CENTRO UNIVERSITÁRIO DE BRASÍLIA - UNICEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS - FATECS
CURSO: ENGENHARIA DE COMPUTAÇÃO

ESTUDO DE CASO DE E-MAIL UTILIZANDO AS TECNOLOGIAS SPF E DKIM

RODRIGO LOURENÇO DA SILVA
RA 2012653/5

PROFESSOR ORIENTADOR: MARCO ANTÔNIO DE OLIVEIRA ARAÚJO

Brasília/DF, julho de 2008.

RODRIGO LOURENÇO DA SILVA

ESTUDO DE CASO DE E-MAIL UTILIZANDO AS TECNOLOGIAS SPF E DKIM

Trabalho apresentado à banca examinadora do curso de Engenharia da Computação da FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas do UniCEUB – Centro Universitário de Brasília, como requisito parcial para obtenção do título de Engenheiro da Computação.

Prof. Orientador: MSc. Marco Antônio.

Brasília/DF, julho de 2008.

RODRIGO LOURENÇO DA SILVA

ESTUDO DE CASO DE E-MAIL UTILIZANDO AS TECNOLOGIAS SPF E DKIM

Trabalho apresentado à banca examinadora do curso de Engenharia da Computação da FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas do UniCEUB – Centro Universitário de Brasília, como requisito parcial para obtenção do título de Engenheiro da Computação.

Prof. Orientador: MSc. Marco Antônio.

BANCA EXAMINADORA

_____
Prof. Marco Antônio de Oliveira Araújo
Orientador

_____
Prof. Fabiano Mariath D' Oliveira
Examinador

_____
Prof. Roberto Schaefer de Azeredo
Examinador

Brasília/DF, julho de 2008.

DEDICATÓRIA

A Jesus primeiramente,

Por sempre estar ao meu lado, me orientando em cada passo de minha vida.

À Minha família,

Que me deu a base necessária para a formação do meu caráter, independência e educação, me orientando sempre a seguir pelo caminho correto, valorizando minhas idéias e respeitando meus objetivos de vida.

AGRADECIMENTOS

A Jesus,

Agradeço por tudo o que tem feito em minha vida, me guiando em todos os momentos e iluminando meus pensamentos para que eu possa tomar decisões sábias e frutificadoras.

À Família,

Meus pais Pedro e Edi e meus irmãos Ricardo e Adriana, por estarem ao meu lado durante todo o tempo, sempre desejando meu sucesso pessoal e profissional.

Ao meu professor Lucas Brasilino, que me auxiliou durante toda a trajetória do meu projeto, sempre me motivando nas horas de exaustão e desânimo, me trazendo de volta a coragem e força para continuar.

Aos meus amigos todos,

Em especial Rafael Gomes e Thiago Azevedo, que tiveram sempre paciência para esclarecer minhas dúvidas, me auxiliando e propondo melhorias para o projeto.

À Karine,

Minha namorada, que esteve ao meu lado em ocasiões muito importantes e marcantes da minha vida, me trazendo sempre grande alegria com sua presença.

"O único homem que está isento de erros, é aquele que não arrisca acertar."
(Albert Einstein)

SUMÁRIO

LISTA DE QUADROS

LISTA DE FIGURAS

LISTA DE TABELAS

ÍNDICE DE SIGLAS E ABREVIATURAS

A	*Address Record* ou Registro de Endereçamento
BCC	*Blind Carbon Copy* ou Com Cópia Oculta
CC	*Carbon Copy* ou Com Cópia
DKIM	*DomainKeys Identified Mail* ou Sistema de Correio Identificado Por Chaves-de-Domínio
DNS	*Domain Name System* ou Sistema de Nome de Domínio
GUI	*Graphical User Interface* ou Interface Gráfica de Usuário
HTML	*Hypertext Markup Language* ou Linguagem de Marcação de Hipertexto
HTTP	*Hypertext Transfer Protocol* ou Protocolo de Transferência de Hipertexto
IMAP	*Internet Message Access Protocol* ou Protocolo de Acesso de Mensagens da Internet
IP	*Internet Protocol* ou Protocolo da Internet
ISP	*Internet Service Provider* ou Provedor de Serviços da Internet
MAA	*Mail Access Agent* ou Agente de Acesso de E-mail
MDA	*Mail Delivery Agent* ou Agente de Entrega de E-mail
MRA	*Mail Retrieval Agent* ou Agente de Recuperação de E-mail
MSA	*Mail Submision Agent* ou Agente de Submissão de E-mail
MTA	*Mail Transport Agent* ou Agente de Transporte de E-mail
MUA	*Mail User Agent* ou Agente de Usuário de E-mail
MX	*Mail Exchange Record* ou Registro de E-mail
NS	*Name Server Record* ou Registro de Nome do Servidor
POP	*Post Office Protocol* ou Protocolo de Escritório de Correio
PTR	*Pointer Record* ou Registro Reverso
RBL	*Real Time Black List* ou Lista Negra em Tempo Real

RFC	*Request for Comments*
RSA	Rivest, Shamir and Adleman *Algorithm*, Algoritmo de Rivest, Shamir e Adleman
SHA	*Secure Hash Algorithm* ou Algoritmo de Segurança *Hash*
SMTP	*Simple Mail Transfer Protocol* ou Protocolo de Transferência de Correio Simples
SPF	*Sender Policy Framework* ou Estrutura de Política de Remetente
TCP	*Transmission Control Protocol* ou Protocolo de Controle da Transmissão
TXT	*Text Record* ou Registro de Texto
UCE	*Unsolicited Commercial E-mail* ou Mensagem Comercial não Solicitada
URL	*Uniform Resource Locator* ou Localizador Uniforme de Recursos
WWW	*World Wide Web* ou Rede de Alcance Mundial

RESUMO

A idéia-chave deste projeto final é a análise do comportamento das ferramentas SPF e DKIM, que atuam no combate à falsificação de remetentes de correio eletrônico. Para tal, foi realizado um estudo de caso destas tecnologias, visando comparar a atuação de cada uma delas mediante determinadas situações. Os resultados coletados com a realização deste estudo permitiram conhecer em detalhes estas tecnologias, facilitando administradores de sistemas a estabelecer critérios de comparação que possam ajudar futuras empresas na escolha de uma delas como solução alternativa de segurança de e-mail. Portanto, um estudo sobre estas ferramentas se mostra interessante para profissionais de segurança, já que ambas representam soluções livres e visam aumentar o nível de segurança de e-mail em um ambiente de rede através do conhecimento da real identidade de um domínio de origem ou de uma referida pessoa que diz autoridade remetente no transporte de uma mensagem de correio.

Palavras-chave: estudo; ferramentas; solução alternativa; segurança; soluções livres; identidade; origem.

ABSTRACT

The key idea of this final paper is the analysis of the behavior of SPF and DKIM tools, that act in the combat to the fake of e-mail senders. For such, it was made a study of case of these tools, with the intention to compare the performance of each one of them by means of determined situations. The collected results with the accomplishment of this study allowed knowing in details these technologies, to make easy system administrators to establish comparison criteria that can help future companies in the choice of one of them as alternative solution of e-mail security. Therefore, a study on these tools it shows interesting for security professionals, since both represent free solutions and they objectify to increase the level e-mail security in a net environment through the knowledge of the real identity of an origin domain or one related person who says sender authority in the transport of a post office message.

Key-Words: study; tools; alternative solution; security, free solutions, identity, origin.

CAPÍTULO 1. INTRODUÇÃO

1.1 MOTIVAÇÃO/IDENTIFICAÇÃO DO PROBLEMA

Atualmente empresas de pequeno porte têm apresentado dificuldades na implementação de seus serviços de rede com segurança, tanto pelo fato de não disporem de profissionais específicos responsáveis pelo setor de informática ou área de tecnologia da informação, como também pela falta de orientação e conhecimento adequado para a respectiva solução de serviço de rede. Um exemplo claro disto é o serviço de entrega e recebimento de correio eletrônico (e-mail).

Os sistemas de correio evoluíram de uma forma tão grande que são poucas as empresas que não possuem uma implementação de serviço de e-mail. Entretanto, são muitas as empresas que não possuem uma implementação de serviço de e-mail segura. O grande diferencial é que, para as pequenas empresas, este serviço muitas vezes nem mesmo é implementado por falta de pessoal capacitado. Este fator as remete a duas opções de solução: ou contratam outras empresas especializadas em serviços de rede para a implantação dos serviços de correio corretamente configurados, inclusos custos de licenciamento de *software* e manutenção do sistema após o seu funcionamento, ou utilizam servidores de e-mail da *Internet* gratuitos, alocados remotamente, para gerenciar suas contas corporativas.

A primeira opção de solução torna-se inviável para uma empresa de pequeno porte, já que o capital disponível para a mesma não prevê maiores gastos. A segunda opção de solução compromete a segurança e a confiabilidade: contas de e-mail gratuitas estão associadas a muita fraude na *Internet*. Portanto, adotar um e-mail corporativo com o sufixo[1] de um servidor de e-mail grátis (exemplo: <nome_da_empresa>@yahoo.com.br), além de não transmitir uma "boa imagem" desta empresa, pode causar o bloqueio da maioria das mensagens enviadas por este servidor, já que muitos servidores destinatários estão configurados para descartar mensagens oriundas destes endereços. Além disto, o funcionamento do correio eletrônico ficaria submisso às limitações dos provedores destes serviços, como o número de mensagens que se pode enviar simultaneamente, bem como o limite de envio diário de mensagens. Estas limitações são

[1] Sufixo, em informática, refere-se ao nome que identifica o domínio ou parte após o @ do endereço de e-mail.

impostas para se prevenir contra remetentes de *spams*[2], que utilizam contas de usuários cadastrados nestes servidores para o envio de mensagens em massa a outros servidores de e-mail. Por isto, há a necessidade de conhecimento da identidade do remetente, já que somente *spammers*[3] deveriam estar sujeitos a estas limitações. Lembrando que *spammers* também podem utilizar seus próprios servidores de e-mail pessoais para propagarem suas mensagens indesejáveis pela *Internet*. Em decorrência destas duas opções de solução e as desvantagens de ambas, a utilização de *softwares* piratas para o gerenciamento de correio acaba sendo considerada outra opção de solução, considerando os fatores custos X segurança/confiabilidade.

A grande problemática recentemente vinculada ao serviço de correio eletrônico está em comprovar a autenticidade da origem de uma mensagem. As tecnologias no combate aos e-mails forjados são, em sua maioria, limitadas. Estes e-mails degradam o serviço de correio eletrônico, consumindo alto percentual da banda contratada e recursos computacionais como memória e processamento, diminuindo o desempenho de um servidor e prejudicando o funcionamento de uma rede.

As técnicas atuais usadas para se garantir a segurança de correio eletrônico, na maior parte dos casos, geram falhas e de forma alguma representam um mecanismo de proteção ideal. Exemplos destas falhas é a ocorrência de falso-positivos e falso-negativos. Falso-positivos são e-mails reais que foram classificadas como *spams* e falso-negativos *spams* que foram classificados como e-mails legítimos. As tecnologias SPF e DKIM visam diminuir os índices de falso-positivos e falso-negativos gerados por outras ferramentas de segurança consumindo uma demanda baixa de recursos computacionais, diferentemente do percentual computacional gasto por uma análise heurística[4] completa para filtragem da mensagem. Lembrando que um *spam* não necessariamente se originou de um remetente forjado, assim como nem toda origem falsificada representa um *spam*. Entretanto, como a maioria dos e-mails decorrentes da falsificação de remetentes são *spams*, para o conceito desta monografia, o termo *spam* estará se referindo a um e-mail falsificado.

---

[2] *Spams* são mensagens enviadas que não foram solicitadas.

[3] *Spammers* são pessoas que atuam na propagação de *spams*.

[4] *Análise Heurística* refere-se à verificação de uma mensagem baseada em características da mesma, onde é atribuída uma pontuação de acordo com cada característica, e a soma dos pontos classifica a mensagem como *spam* ou não.

1.2 OBJETIVOS

O objetivo geral do projeto é a realização de um estudo que possibilite um conhecimento aprofundado sobre as tecnologias SPF e DKIM, permitindo enquadrá-las como soluções alternativas de segurança de correio, já que representam ferramentas livres e por isto dispensam custos relativos ao licenciamento de *software*. Os objetivos específicos são:

a) Montagem de ambiente de e-mail. Será montado um ambiente de e-mail virtualizado para simulação da estrutura de rede física do projeto.

b) Implementação de solução alternativa. O ponto de partida é o conhecimento e análise das ferramentas SPF e DKIM. Estas ferramentas não objetivam acabar definitivamente com o problema dos falso-positivos e falso-negativos, porque não foram projetadas para esta finalidade, mas sim para deixar o administrador decidir quem deve acessar ou não um domínio. (verificação do remetente da mensagem). Entretanto, pode-se afirmar que a utilização destas ferramentas prevê uma redução da taxa de falso-positivos e falso-negativos quando comparadas aos índices gerados por outras tecnologias de combate à falsificação de remetentes.

c) Realização do estudo de caso das ferramentas SPF e DKIM. Serão realizados testes para se saber a atuação de cada ferramenta mediante simulações de forjamento e de envio de e-mails legítimos. O resultado dos testes ajudará administradores de sistema a conhecer detalhadamente estas soluções, e saber como cada uma delas se comporta mediante determinada situação.

