.141 windows
```

C - Esquema da estrutura da rede dados do ASC

Nota: Esquema não incluído por razões de segurança

D - Esquema da estrutura da rede de fibra óptica do ASC

Nota: Esquema não incluído por razões de segurança

E - Esquema da estrutura da rede do VoIP do ASC

Nota: Esquema não incluído por razões de segurança