1.3 METODOLOGIA

O desenvolvimento do projeto começou com pesquisas bibliográficas para definir com quais tecnologias a implementação seria elaborada. Após isso, deu-se início à elaboração da parte escrita referente ao desenvolvimento teórico. Em paralelo a essa escrita, foi criado o ambiente de rede virtual necessário para a elaboração do projeto. Com a arquitetura de rede pronta, foram instaladas as ferramentas e configurados os serviços associados às mesmas que proporcionassem a realização do estudo de caso.

Cada uma destas tecnologias foi analisada separadamente. Terminada a fase de instalação e configuração da ferramenta SPF, deu-se início à fase de testes e experimentação para coleta de resultados da mesma. O mesmo procedimento foi realizado com a ferramenta DKIM. Com os testes e experimentação dos aplicativos finalizados, foi possível concluir a parte escrita da monografia referente aos testes e análise de resultados, utilizando os dados obtidos e tiradas as devidas conclusões.

1.4 ESTRUTURA DA MONOGRAFIA

Este trabalho foi elaborado em seis capítulos, incluindo a introdução. Segue uma breve descrição do que será exposto nos capítulos seguintes:

Capítulo 02 – A Mensagem Eletrônica: Traz conceitos-chave acerca do funcionamento de um sistema de e-mail, informações sobre o surgimento do e-mail (história) e a infra-estrutura de correio eletrônico (infra-estrutura de transporte de e-mail, formato de uma mensagem e infra-estrutura de e-mail na *Internet*).

Capítulo 03 – Segurança de e-mail: Descreve conceitos relativos à segurança de e-mail. São esplanadas as principais ameaças externas decorrentes da falsificação de remetentes, bem como alguns métodos de defesa contra estas ameaças (inclusos como estratégia de defesa o funcionamento das ferramentas SPF e DKIM).

Capítulo 04 – Infra-estrutura do projeto: Neste capítulo são abordados conceitos sobre as especificações técnicas para a elaboração da implementação do projeto, como o ambiente de rede virtual (cenários montados), a descrição do *hardware* e *software* utilizado, os processos de instalação e configuração dos servidores de e-mail e os serviços de autenticação de remetentes associados aos mesmos para coleta de resultados.

Capítulo 05 – Testes e Análise de Resultados: Mostra os testes e experimentação realizados e os resultados obtidos com o estudo das ferramentas.

Capítulo 06 – Conclusões: Apresenta as considerações finais do projeto com base nos resultados obtidos e sugestões para trabalhos futuros.

CAPÍTULO 2. A MENSAGEM ELETRÔNICA

2.1 HISTÓRIA DO E-MAIL

Todos os dias, milhares de mensagens são trocadas por usuários da *Internet* com o intuito de se estabelecer uma comunicação ou troca de informações. Todavia, a idéia de se estabelecer uma comunicação eletrônica é anterior ao surgimento da *Internet* e foi um dos motivos que impulsionou o surgimento da mesma.

Apesar da história do e-mail ter a sua origem baseada em informações imprecisas, o primeiro relato sobre o surgimento do correio eletrônico ocorreu, de fato, em 1965, no qual um computador de grande porte estabelecia uma comunicação entre seus múltiplos usuários e permitia, desta forma, uma troca de mensagens entre os mesmos. Entretanto, a idéia de se trocar mensagens utilizando um sistema multiusuário era limitada quando se analisava o conceito de rede e suas demais utilidades, visto que no ano seguinte (1966) foi implantado o sistema *AUTODIN*, o qual possibilitava que usuários situados em diferentes máquinas estabelecessem uma comunicação. Novamente não se tem certeza da origem deste fato, pois outro sistema (*SAGE)* já poderia realizar esta mesma funcionalidade algum tempo antes. [WIKIPEDIA2007]

A primeira idéia sólida sobre o surgimento do e-mail ocorreu em 1971. Ray Tomlinson, engenheiro contratado pela *Bolt Beranek and Newman* (BBN), empresa que havia recebido do governo norte-americano a tarefa de construir a estrutura básica da *ARPANET* [5], compilou um programa capaz de enviar pequenas mensagens eletrônicas. Inicialmente, ele utilizou uma aplicação específica para estabelecer somente a comunicação interna da *ARPANET* (*SNDMSG* - contração da expressão "send message"), que permitia a um usuário deixar uma mensagem de texto a outro usuário situado na mesma máquina ou *host*[6]. Porém, com o aprimoramento de seus estudos, o pesquisador somou as funcionalidades do aplicativo *SNDMSG* com o *READMAIL*, outro aplicativo também de seu desenvolvimento usado para a leitura da mensagem. Entretanto, este conjunto de aplicações restringia-se somente ao compartilhamento de arquivos de texto em uma máquina local. Assim, Ray trabalhou em um protocolo de transferência de arquivos para estabelecer uma comunicação entre computadores

[5] *Arpanet* foi a rede militar que serviu de base para a criação da Internet

[6] *Host*, em informática, refere-se a qualquer máquina ou computador conectado a uma rede.

situados em rede. O protocolo utilizado foi o *CPYNET* e através da vinculação deste com os dois programas citados anteriormente, Ray conseguiu enviar uma mensagem para um computador remoto, mais especificadamente para seus colaboradores, anunciando sua criação. Ele também estabeleceu o uso do sinal @ para fazer a distinção entre o nome do usuário e o nome da máquina ou domínio[7] correspondente. Dessa forma, o uso do e-mail rapidamente se expandiu, prova esta que dois anos após sua criação, o serviço de correio eletrônico era responsável por setenta e cinco por cento de todo o tráfego de dados da *ARPANET*. [WIKIPEDIA2007]

Inicialmente, o objetivo da *ARPANET* era estabelecer somente uma comunicação entre máquinas remotas. Entretanto, com o avanço das tecnologias, e aprofundamento de conhecimentos tanto por parte de administradores como também pelos próprios invasores de sistemas, a segurança passou a ser o foco principal e necessidade fundamental para se estabelecer um vínculo de comunicação eletrônica, já que o e-mail passou a ser o principal alvo dos atacantes.

2.2 INFRA-ESTRUTURA DE E-MAIL

2.2.1 Infra-Estrutura de Transporte de E-mail

A infra-estrutura de transporte de e-mail fundamenta-se com o uso de determinados *softwares* agentes que funcionam como emissores e/ou receptores de mensagens. Em suma, os agentes são responsáveis por todo o envio e recebimento de mensagens de correio, sejam elas locais, ou externas (*Internet*). Segundo RFC 2821, são eles:

2.2.1.1 Agente de Usuário de E-mail

Um agente de usuário (*Mail User Agent* ou acrônimo MUA) é um aplicativo usado para o envio e recebimento de e-mails, ou seja, é o programa cliente que permite ao usuário ler e compor seu e-mail. São exemplos, dentre outros, de agentes de usuários: *KMail, Sylpheed, Elm, Mutt, Pine, Mozilla Thunderbird, Outlook e Outlook Express.*

[7] Domínio é um nome que serve para localizar e identificar uma rede na estrutura da *Internet*.

2.2.1.2 Agente de Transporte de E-mail

Um agente de transporte (*Mail Transport Agent* ou acrônimo MTA) é o programa responsável pela transferência de e-mail de uma máquina a outra. Refere-se à máquina servidora de e-mail. Ele pode receber as mensagens de outro agente de transporte, de um agente de submissão de e-mail (MSA) ou diretamente de um MUA. São exemplos dos principais MTAs: *Courier-MTA, Sendmail, Postfix, Qmail, Exim e Microsoft Exchange Server*[8].

2.2.1.3 Agente de Entrega de E-mail

O agente de entrega de e-mail (*Mail Delivery Agent* ou acrônimo MDA) é o *software* que o MTA utiliza para armazenar as mensagens dos usuários em sua caixa-postal. O MDA recebe o e-mail do MTA e o deposita na caixa de correio do usuário. Exemplos de MDAs são: *Procmail, Maildrop e Microsoft Exchange Server*.

2.2.1.4 Agente de Recuperação de E-mail

O agente de recuperação de e-mail (*Mail Retrieval Agent* ou acrônimo MRA) é responsável por fazer o *download* das mensagens do MTA. Ele estabelece a conexão com o servidor, buscando as mensagens em MTA remoto e as enviando para um MUA local. Tal componente pode ser chamado também de MAA (*Mail Acess Agent*). Exemplos: *Fetchmail, Getmail e Retchmail.*

2.2.1.5 Agente de Submissão de E-mail

Uma medida de segurança bastante eficaz contra ameaças externas consiste em reservar a porta 25/TCP somente para troca de mensagens entre MTAs e usar a porta 587/TCP para mensagens enviadas por um cliente para o seu próprio MTA. Costuma-se usar o termo MSA (*Mail Submission Agent* ou acrônimo MSA) para o MTA configurado por responder pela porta 587/TCP.

A figura a seguir descreve os componentes de um sistema de correio durante o transporte de uma mensagem de e-mail:

[8] O Microsoft Exchange Server é um servidor de e-mail que atua como um todo, englobando desta forma, as funcionalidades de MTA e MDA.

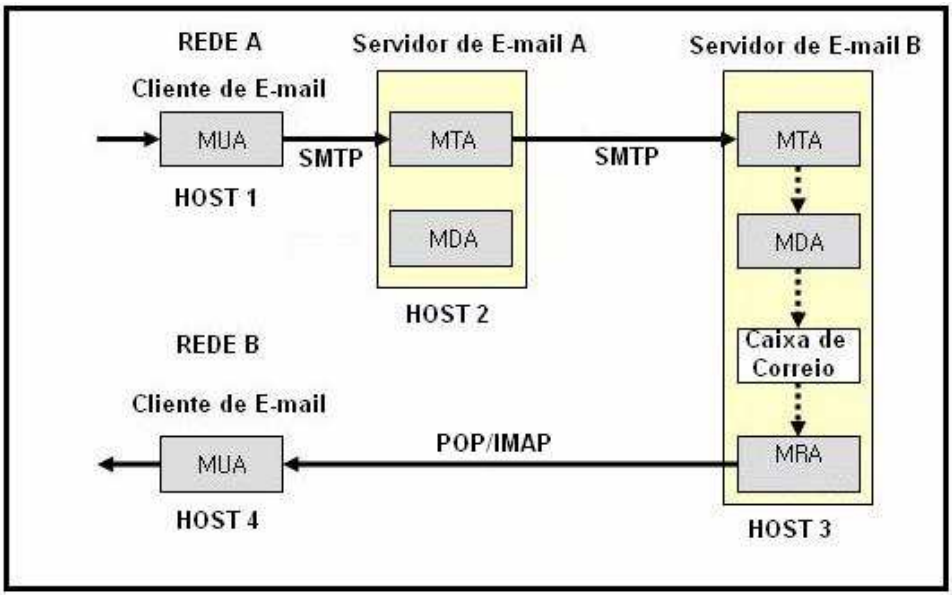

Figura 2.1 – Componentes de um sistema de correio eletrônico.

Fonte: Elaborado por Autor [2008].

Como mostra a figura, a correspondência é gerada por um MUA de origem (*HOST 1*), que repassa a mensagem para o seu MTA (*HOST 2*) via protocolo SMTP (os protocolos envolvidos no sistema de correio eletrônico serão detalhados mais adiante). Este, por sua vez, tem a função de encaminhar a mensagem para o usuário destino, caso este usuário se encontre na mesma rede (neste caso o MDA do *HOST 2* faz a entrega), ou repassá-lo para outro MTA, caso o destinatário não se encontre na mesma rede (este transporte ocorre via SMTP também). Assim, quando a mensagem chega ao MTA destinatário responsável (*HOST 3*), este a repassa para o seu MDA, que se encarrega de entregar a correspondência na caixa de correio do usuário. O MUA final (*HOST 4*) tem a função de recuperar a correspondência, bem como de passá-la para uma interface gráfica legível ao entendimento do usuário (GUI – Interface Gráfica de Usuário). Logo, ele recupera a informação diretamente da caixa de correio ou entra em contato com um MRA através do protocolo POP ou IMAP.

2.2.2 Formato de Uma Mensagem de E-mail

Uma mensagem de correio-eletrônico é composta por três partes [JUNIOR2007]:

a) *1ª parte – Envelope:* Este não é visualizado pelo usuário final. É utilizado pelos MTAs para se saber o destino do pacote, ou seja, para quem a mensagem deverá ser encaminhada, ou para qual servidor ela deverá ser repassada.

O conteúdo da mensagem refere-se ao cabeçalho e o corpo da mesma:

b) *2ª parte – Cabeçalho:* Contém o remetente, destinatário e outras informações de controle sobre a mensagem. Um cabeçalho é composto por uma sequência de linhas (*strings* contendo um ou mais *bytes*) consistindo de pares de chaves. Uma chave consiste em um campo, delimitado por *strings* separadas por dois pontos, os quais separam o nome e, na maioria das vezes, o valor respectivo a este nome. Detalharemos os principais campos.

- *From* (De) – Refere-se ao remetente da mensagem.
- *To* (Para) – Refere-se ao destinatário da mensagem. Este destinatário pode ser uma pessoa física, empresa ou lista para a qual se deseja enviar a mensagem.
- *CC* (Com Cópia) – Do inglês *Carbon Copy*, este campo serve para enviar uma cópia do documento para o destinatário especificado. Caso haja mais de um destinatário neste campo, o mesmo sabe que os outros destinatários também receberam a mensagem.
- *BCC* (*Blind Carbon Copy*) ou *CCO* (Com Cópia Oculta) – A única diferença deste campo em relação ao CC é exatamente a privacidade do destinatário. Por exemplo, quando se envia uma mensagem para *FULANO* usando o campo TO e *CICLANO* usando o campo BCC, *CICLANO* não saberá que a mensagem foi enviada também para FULANO, pois não ficará visível.
- *Attachment* (Anexo) – Este campo foi criado para se anexar algum tipo de arquivo à mensagem, seja ele um vídeo, uma imagem, uma apresentação de *slides* etc.
- *Date* (Data) – Refere-se ao dia, hora e ano da mensagem.
- *Reply* (Resposta) – Refere-se ao campo destinado para retornar uma mensagem recebida de referente remetente.
- *Forward* (Encaminhamento) – É o campo usado para se encaminhar uma mensagem e reenviá-la para outro/s destinatário/s.

A figura a seguir mostra o mínimo requerido para um cabeçalho de e-mail:

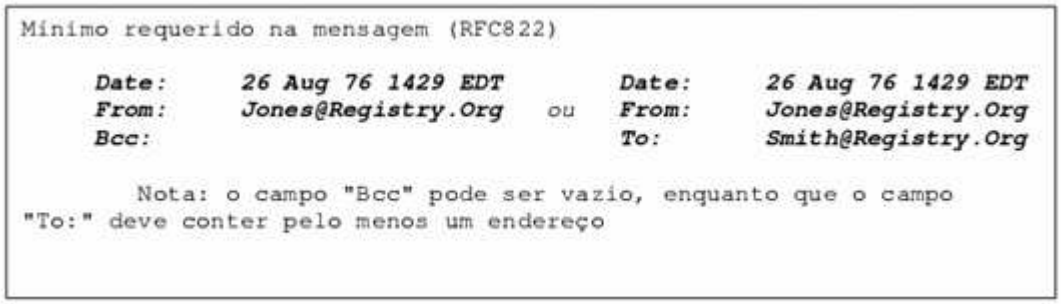

Figura 2.2 – Campos mínimos requeridos para o cabeçalho de e-mail.

Fonte: Elaborado por CROCKER [p. 38, 1982]

c) *3ª parte – Corpo:* Contém a mensagem propriamente dita. O corpo é separado do cabeçalho por uma linha em branco.

A próxima figura mostra a estrutura completa de uma mensagem de e-mail:

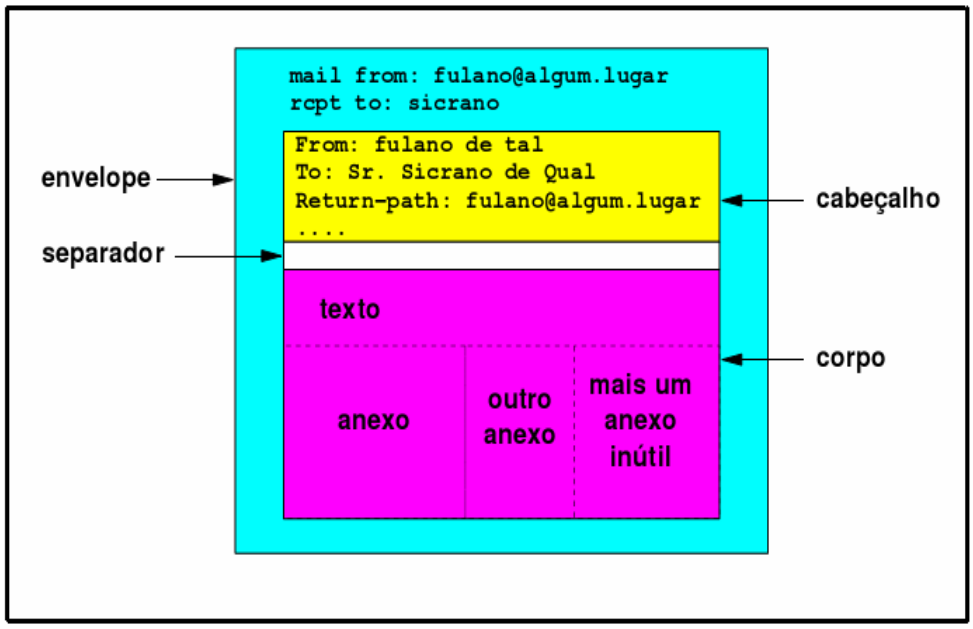

Figura 2.3 – Estrutura completa de uma mensagem de e-mail.

Fonte: Elaborado por NUNES [p. 6, 2006].

2.2.3 Infra-Estrutura de E-mail na Internet

A estrutura de transporte de e-mail na *Internet* segue o modelo cliente-servidor. A figura 2.4 mostra o caminho percorrido por uma mensagem de e-mail típica da *Internet*. Baseado nela, quando o usuário *USER1* da máquina *HOST1* pertencente ao domínio *DOMAIN1* quer enviar uma mensagem para o usuário *USER2* referente à máquina *HOST2* pertinente ao

domínio *DOMAIN2*, o *software* cliente instalado no *HOST1* se comunica com o *software* servidor residente na máquina *SERVIDOR DE CORREIO 1*. Desse modo, o servidor valida o *USER1* como membro pertencente ao seu domínio, e encaminha o pacote para a máquina servidora do outro domínio (*SERVIDOR DE CORREIO 3*). O *software* servidor presente nesta máquina verifica se o usuário destino pertence ao seu domínio, e encaminha a mensagem para a máquina cliente (*USER2*). Esta, por sua vez, utiliza seu *software* cliente para processar a mensagem e a imprime na tela do monitor. Notoriamente, o pacote passa por outras máquinas servidoras durante a trajetória de comunicação entre dois servidores de e-mail principais (*SERVIDOR DE CORREIO 2*). Estes computadores são denominados servidores de *relay*. Nestes, nem o remetente e nem o destinatário são usuários de seu domínio. Segue ilustrado abaixo o modelo de transporte de e-mail na *Internet*:

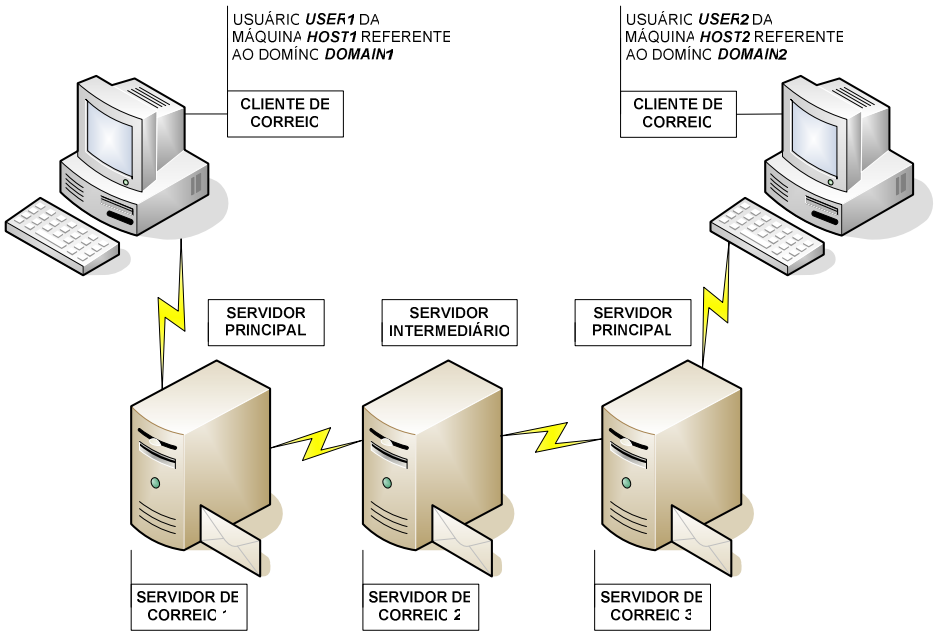

Figura 2.4 – Modelo de transporte de e-mail na Internet.

Fonte: Elaborado por Autor [2008].

2.2.3.1 O Papel do DNS

A localização de um servidor de e-mail é feita através de uma consulta ao servidor DNS (Sistema de Nome de Domínio), que, através de um registro MX[9], diz à máquina que realiza a pesquisa qual é o servidor de e-mail referente ao domínio consultado, com o intuito de

[9] *Registro MX* refere-se ao registro DNS que localiza o servidor de e-mail responsável pelo domínio pesquisado.

se saber o destino final da mensagem. Por isto, é fundamental que o serviço de resolução de nomes e e-mail estejam vinculados.

2.2.3.2 Protocolo de Transferência de Correio Simples

O SMTP (*Simple Mail Transfer Protocol*) é o protocolo padrão de comunicação usado para o envio de mensagens.

Segundo Bäck [2007, p. 7]:

"Seu propósito é transmitir mensagens de e-mail entre dois computadores. Esses computadores podem ser ambos servidores ou um deles pode ser uma máquina cliente em que os usuários executam o aplicativo de e-mail – Outlook, Thunderbird, Eudora ou qualquer outro. Para coletar novas mensagens, o usuário final não utiliza o SMTP. E aí que entram o Post Office Protocol (POP) e o Internet Message Acess Protocol (IMAP)."

2.2.3.3 Protocolos de Recebimento de E-mail

Para se recebimento de correio, são utilizados os protocolos POP e IMAP.

No POP, as mensagens são processadas pelo servidor principal, armazenadas, e posteriormente, transferidas para o receptor local. O POP pode operar somente transferindo as mensagens para o cliente remoto e descarregando sua base de dados ou transferindo as mesmas para o cliente, porém, fazendo uma cópia para ser armazenada no servidor.

Com a utilização do IMAP, as mensagens ficam armazenadas no servidor e, quando o usuário necessita ler uma mensagem, é feita uma cópia do documento e enviado para a máquina local do usuário. Nota-se que esta cópia vai para a área de arquivos temporários do usuário e é apagada algum tempo depois da mensagem ser lida pelo mesmo. A figura a seguir representa um quadro comparativo dos protocolos POP e IMAP, conceituando suas principais diferenças.

POP	IMAP
E-mails precisam ser baixados do servidor para os computadores clientes para permitir a visualização dos mesmos, podendo ser apagados do servidor, dependendo das configurações do programa cliente. a) Necessidade de se baixar o e-mail novamente quando se utiliza outro computador cliente para checagem de mensagens. b) Marcações nas mensagens, bem como apagamento das mesmas são modificadas no computador cliente, o que acarreta uma confusão entre e-mails lidos e não lidos quando se utiliza mais de um computador para checagem de correio.	O IMAP funciona com um espelho, refletindo o conteúdo do servidor para o programa cliente. Assim, os e-mails são mantidos no servidor e as modificações feitas no cliente, como apagamento ou marcações nas mensagens, são alteradas no servidor. a) Não há a necessidade de se baixar o e-mail novamente quando se utiliza outro computador para checagem do mesmo. b) Fácil distinção entre e-mails lidos e não-lidos.
Todas as mensagens, bem como seus anexos são baixadas para o computador cliente durante a checagem de novas mensagens (“new check mail process”)	A mensagem só será baixada por completa se estiver habilitada para mostrar o conteúdo de seus anexos.
Possibilidade de criação de apenas uma caixa-postal no servidor. Caixas-postais devem ser criadas nos computadores clientes.	Múltiplas caixas-postais podem ser criadas tanto no servidor quando no cliente.
Filtros podem atuar transferindo mensagens de entrada/saída apenas para caixas-postais locais dos computadores clientes.	Filtros podem atuar transferindo mensagens de entrada/saúda tanto para caixas-postais dos clientes como para caixas-postais do servidor.
E-mails de saída são armazenados apenas no computador cliente.	E-mails de saída podem ser filtrados para uma caixa-postal do servidor e acessados por outra máquina para visualização.

Quadro 2.1 – Quadro comparativo dos protocolos POP e IMAP.

Fonte: http://email.cityu.edu.hk/faq/popimap.htm.

Abaixo segue um desenho dos principais protocolos envolvidos durante a transmissão de um e-mail:

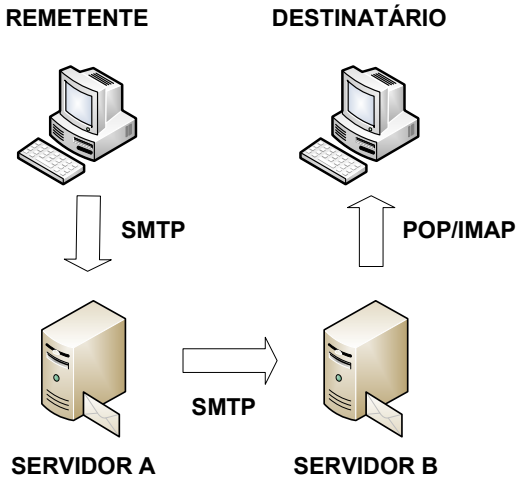

Figura 2.5 – Protocolos utilizados no envio/recebimento de e-mail.
Fonte: Elaborado por Autor [2008].

2.2.3.4 Tipos De Servidores Quanto À Localização

Quanto à localização existem servidores de *Webmail* e servidores de *Intramail*.

Os servidores de correio eletrônico que realizam suas tarefas através da *World Wide Web* (WWW) são chamados servidores de *Webmail*. Nestes, é utilizado o *browser* ou navegador web como programa cliente de e-mail. Em outras palavras, é utilizado o protocolo HTTP para os processos de leitura, composição e envio de correio, o que possibilita ao usuário acessar o seu e-mail em qualquer localidade, diferentemente de um *software* cliente específico que restringe o acesso ao e-mail somente pela máquina em que o mesmo está instalado e configurado.

Quando o gerenciamento do serviço de correio eletrônico é feita por uma máquina e este gerenciamento não depende da *Internet*, dizemos que esta máquina é um servidor de e-mail de *Intramail*, já que mensagens internas podem circular sem o uso da *Internet*.

CAPÍTULO 3. SEGURANÇA DE E-MAIL

A segurança de e-mail aqui tratada será específica às técnicas de combate ao forjamento de e-mails. Para isto, serão citadas as principais ameaças referentes à falsificação de remetentes, bem como as tecnologias de combate às mesmas.

3.1 ENGENHARIA SOCIAL

Analisando o conceito de segurança da informação, a engenharia social focaliza os métodos utilizados para se obter acesso a informações sigilosas de organizações ou pessoas por meio de artíficios de forjamento da identidade dos mesmos. Estes métodos exploram as falhas de segurança dos usuários alvos, os quais, sendo considerados leigos para esses tipos de ataques, ficam vulneráveis e podem ser facilmente manipulados. [WIKIPEDIA2008a]

Estelionatários utilizam técnicas de engenharia social para enviar mensagens de correio tentando induzir os usuários a registrarem informações pessoais nas respectivas mensagem eletrônicas. As informações capturadas são então usadas para falsificar a identidade de entidades confiáveis, com o intuito de se obter vantagem financeira com o devido roubo eletrônico. Estes e-mails, por sua vez, podem ser disparados em massa (*spams*) a inúmeros endereços da *Internet* ou então focalizarem alvos específicos. [WIKIPEDIA2008a]

A engenharia social pode visar proteger um sistema, bem como atacá-lo. Dentre os ataques de engenharia social (voltados para forjamento de mensagens eletrônicas) atuais destacam-se os *spams* e *trojans*. Serão conceituadas as técnicas de *phishing* e *ip-spoofing* em detrimento dos ataques de *spams* e *trojans*[10], que também serão explanados.

[10] *Trojan* é um programa de computador que aparentemente parecer ser útil e inofensivo, mas que contém códigos maliciosos escondidos para, quando executado, causar danos ao sistema.

3.2 AMEAÇAS EXTERNAS

Os serviços de correio eletrônico foram inicialmente programados para apresentar simplicidade e praticidade, já que antigamente o público alvo se restringia à comunidade acadêmica. Todavia, com o avanço de tecnologias e crescimento da *Internet*, estes serviços passaram a expor suas vulnerabilidades de segurança, pois as tecnologias apresentadas não previam o desenvolvimento acelerado da *Internet*.

Os códigos maliciosos comprometem o desempenho de um sistema, consomem recursos desnecessários e contribuem para diminuir os índices de segurança e usabilidade dos ambientes de rede atuais.

3.2.1 *Phishing*

Phishing é uma técnica usada para se roubar informações confidenciais de uma pessoa, como números de cartões de crédito, senhas ou qualquer informação registrada na *Internet* que seja considerada uma poderosa ferramenta em “mãos erradas”. Praticantes de *phishing* enviam milhares de *spams* utilizando a identidade de entidades populares consideradas confiáveis, tais como sites de bancos, empresas de cartões de crédito, sites governamentais, lojas comerciais, dentre outros.

Mensagens de *phishing* podem acionar um programa malicioso através de um simples clique no respectivo *link* ou encaminhar o usuário a um site falso, com as mesmas informações de um site legítimo, induzindo a vítima ao preenchimento de formulários forjados para a captura e manipulação de dados. Ver figura a seguir:

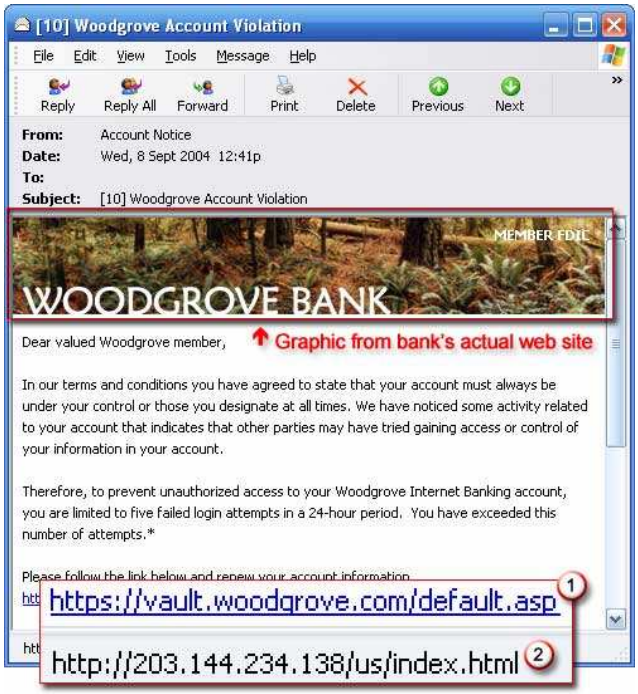

Figura 3.1 – Impressão de tela de e-mail contendo *phishing*.

Fonte: http://www.microsoft.com/protect/yourself/phishing/identify.mspx.

Neste exemplo o *spammer* copia o *layout* da página e o salva em um site temporário, com uma *url* semelhante ao site original (*https://vault.woodgrove.com/default.asp*). Note que o endereço real do atacante é *http://203.144.234.138/us/index.html* e que *https://vault.woodgrovebank.com/default.asp* refere-se ao endereço verdadeiro do site bancário. Deste modo, são feitas cópias de campos aonde o usuário necessita digitar suas informações pessoais, como o número da conta bancária e senha do cartão, sendo falsificado também o *layout* de e-mail original do banco. Assim, os milhares de *spams* são enviados com este *layout* falso à inúmeros endereços alvo da *Internet*, que apontam para o endereço do site falso. Os atacantes podem ainda estimular o usuário a clicar em algum *link* inserido no e-mail. No golpe da figura acima, o "banco" diz que para se prevenir contra acessos não autorizados, o correntista tem um limite de até 5 tentativas falhas de logon em um período de 24 horas e que este limite foi ultrapassado pelo mesmo, causando a desativação da conta, induzindo a vítima a clicar no *link* para atualizar suas informações.

Mensagens deste tipo geralmente são transmitidas via e-mail, mas podem também ser propagadas por programas de mensagens instantâneas, ou através de sites de relacionamento da *Internet*, dentre eles, o conhecido *Orkut*.

3.2.2 *IP Spoofing*

O termo *spoof* faz referência ao verbo mascarar. Esta técnica consiste no mascaramento de endereços IPs com endereços de remetentes falsos.

Devido as vulnerabilidades de segurança do protocolo IP, o pacote IP reencaminhado segue rumo ao seu destinatário sem a verificação do remetente da mensagem. Em outras palavras, durante a "estadia" do pacote no protocolo IP (camada de rede) não ocorre a checagem do IP do remetente, podendo este ser facilmente trocado por um IP válido com uma simples manipulação de cabeçalho. Logo, o IP que chega ao TCP da vítima é de um *host* confiável (usado para burlar a identidade remetente), possibilitando a infecção *IP Spoofing*. Veja a figura a seguir:

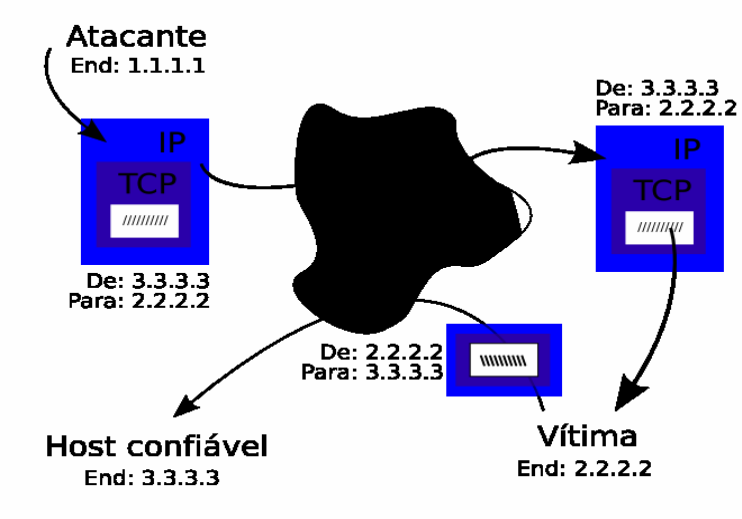

Figura 3.2 – Método de ataque *IP Spoofing*.
Fonte: http://pt.wikipedia.org/wiki/Ip_spoofing

Cada pacote enviado estará associada uma resposta (*reply*) do protocolo da camada superior. Isto porque para o pedido da conexão TCP do atacante com a vítima exige uma resposta, e esta resposta vai para quem enviou o pedido. Se o invasor forjou o endereço IP pelo IP do *host* confiável, isto explica o fato da resposta ser enviada para este *host* e não para o atacante a não ser que se utilize uma conexão UDP que envia comandos sem estabelecimento de conexão ou garantia de entrega.

Lembrando que esta técnica pode ser usada para o envio de *spams* e *trojans*.

3.2.3 Spams

O avanço da popularidade da *Internet* tem impulsionado comerciantes a utilizar o correio eletrônico como forma de propagada, enviando mensagens não solicitadas a diversos usuários em diferentes localidades. “Quando o conteúdo é exclusivamente comercial, esse tipo de mensagem é chamada de UCE (do inglês *Unsolicited Commercial E-mail*)” [ANTISPAM2008a]. Todavia, qualquer mensagem não solicitada pode ser considerada *spam.* Estes e-mails consomem recursos importantes e provocam uma redução de desempenho em sistemas de computação.

Os *spams* representam hoje o meio de ataque mais comum, e por serem disparados a inúmeros endereços alvo, possibilitam uma infecção generalizada, sendo considerados um dos tipos de ataque mais relevantes da engenharia social.

Dentre os problemas oriundos deste tipo de ameaça destacam-se:

- O não recebimento de e-mails válidos provocado pela saturação das caixas de correio devido ao acúmulo de *spams*;
- O tempo gasto para se receber estas mensagens (no caso de uma conexão discada este fator é altamente crítico);
- A perda de produtividade devido ao tempo perdido com a leitura destes e-mails;
- O consumo desnecessário de banda útil de rede;
- Desperdício de espaço em disco.

Os *spammers* tentam forjar endereços de sites confiáveis explorando as fragilidades do protocolo SMTP, conseguindo, por exemplo, alterar campos do cabeçalho de e-mail, induzindo o usuário a inserir seu dados no site falso com o intuito de se obter vantagem financeira através do respectivo golpe.

Lembrando que um *trojan* pode vir sobre forma de *spam*, porém nem todos os *spams* necessariamente contêm um *trojan* acoplado em uma mensagem de e-mail.

3.2.4 *Trojans*

O termo *trojan* ou cavalo de tróia teve sua origem da famosa conquista de Tróia pela Grécia, na chamada Guerra de Tróia, onde os gregos prepararam uma armadilha para os

troianos para a conquista da região. Refere-se a um programa ou código malicioso que assume uma identidade falsa e que abre portas seguras para invasão. Este programa atua como uma armadilha ou emboscada justamente por enganar o usuário e influenciá-lo a acreditar ser o mesmo uma aplicação segura.

Os *trojans* não criam cópias de si, distinguindo-se dos *vírus* e *worms* por tal característica, e não precisam de um programa hospedeiro para se disseminar, distinguindo-se somente dos *vírus* agora. Eles atuam como programa ou códigos independentes instalados diretamente no computador da vítima. Alguns até podem ser programados para se auto-destruirem através de um comando (induzido pelo atacante) do cliente ou depois de um certo tempo.

Os atacantes tomam o controle de outra máquina apenas com o envio de um arquivo, o qual, na maioria das vezes, é enviado por e-mail.

3.3 MECANISMOS DE DEFESA

As implementações de segurança de e-mail baseiam-se, principalmente, em métodos envolvendo a análise do corpo ou *payload*[11] da mensagem. Lembrando que em segurança da informação *payload* refere-se ao custo de infecção provocada por algum código malicioso. O protocolo SMTP não foi projetado para prover segurança e confiabilidade durante o transporte das mensagens entre os diversos agentes, e por tal motivo, não possui mecanismos robustos de autenticação. Algumas extensões do protocolo e opções de configuração ainda oferecem autenticação no sistema de transporte de mensagens, mas ainda assim geram falhas. [LYNEX2008]

Dentre as tecnologias de segurança para se evitar o forjamento de e-mails baseadas em autenticação de remetentes destacam-se o SPF e o DKIM. Outros métodos também prevêem um mecanismo de segurança, como *softwares anti-spams*, uso de listas-negras e mecanismos de reputação de remetentes. Além disto, outros tipos de listas também são usadas para se evitar o forjamento de e-mails, entretanto, as mesmas não serão detalhadas nesta monografia devido à extensão que isto proporcionaria ao projeto.

[11] *Payload* refere-se à carga útil ou conteúdo real de uma mensagem de e-mail, tirando informações de controle e roteamento.

A hierarquia dos métodos de filtragem de uma mensagem recebida pode ser manipulada, de acordo com a situação que melhor se adapte a um determinado ambiente de e-mail, analisando para isto os quesitos segurança e carga computacional processada. A figura seguinte mostra a estrutura de hierarquia de tratamento de mensagens proposta como modelo para elaboração deste projeto:

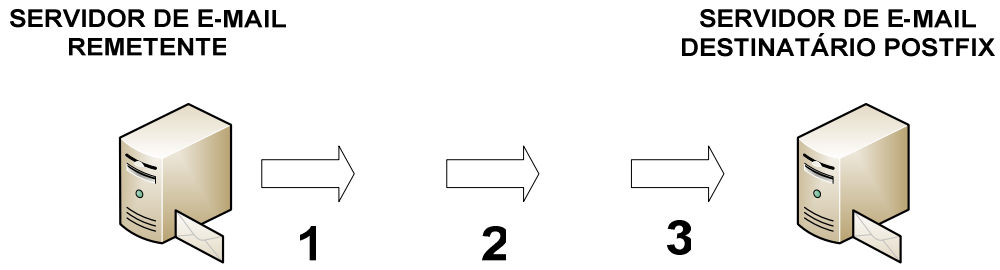

Figura 3.3 - Hierarquia proposta para tratamento de mensagens pelo MTA.
Fonte: Elaborado por Autor [2008].

1) Análise de filtragem de mensagens com a utilização de mecanismos de listas-negras. As tecnologias SPF e DKIM entram como primeiro mecanismo de filtragem em substituição a estas listas.

2) Análise da mensagem através de mecanismos de reputação de remetentes.

3) Análise detalhada da mensagem baseada em verificação heurística feita por *softwares anti-spams*.

3.3.1 Métodos *Anti-Spams*

Existem ferramentas *anti-spams* que classificam uma mensagem como *spam* ou não analisando apenas o cabeçalho e o corpo da mensagem, e dando uma pontuação a cada mensagem que entra com base no seu conteúdo. Outras se utilizam de métodos mais interativos, como desafios fáceis para o remetente. Estas, ao receber uma mensagem, enviam outra (*reply*) para o remetente, induzindo-o a responder uma pergunta ou tomar outra ação que seja fácil. Esta interação com o remetente permite que a mensagem seja liberada ou bloqueada. Caso o remetente responda corretamente (para isto ele deve ser o remetente real), a mensagem original é entregue ao destinatário. Caso contrário, é bloqueada. O quadro a seguir mostra alguns testes

feitos pelo *software SpamAssassin* para se pontuar uma mensagem de e-mail baseada em diversos parâmetros de verificação:

Parte do e-mail	**Descrição do teste**	**Nome do teste**	**Pontuação Adquirida** *(local, net, with bayes, with bayes+net)*
Cabeçalho	Tentativas de disfarçar a palavra "viagra".	VIA_GAP_GRA	2.203 1.053 2.004 0.133 modo 1 modo 2 modo 3 modo 4
Cabeçalho	Remetente do envelope encontrado em lista negra *dnsbl.ahbl.org*.	DNS_FROM_AHBL_RHSBL	0 (modo1) 2.025 (modo 2) 0 (modo 3) 0.692 (modo 4)
Cabeçalho	SORBS: remetente é um *open SMTP relay*.	RCVD_IN_SORBS_SMTP	1 (para qualquer modo)
Corpo	O corpo da mensagem apresenta 80 a 90% de linhas em branco.	BLANK_LINES_80_90	1 (para qualquer modo)
Corpo	Falta de uma linha em branco distinguindo o cabeçalho e o corpo do e-mail.	MISSING_MIME_HB_SEP	2,599 (modo1) 2.699 (modo 2) 2,205 (modo 3) 2,119 (modo 4)
Corpo	Tipos de *caracteres* que indicam linguagem estrangeira.	CHARSET_FARAWAY	3,200 (para qualquer modo)

Quadro 3.1 – Quadro de pontuação adquirida por mensagem classificada por *SpamAssassin*.
Fonte: http://spamassassin.apache.org/tests_3_2_x.html

Nota-se que em alguns casos são listados quatro pontuações para o *SpamAssassin.* Esta pontuação depende do modo como o *SpamAssassin* está sendo utilizado. No primeiro modo (*local*) são realizadas checagens sem o uso da *Internet*, e às análises que envolvem pesquisas em servidores web ou utilizam o *algoritmo bayesiano*[12] são atribuídas a pontuação 0.

[12] O *algoritmo bayesiano* refere-se ao algoritmo criado por Naive Bayes, aonde o filtro anti-spam é treinado para "aprender" o que é e o que não é spam.

No segundo modo (*net)*, são habilitadas as checagens de testes que precisam do uso da *Internet*, de modo que quando uma mensagem seja checada, são feitas perguntas à servidores da *Internet* para análise da respostas. Uma consulta a uma lista-negra é um exemplo desta checagem. Da mesma forma aos modos *local* e *with bayes* são atribuídos a pontuação 0 se somente *net* estiver habilitado. No terceiro modo é utilizado o algoritmo estático de Bayes. Da mesma forma, aos parâmetros de pesquisa que não usam este algoritmo são atribuídos a pontuação 0. (*local* e *net*). No quarto modo, é feita a combinação do modo *net* com o modo *with bayes*, que é o padrão utilizado pelo *SpamAssassin.*

Seja em qual modo ele estiver atuando, se a pontuação adquirida pela mensagem for acima de 5 pontos (padrão), mas não muito acima, ela já é considerada *spam*, mas é colocada num arquivo chamado “bulk” dentro da pasta “mail” do usuário. Se a pontuação ultrapassar os 5 pontos de maneira exagerada (14 pontos, por exemplo), ela é colocada no arquivo *SPAN* localizado na pasta do perfil do usuário. Se a pontuação for inferior a 5 pontos, ela é colocada na pasta usual referente à caixa de correio do usuário, sendo considerada uma mensagem real.

Todavia, seja qual for o critério de bloqueio estabelecido por estes *softwares*, existe um tempo de processamento para a análise de pontuação da mensagem e a sua respectiva classificação como *spam* ou não pode acarretar uma sobrecarga de processamento no servidor de e-mail, dependendo do volume de mensagens a ser processado e dos parâmetros configurados para análise das mesmas.

Por isto, bloquear uma mensagem considerada *spam* de maneira correta por um critério que anteceda esta análise seria o ideal, já que evitaria processamento desnecessário decorrente da árdua tarefa desta filtragem.

3.3.2 Uso de Listas-Negras

As listas negras consistem em registros de endereços Ips, e-mails ou domínios que identificam fontes de *spam*. Normalmente a consulta a este tipo de lista é feita através do serviço de DNS pelo MTA. (a maioria dos MTAs oferecem suporte a estas consultas), que pesquisa as respectivas bases de dados. Caso a fonte seja suspeita de *spam*, o DNS não resolve o IP especificado. Esse tipo de lista-negra é conhecido por DNS RBL (Real-Time Black-List in Domain Name Service), ou seja, listas negras consultadas através do DNS em tempo real.

O funcionamento de uma lista-negra se dá da seguinte forma: Caso o IP ***10.20.30.40*** seja uma fonte catalogada em uma dessas listas, ele aparecerá como ***40.30.20.10.nome.da.lista*** na respectiva lista. Caso se consulte esta lista referente a este IP e se obtenha uma resposta, a qual indica o motivo da inclusão deste endereço IP na lista, significa que o IP faz parte da lista e deve ser bloqueado.

Entretanto, este método de segurança de e-mail possui vulnerabilidades. “Algumas listas possuem critérios de inclusão de IPs pouco seletivos” [ANTISPAM2008b] e, por tal motivo, chegam a incluir IPs seguros, ou que não representam ameaças.

Um exemplo de falso-positivo provocado pelo uso destas listas é o fato de muitos administradores bloquearem, ao invés de IPs de servidores específicos, a faixa de endereçamento IP inteira nas quais estes servidores estavam, e como conseqüência, alguns servidores reais acabam sendo catalogados nas mesmas.

Nota-se claramente que o conhecimento da identidade do remetente passa então a ser fundamental para se amenizar estas falhas. Desta forma, foram desenvolvidas técnicas que permitissem conhecer a verdadeira origem da mensagem e que otimizassem o processamento nos servidores. Os serviços de reputação bem como as ferramentas SPF e DKIM são alguns exemplos destes métodos.

3.3.3 Mecanismos de Reputação

Os serviços de reputação representam listas de IPs que foram bloqueados em função de terem sido considerados como endereços suspeitos ou com uma má reputação. Estes serviços trabalham monitorando o tráfego de pacotes enviados aos servidores de e-mail para se coletar informações a respeito do endereço IP de origem da mensagem. São analisados vários parâmetros de comparação, e a cada parâmetro deste é dada uma nota. A pontuação relativa à soma das notas dos parâmetros qualifica ou não o e-mail enviado, ou seja, estabelece uma boa ou má reputação ao respectivo IP de origem. Geralmente os recursos de reputação vêem vinculados a um *software* de combate aos *spams*, como uma funcionalidade *anti-spam* habilitada em servidores de e-mail para bloquear mensagens de acordo com as diversas características do remetente.

A diferença básica de um monitoramento através da análise da reputação em comparação a uma lista-negra está na verificação contínua do comportamento dos IPs que enviam mensagens e os respectivos domínios.

A análise da reputação contribui para diminuir a incidência dos falso-positivos, reduzir a largura de banda do canal SMTP, otimizar o desempenho de processamento e reduzir o armazenamento de mensagens, já que diminui o ocorrência das mesmas.

3.3.4 Autenticação de Remetentes

Um dos primordiais passos para se contra-atacar os problemas relativos à segurança de e-mail atual é exatamente conhecer a identidade do provável atacante, visto que *spams*, *trojans* ou qualquer outro código malicioso são geralmente propagados por usuários de domínios forjados.

Diversas tecnologias de identificação vêm sendo elaboradas em resposta à crescente propagação do *spam*. Mensagens de *spam* normalmente são enviadas pela *Internet* sem nenhuma autenticação. Na verdade, é fácil enviar um e-mail em nome de outra pessoa, pois não existem métodos automatizados para detecção de mensagens forjadas. As tecnologias de autenticação de remetentes tratam especificadamente desta brecha.

O SPF é caracterizado por ser uma extensão do protocolo SMTP e atua diretamente no servidor DNS. O servidor de e-mail destinatário utiliza recursos do DNS remetente para conseguir informações sobre o servidor de origem da mensagem. Desta forma, é criada uma estrutura de verificação de autenticidade entre servidores de e-mail.

A tecnologia DKIM consiste em assinar as mensagens que saem com uma chave privada, e publicar uma chave pública no servidor DNS, para que o servidor destinatário possa comparar as duas chaves e autenticar o emissor com base nesta análise.

Vale ressaltar que nem o SPF nem o DKIM analisam o corpo da mensagem.

3.3.4.1 *Sender Police Framework*

As políticas e os filtros SPF foram estruturados a partir de uma linguagem simples, a qual aponta o endereço real de um determinado servidor de e-mail. Desta maneira, servidores

são capazes de comprovar a identidade de determinado domínio, e de descartar, analisar ou então manipular o e-mail forjado de acordo com as políticas estabelecidas neste servidor.

3.3.4.1.1 Funcionamento SPF

Resumidamente, cada domínio interessado em utilizar esta ferramenta acrescenta uma linha de texto padronizada à configuração do seu próprio servidor DNS, descrevendo quais os endereços dos servidores de e-mail autorizados a enviar mensagens daquele domínio.

Servidores que utilizam a autenticação de remetentes baseada nos mecanismos do SPF deverão possuir atributos ou características únicas (registros SPF) que serão utilizadas por outros servidores para viabilizar o processo de autenticação. Estes registros são armazenados no servidor DNS em um campo TXT para facilitar a consulta.

Para que o SPF funcione, são necessárias duas ações:

- A correta publicação do registro SPF no servidor DNS responsável pelo domínio remetente;

- A ativação do serviço de verificação SPF no servidor MTA destinatário, de forma que este servidor descarte mensagens oriundas de servidores que não possam ser autorizados de acordo com a política SPF;

A figura a seguir mostra o funcionamento do SPF:

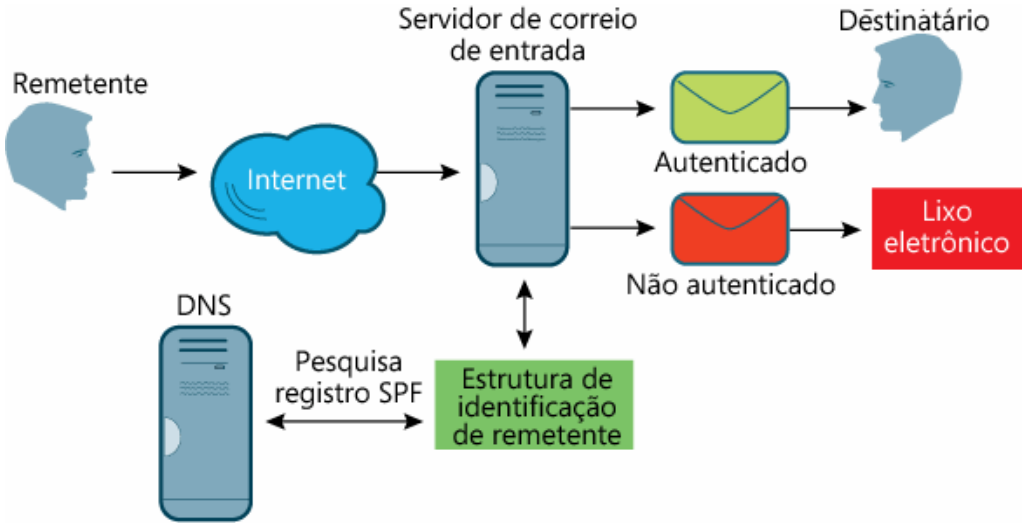

Figura 3.4 – Funcionamento SPF.

Fonte: http://technet.microsoft.com/pt-br/magazine/cc160870.aspx.

Baseado na figura 3.4, o *servidor de correio de entrada*, ao receber uma mensagem, fará uma requisição ao seu serviço de verificação SPF pedindo a validação do domínio do servidor de e-mail emissor. Para isto irá pesquisar o registro SPF presente no servidor DNS deste remetente, que retornará uma das seguintes respostas descritas abaixo:

Resultado	**Motivo**	**Ação**
None	Não existem registros SPF publicados em nome do domínio emissor.	Aceitar
Neutral	O registro não especifica nada sobre validação do nome ou IP. Comportamento indefinido.	Aceitar
Pass	O registro especifica nome ou IP como “autorizado”.	Aceitar
Fail	O registro especifica nome ou IP como “não autorizado”.	Rejeitar
Softfail	O registro especifica que o domínio remetente não está autorizado, mas o estado da mensagem não é definido com segurança. Assim, a mensagem é aceita e marcada como suspeita.	Aceitar e marcar
TempError	O servidor DNS do domínio remetente encontra-se em situação de erro durante o processo de checagem.	Aceitar ou rejeitar
PermError	O registro apresenta um erro permanente. O domínio não entende ou não interpreta a identidade “Mail From” de maneira correta, podendo rejeitar a mensagem. Ex: má formatação do registro SPF.	Não definido

Quadro 3.2 – Quadro de resposta DNS ao servidor destinatário de correio.

Fonte: Elaborado por Autor [2008].

Vale ressaltar que as informações SPF são retiradas dos campos “HELO” e “MAIL FROM” presentes no envelope da mensagem (figura 3.5), oriundos da RFC 2821. O campo “HELO” identifica o cliente SMTP (MTA remetente) ao servidor SMTP (MTA destinatário) e tem a sua definição por “HELO” ou “EHLO” e o “MAIL FROM” serve para iniciar a transação da mensagem e deve conter o *reverse-path* (endereço de e-mail do remetente). Este endereço serve para saber para onde retornar mensagens de erro. Veja a figura abaixo:

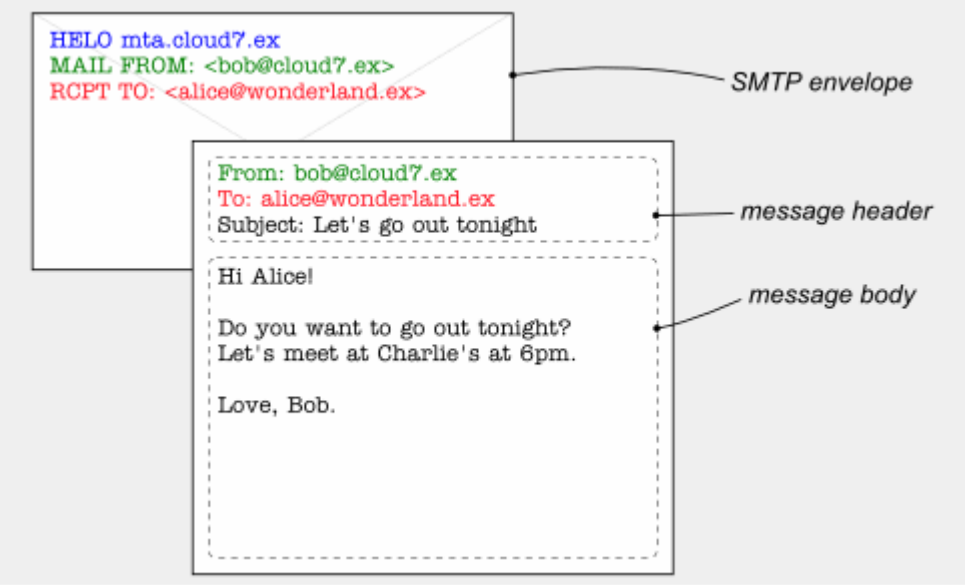

Figura 3.5 – Campos do Envelope SPF.

Fonte: http://www.openspf.org/Related_Solutions.

O SPF autentica o “HELO” e o “MAIL FROM” comparando o endereço IP do “HELO” (*mta.cloud7.ex*) com a lista de endereços IP do/s servidor/es de e-mail autorizado/s a enviar e-mail pelo referido domínio.

O protocolo SPF atua no envelope da mensagem, durante a transação SMTP. Caso ele consiga identificar o forjamento no envelope, é impedida a transmissão do cabeçalho e consecutivamente do corpo do pacote.

3.3.4.1.2 Registros SPF

Os registros SPF são armazenados nos servidores DNS das máquinas que se adequaram à política SPF e servem para armazenar informações sobre domínios de e-mail na

Internet, a fim de facilitar a autenticação dos mesmos. Segue formato de registro SPF e exemplo logo em seguida:

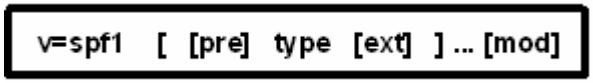

Figura 3.6 - Formato de registro SPF.
Fonte: Elaborado por Autor [2008].

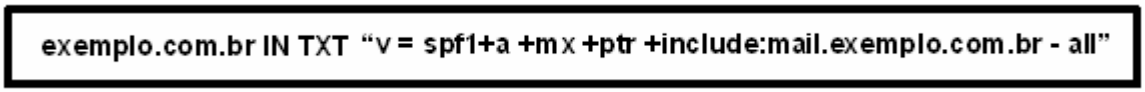

Figura 3.7 - Exemplo de registro SPF na forma de registro TXT do DNS.
Fonte: Elaborado por Autor [2008].

O quadro a seguir representa cada um destes parâmetros e a sua respectiva descrição:

Parâmetro	Descrição
v=spf1	Versão do protocolo: Um cliente SPF ou máquina que utilize este recurso deverá utilizar apenas registros da mesma versão ou então versão compatível ao seu processador SPF.
pre	Prefixos: Os prefixos relatam o comportamento dos mecanismos de autenticação caso se encontre um registro SPF. Eles antecedem os mecanismos para definir o seu comportamento. As condições definidas por eles são: + (*pass*), - (*fail*), ~ (*softfail*) e ? (*neutral*). A utilização do SPF se faz negando ou permitindo endereços combinando as condições com os parâmetros. **Pass:** Permite o parâmetro – Este é o prefixo padrão caso nenhum seja definido. **Fail:** Nega o parâmetro. **Softfail:** Estado intermediário entre *fail* e *neutral.* **Neutral:** Neutraliza o parâmetro. Interpretado como *none*.

type	Tipos: Define quais os tipos de mecanismos SPF usados. São eles: • **all:** Aplica a regra para todos os endereços. Deve ser colocado o parâmetro mais à direita do registro (último parâmetro). **Exemplo:** **v=spf1 a –all** Este exemplo significa que serão liberados o/s IP/s presentes no registro A do DNS e qualquer outro será bloqueado (- *all*). • **include:** Serve para incluir mais de um domínio com apenas um registro SPF. Possibilita que um domínio possa designar outros domínios ou subdomínios autorizados na política SPF. No exemplo abaixo os servidores *example.com* e *example.org* estão autorizados a enviar mensagens por determinado domínio. **Exemplo:** **IN TXT "v=spf1 include:example.com include: example.org -all** • **a:** Aplica a regra para um *host* específico. Usa o registro A para verificação, aplicando a regra a ele caso este registro seja um IP válido. **Exemplo:** **v=spf1 a –all** **v=spf1 a:example.org –all** • **mx:** Aplica a regra se a máquina de origem for um servidor de e-mail. Usa o registro MX para verificação. O registro MX define no DNS quem é o MTA que recebe correio. Caso não seja o mesmo MTA que envia, os testes MX falharão. **Exemplo:** **v=spf1 mx -all** **v=spf1 mx mx:example.org -all** Neste segundo exemplo são permitidos os IPs cadastrados no/s registro/s MX do DNS e é verificado se *example.org* refere-se a um registro MX. Se for o caso, o servidor é permitido.

- **ptr:** Aplica a regra de acordo com o registro PTR do DNS. Verifica se o nome da máquina de origem encontra-se em um determinado domínio. Para isto, é feita uma consulta reversa para checar se o endereço IP confere com o nome especificado.

Exemplo:

v=spf1 ptr –all

v=spf1 ptr:example.com –all

No primeiro exemplo, o/s endereço/s IP/s associados à consulta PTR de DNS serão permitidos para o envio de e-mails. No segundo exemplo, só será permitido o IP que se refira à consulta PTR referente ao domínio *example.com*, caso este domínio exista.

- **ip4:** Testa se a máquina de origem usa uma rede baseada no protocolo IPv4 e aplica a regra.

Exemplo:

v=spf1 ip4:192.0.2.128/28 –all

Neste exemplo, é verificado a existência de registros do tipo A que pertencem à faixa de endereçamento estipulada acima. Sendo assim, um *host* que tiver o IP 192.0.1.65 não será permitido e um que tiver o IP 192.0.2.129 será permitido.

- **Ip6:** Testa se a máquina de origem usa uma rede baseada no protocolo IPv6 e aplica a regra.

Exemplo:

v=spf1 ip6 –all

- **exists:** Testa a existência de um domínio.

Exemplo:

v=spf1 exists:example.com –all

ext	Parâmetro que define uma extensão especial ao tipo. Se for omitido, significa que só é usado um tipo de registro para as questões.
mod	Modificadores: Os modificadores servem para prover informações adicionais ou alterações no processamento dos mecanismos do SPF. Os modificadores são: *redirect e exp:* • **redirect**: Caso todos os mecanismos falhem e estando definido um modificador *redirect*, então o mesmo irá alterar o domínio para ser processado. É usado na forma *redirect=domain* **Exemplo:** **v=spf1 mx -all redirect=example.org** • **exp:** Gera uma mensagem SMTP para a máquina emissora caso o processamento do SPF resultar em rejeição, ou em outras palavras, personaliza uma mensagem de erro. **Exemplo:** **v=spf1 mx -all exp= "Erro por tal motivo"**

Quadro 3.3 – Descrição dos parâmetros de registro SPF publicado em DNS.
Fonte: Elaborado por Autor [2008].

3.3.4.2 *DomainKeys Identified Mail*

O DKIM, é a união do *DomainKeys*, da *Yahoo*, e do *Identified Internet Mail*, da *Cisco System Inc*, como sendo o resultado de um longo trabalho de colaboração entre diferentes empresas para o desenvolvimento de um protocolo aberto de autenticação de correio eletrônico definido pela RFC 4871.

3.3.4.2.1 Funcionamento DKIM

O DKIM age assinando as mensagens que saem do MTA via chave privada e verificando as mensagens que chegam via chave pública através de uma consulta ao servidor DNS. Portanto, para ativação do DKIM, é necessário:

- Criar um par de chaves pública/privada;

- Deixar a chave pública disponível via DNS, de forma similar a publicação de registro SPF;

- Deixar a chave privada no MTA responsável pelo envio da mensagem.

Quando a mensagem chega ao servidor destinatário, este deve submetê-la aos seguintes processos de autenticação:

- Extração da chave privada do emissor de e-mail;

- Solicitação da chave pública do domínio em questão;

- Utilização da chave pública para verificar se a assinatura foi gerada a partir da chave privada do servidor remetente.

Para se entender o funcionamento DKIM, é preciso primeiro entender o conceito de criptografia. A criptografia nada mais é do que uma maneira de disfarçar informações. Para isto, é usado um algoritmo de criptografia (uma fórmula matemática) que codificará a informação. Para que o algoritmo funcione, é preciso inserir uma chave ou valor que o mesmo irá utilizar para codificar ou decodificar a informação.

O DKIM usa criptografia de chave pública ou assimétrica, aonde são disponibilizadas duas chaves, uma pública e outra privada. O que uma chave encripta, somente a outra respectiva a esta decripta. Assim, uma mensagem cifrada com a chave pública, pode somente ser decifrada com sua chave privada correspondente (figura 3.8). Do mesmo modo, uma mensagem cifrada com sua chave privada, pode somente ser decifrada pela sua chave pública correspondente. (figura 3.9). A chave privada somente é conhecida pelo seu dono e a chave pública é de acesso a todos. O primeiro método garante confidencialidade e o segundo autenticidade. Ver figuras seguintes:

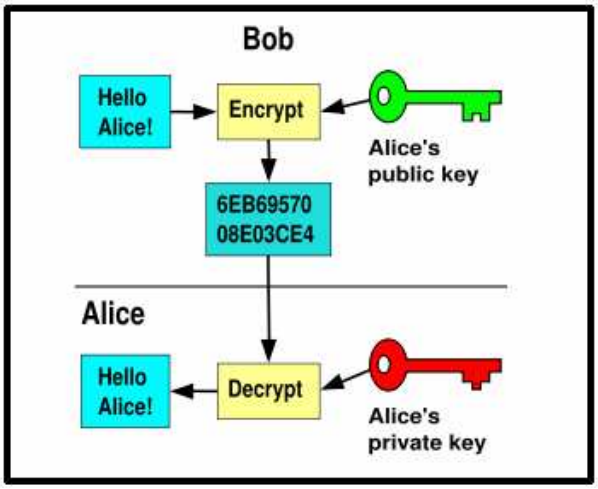

Figura 3.8 – Mensagem cifrada com chave pública e decifrada com chave privada.

Fonte: http://en.wikipedia.org/wiki/Public-key_cryptography.

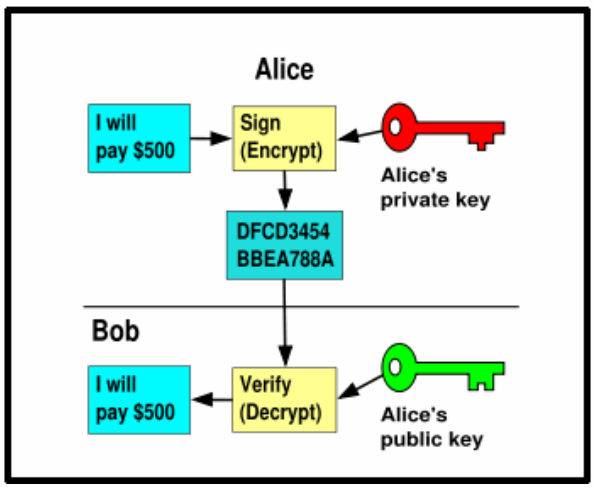

Figura 3.9 – Mensagem cifrada com chave privada e decifrada com chave pública.

Fonte: http://en.wikipedia.org/wiki/Public-key_cryptography.

O DKIM utiliza o segundo método, assinando a mensagem com uma chave privada e disponibilizando uma chave pública de acesso a todos.

A figura a seguir representa o funcionamento DKIM:

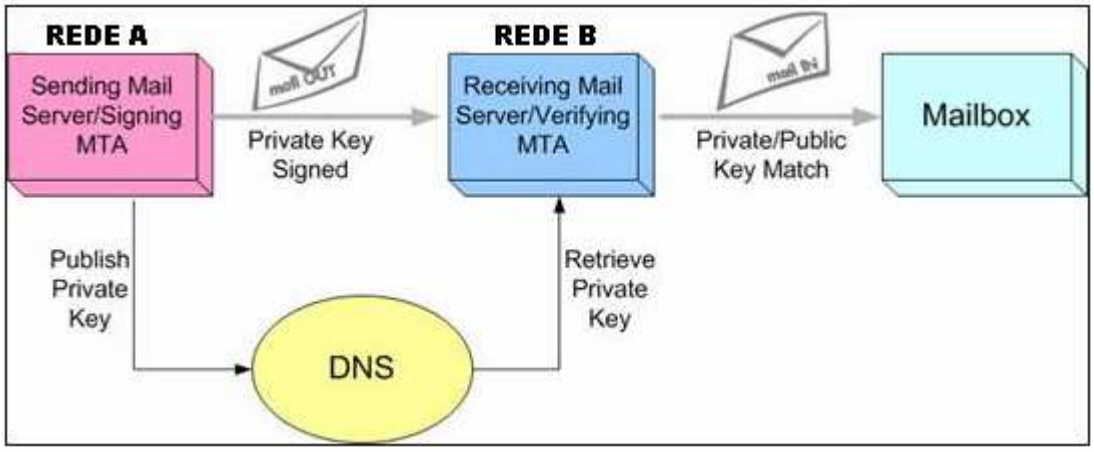

Figura 3.10 – Funcionamento DKIM.

Fonte: http://www.axigen.com/articles/introduction-to-antispam-practices_19.html.

O assinante DKIM (*REDE A*) gera um *valor hash* para o corpo da mensagem enviada através do algoritmo SHA1 ou SHA256 e criptografa este valor usando sua chave privada RSA. O resultado desta encriptação (assinatura digital) é armazenado em um campo cabeçalho de e-mail e enviado para o destinatário, juntamente com outras informações de controle. O *valor hash* refere-se a uma versão codificada resumida da mensagem e serve para não permitir a modificação da mesma durante o transporte, já que, caso aconteça, o *valor hash* também será modificado. A assinatura digital é o resultado do *valor hash* criptografado com a chave privada. A assinatura da mensagem enviada pode ser realizada por qualquer um dos *softwares* agentes: MUA, MSA ou MTA ou pode ser feita por *softwares* de terceiros autorizados. Após a assinatura, qualquer agente no caminho de tráfego da mensagem pode validá-la, mas geralmente é feita pelo MTA destinatário (*REDE B*). Paralelo aos passos descritos anteriormente, o assinante DKIM também disponibiliza uma chave pública em seu servidor DNS.

No lado receptor do processo de comunicação, o servidor destinatário realiza uma consulta ao servidor DNS do remetente ou assinante DKIM, e busca informações sobre o servidor de origem, dentre estas, a chave pública correspondente. O destinatário então recria o *valor hash* para a mensagem recebida, e utiliza a chave pública para decifrar a mensagem que foi assinada pelo emissor presente no cabeçalho de e-mail. Se este *valor hash*, obtido com a decifração desta assinatura, corresponder ao *valor hash* recriado da mensagem, o remetente comprovou sua identidade e o *valor hash* poderá então ser decifrado para o formato original da mensagem de e-mail.

O mecanismo de *hash* por si só, não garante autenticação. Deve ser usado em conjunto com algum outro método de cifragem. No caso do DKIM é utilizado em conjunto com o algoritmo RSA. RSA refere-se a um tipo de algoritmo baseado na criptografia assimétrica. Este nome deve-se às iniciais dos sobrenomes dos fundadores desta tecnologia: Ron **R**ivest, Adi **S**hamir e Len **A**dleman. SHA1 ou SHA256 refere-se ao algoritmo utilizado para geração do código *hash*. Portanto, o algoritmo de criptografia completo é chamado de RSA: SHA1 ou RSA: SHA256. [WIKIPEDIA2008b]

A figura abaixo mostra um exemplo de *valor hash* calculado para o corpo da mensagem de e-mail:

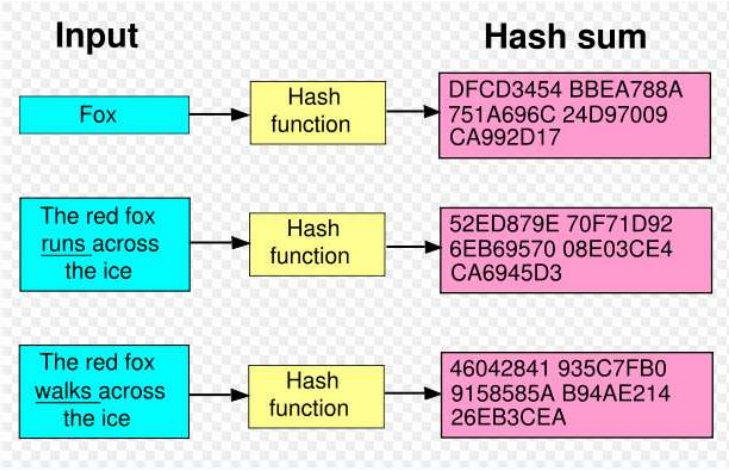

Figura 3.11 – Exemplo de código *hash*.

Fonte: http://en.wikipedia.org/wiki/Cryptographic_hash.

Este código possui um tamanho de 160 bits, quando se é utilizado o algoritmo SHA1 (*Secure Hash Algorithm 1*) e 256 bits quando se é utilizado o algoritmo SHA256 (*Secure Hash Algorithm 256*). O *valor hash* é também chamado de *checksum*, *digest*, *message digest* ou resumo criptográfico.

3.3.4.2.2 Registros DKIM

O processo de autenticação DKIM é diferente do SPF. Como já dito, há um registro DKIM presente no cabeçalho de e-mail e um registro DKIM presente no DNS deste domínio de origem.

Em relação ao registro presente no DNS, este se refere à chave pública do domínio em questão, sendo composto pelo nome de domínio de origem, a seqüência de caracteres *_domainkey* e um seletor do cabeçalho da mensagem, além de outras informações de controle.

Um seletor nada mais é do que uma *string* criada no *namespace* do DNS para permitir a publicação de múltiplas chaves para um mesmo domínio, para que este possa ter controles de assinaturas diferentes distinguindo-as por departamentos ou setores da empresa, por datas ou por terceiros agindo em nome de um determinado domínio. *Namespace* é o espaço de nomes reservado para preenchimento da *string* referente ao domínio consultado dentro da estrutura do DNS. Considerando estas informações, no exemplo abaixo, *may2005* refere-se ao seletor, a string *_domainkey* refere-se ao *namespace* especial proposto pela especificação DKIM para publicação de chaves públicas de DNS e *dominio1.com* refere-se ao domínio em questão.

(selector._domainkey.example.com)

may2005_domainkey.dominio1.com

O DKIM segue a estrutura "parâmetro=valor" para realizar as assinaturas. Segue abaixo exemplo de registro DKIM publicado no DNS.

may2005_domainkey IN TXT "v=DKIM1\g= *; k=rsa\; t=y\;
p=MIdLXdJOc9G2q8LE […]HRNiYzR"

O quadro seguinte representa o parâmetro, o valor correspondente e o significado do mesmo:

Parâmetro	Valor	Significado
v	"DKIM1" ou versão superior	Parâmetro recomendado: Versão do protocolo DKIM.
k	O padrão é "rsa"	Parâmetro opcional: Algoritmo baseado em criptografia assimétrica utilizado para realizar verificação DKIM.
h	"sha1/sha256"	Parâmetro opcional: Lista todos os tipos de algoritmos para geração do código *hash*.

g	O padrão é “*”	Parâmetro opcional: Granularidade da chave. Determina quais e-mails serão validados usando um mesmo seletor (e por conseqüência uma mesma chave). Isto permite que uma mesma chave seja usada para assinar mais de um e-mail, que só será aceito se a granularidade 'casar' com o parâmetro *i* do registro DKIM enviado no cabeçalho e se a chave pública validar a assinatura.
n	Este valor é definido pelo dono do domínio remetente. Exemplo: “Este domínio possui registro DKIM publicado”.	Parâmetro opcional: Este parâmetro serve para publicar alguma informação necessária ou de interesse do domínio referenciado. Nenhuma interpretação é feita pelo *software*.
p	“MIdLXdJOc9G2q8LoXSlEniSbav +y[…]HRNiYzR”	Parâmetro requerido: Dados da chave pública. Um valor nulo significa que a chave foi revogada.
s	Pode ser “e-mail” ou “*”	Parâmetro opcional: Tipo de serviço a que o seletor pode se referenciar. Como publicado na RFC 4871 [ALLMAN2007, p.51], o tipo de serviço especificado é o serviço de correio eletrônico. O valor “*” indica todos os tipos de serviços suportados.
t	Pode ser “y” ou “s”	Parâmetro opcional: Modo de teste. Indica se a chave criada é apenas para teste. “y” Indica que o domínio está apenas testando a ferramenta e “s” indica que o domínio está realmente usando a ferramenta.

Quadro 3.4 – Descrição dos parâmetros do registro DKIM publicado em DNS.

Fonte: Elaborado por Autor [2008].

Em relação ao registro DKIM enviado pelo assinante no cabeçalho de e-mail, os dados assinados pela chave-privada, além de outras informações de controle, são armazenados no campo *DKIM-Signature*. A figura a seguir representa um *print screen* do cabeçalho de e-mail assinado com o DKIM mostrando este campo:

DKIM-Signature:	**v**=1; **a**=rsa-sha256; **c**=relaxed/relaxed; **d**=gmail.com; **s**=gamma; **h**=domainkey-signature:received:received:message-id:date:from:subject:mime-version:content-type; **bh**=OpXbiNWEPlDqy7ToK6HJTW2gNe5UV5kYtE3d87paJS8=; **b**=rFn6rTs3jB94eSTDQTMSKKWMo1RNyGVqYImxwwoKu0d119wEovwGuuvCj2LE+lFvgMUpWYezSOyIrkDTMjRzdHa1ZUibiJXd3umkxni2sMPqPxBohl6VbkyArDYNaUq5SOciNdLcpqx8vqDzoBszkBP6fZ+1OCI+OmPdK2ulciI=

Figura 3.12 – Campo de cabeçalho DKIM-Signature.
Fonte: Elaborado por Autor [2008].

Assim como os parâmetros pertinentes ao registro DKIM armazenado em DNS foram explanados na tabela 3.5, segue a descrição de cada parâmetro, valor e significado correspondente ao campo de cabeçalho *DKIM-Signature*:

Parâmetro	Valor	Significado
v	"1"	Parâmetro que deve estar incluso: Versão do protocolo DKIM.
a	"rsa-sha256"	Parâmetro requerido: Algoritmo criptográfico completo utilizado para gerar a assinatura de envio.
b	"VSdhNaUqewszeslmOyRoaP550xKN+Prij3Skqqw2CJZ3opivo38xVALSaeaWJygc5h7SEcA8gBIWBSzepDdAwomLgoJmdNKlOZ0E3OgHeYBcbeM5KM7srjyGc="	Parâmetro requerido: Assinatura da mensagem feita com a utilização da chave privada (confidencial) e do *valor hash.*

bh	“OpXbiNWEPlDqy7ToK6HJTW2gNe5UV5kYtE3d87paJS8=”	Parâmetro requerido: *Valor hash* do corpo da mensagem.
c	“simple/simple”	Parâmetro opcional: Tipo de algoritmo para se “hashear” a mensagem. O primeiro “simple” indica o *hash* aplicado ao cabeçalho e o segundo “simple” o *hash* aplicado ao corpo da mensagem. Se somente um valor é definido (“c = simple” ou “relaxed”), o segundo valor sempre é tratado como “simple”. Exemplo: “c = relaxed” é o mesmo que “c = relaxed/simple”.
d	“<dominio.com>”	Parâmetro requerido: Indica o nome do domínio. Exemplo: “d = gmail.com”
h	Exemplo: “domainkey-signature:received: received:message-id:date:from:subject: mime- version:content-type”	Parâmetro requerido: Indica os campos do cabeçalho usados pelo algoritmo DKIM (RSA: SHA1 ou RSA: SHA256)
i	“<usuário>@<dominio.com>”	Parâmetro opcional: Especifica o nome do usuário emissor do e-mail, seguido do nome do domínio correspondente. O usuário é separado do domínio por um sinal de @. Exemplo: “i = rodrigo@valido.com”.
l	“76”	Parâmetro opcional: Indica o tamanho do corpo da mensagem antes de se aplicar um *hash* ao mesmo. O valor à esquerda indica que o corpo da mensagem possui 76 *caracteres* decimais.
q	O padrão é “dns/txt”	Parâmetro opcional: Indica o método de publicação da chave pública.

s	“<nome_do_seletor>”	Parâmetro requerido: Indica o nome do seletor presente em registro DKIM publicado no DNS.
t	“1117578”	Parâmetro recomendado: Indica o tempo decorrido (em segundos), desde que a assinatura DKIM foi criada. O valor à esquerda indica que se já se passaram 1.117.578 segundos da criação da assinatura DKIM.
x	“1156911”	Parâmetro recomendado: Indica o tempo de expiração da assinatura (em segundos).
z	“From:rodrigo@valido.com”	“Parâmetro opcional: Indica uma cópia dos campos de cabeçalho contidos no parâmetro z.

Quadro 3.5 – Descrição dos parâmetros de registro DKIM presente em cabeçalho de e-mail.
Fonte: Elaborado por Autor [2008].

O resultado da verificação de assinatura DKIM pode resultar em três conclusões:

1 – A assinatura é valida (*pass*). Isto significa que a mensagem vem realmente do campo de cabeçalho “From”. A mensagem é liberada, caso não seja avaliada por outras técnicas *anti-spam.*

2 – A assinatura não é válida (*fail*). A mensagem é marcada como suspeita ou é recusada.

3 – O domínio do emissor não possui registro DKIM publicado (*none*). A mensagem é liberada, pois não há nenhuma informação DKIM como critério de filtragem.

CAPÍTULO 4. INFRA-ESTRUTURA DO PROJETO

4.1 INTRODUÇÃO

Este capítulo refere-se às especificações técnicas para o desenvolvimento da parte prática do projeto. Inicialmente será apresentada a topologia de rede para os cenários propostos e em seguida detalhada a arquitetura disponibilizada. A arquitetura refere-se ao *hardware* e *software* utilizado.

Os processos de instalação e configuração das ferramentas SPF e DKIM, inclusos a manipulação de *scripts* que facilitarão a checagem destes filtros, edição e configuração de arquivos necessários, bem como alguns erros e dificuldades encontradas serão mostrados neste capítulo.

4.2 TOPOLOGIA

Na prática, o SPF e o DKIM atuam na *Internet*. Cada domínio interessado em publicar um registro SPF ou DKIM o faz no arquivo de zona do DNS responsável pelo seu domínio, geralmente uma máquina diferente da servidora de e-mail, podendo situar-se localmente ou remotamente a esta rede. Portanto, para a correta verificação de checagem de remetente, toda a estrutura de roteamento da *Internet* é levada em consideração.

Para a proposta de implementação desta monografia, e coleta de resultados, estas ferramentas foram implementadas de maneira local e com a utilização do *software* de virtualização *VMWare Server 1.0.5.* A escolha deste deve-se ao fato de ser uma ferramenta gratuita, diferente de outras ferramentas de virtualização desenvolvidas pela empresa norte americana *VMWare* (*http://www.wmware.com*).

4.2.1 Topologia de Rede Virtual

A figura 4.1 mostra a virtualização da estrutura de rede física proposta para a implementação dos dois cenários:

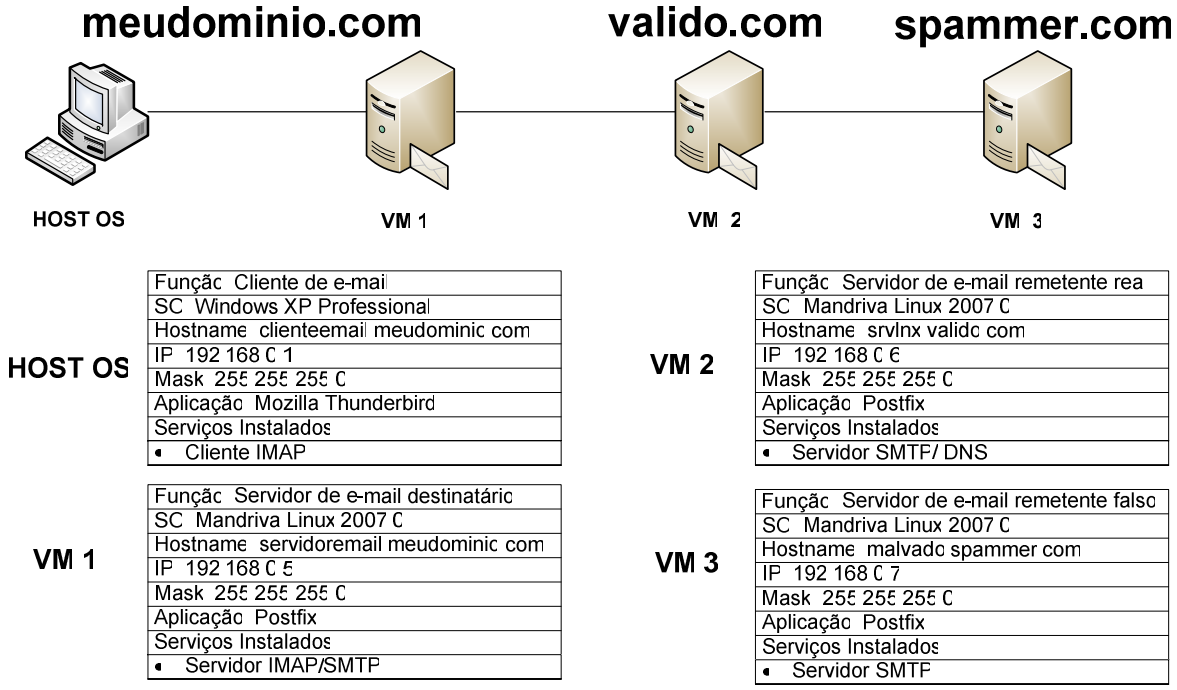

Figura 4.1 – Estrutura de rede virtual para criação dos cenários da implementação.

Fonte: Elaborado por Autor [2008].

Como retratado na figura acima, as máquinas *VM 1*, *VM 2* e *VM 3* são servidores de e-mail e a máquina *HOST OS* representa o cliente de e-mail. A máquina *VM 1* (*servidoremail.meudominio.com*) representa o servidor destinatário implementado com as verificações SPF/DKIM. A máquina *VM 2* (*srvlnx.valido.com*) refere-se ao servidor remetente válido e a máquina *VM 3* representa o servidor de e-mail remetente falso (*malvado.spammer.com*), que tentará se passar pela máquina *VM 2* para as situações simuladas.

No cenário da *Internet*, cada domínio listado acima necessitaria do seu próprio servidor DNS para resolver os pedidos de resolução de nomes pertinentes ao destino de uma mensagem de e-mail. Para a verificação SPF ou DKIM, é necessário que o DNS resolva pelo domínio a qual pertence o servidor de e-mail remetente, já que este é o domínio pesquisado por um servidor destinatário. Entretanto, para que a estrutura de envio/recebimento de e-mail funcione para o cenário virtual, o servidor DNS resolve para os domínios *meudominio.com* e *valido.com.* e foi instalado na máquina *VM 2.* O projeto será dividido em dois cenários:

- Ambiente de e-mail implementado com ferramenta SPF;
- Ambiente de e-mail implementado com ferramenta DKIM;

A figura a seguir mostra a topologia para o primeiro cenário:

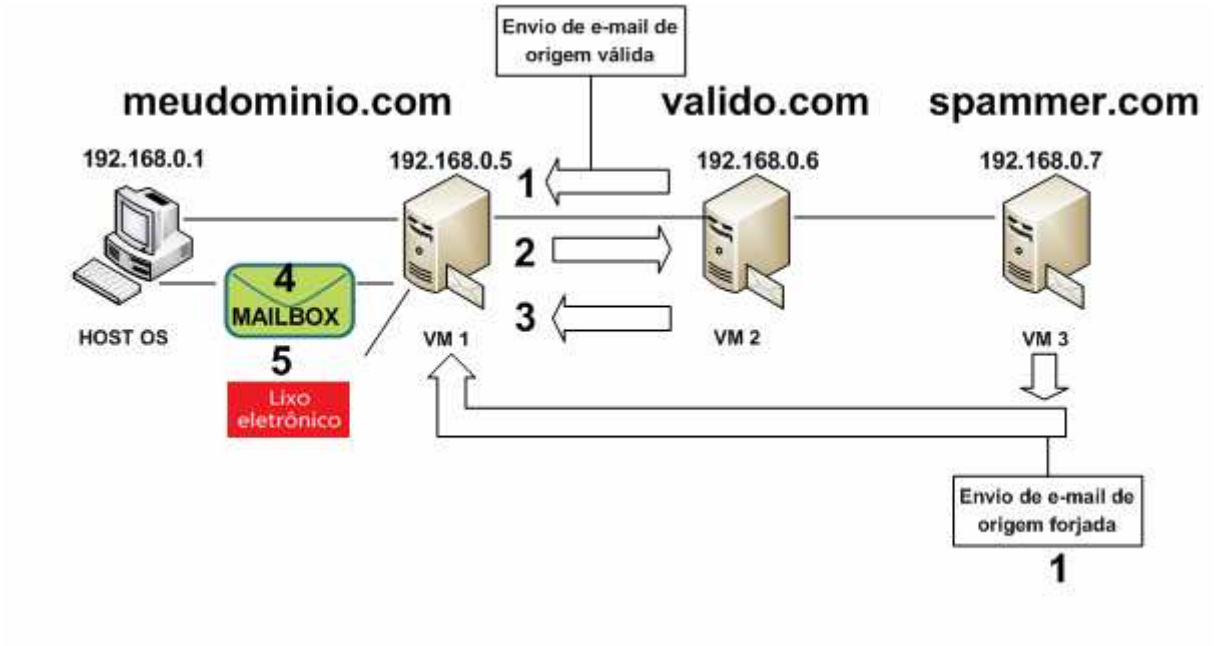

Figura 4.2 – Estrutura de verificação SPF no ambiente virtual.

Fonte: Elaborado por Autor [2008].

Supondo registro SPF cadastrado em DNS, e registros NS, MX e A:

IN TXT "v = spf1 mx – all"

IN NS srvlnx.valido.com

IN MX srvlnx.valido.com

srvlnx.valido.com IN A 192.168.0.6

Um e-mail é enviado de uma origem válida e outro e-mail é enviado de uma origem forjada tentando se passar por esta origem válida (1). O servidor de e-mail de destino (*VM 1*) checa o DNS responsável pelo domínio do remetente (*VM 2*) e pergunta a ele: "Existe registro SPF publicado?" (2). Supondo que exista, o DNS responde para o servidor. "De acordo com registro SPF, somente registro MX de *valido.com* está autorizado a enviar e-mails por este domínio" (3). Assim, a mensagem enviada de *srvlnx.valido.com* (registro NS que corresponde ao registro MX e que aponta para o registro A de IP 192.168.0.6) é entregue na caixa de correio do servidor *VM 1* (4), e a mensagem enviada do domínio do servidor *malvado.spammer.com* tentando se passar por *srvlnx.valido.com* é bloqueada (5), já que esta máquina possui IP 192.168.0.7 e, de acordo com registro SPF acima, somente o IP 192.168.0.6 está autorizado a enviar e-mails em nome de *valido.com.*

Da mesma forma, a topologia para o segundo cenário é apresentada abaixo:

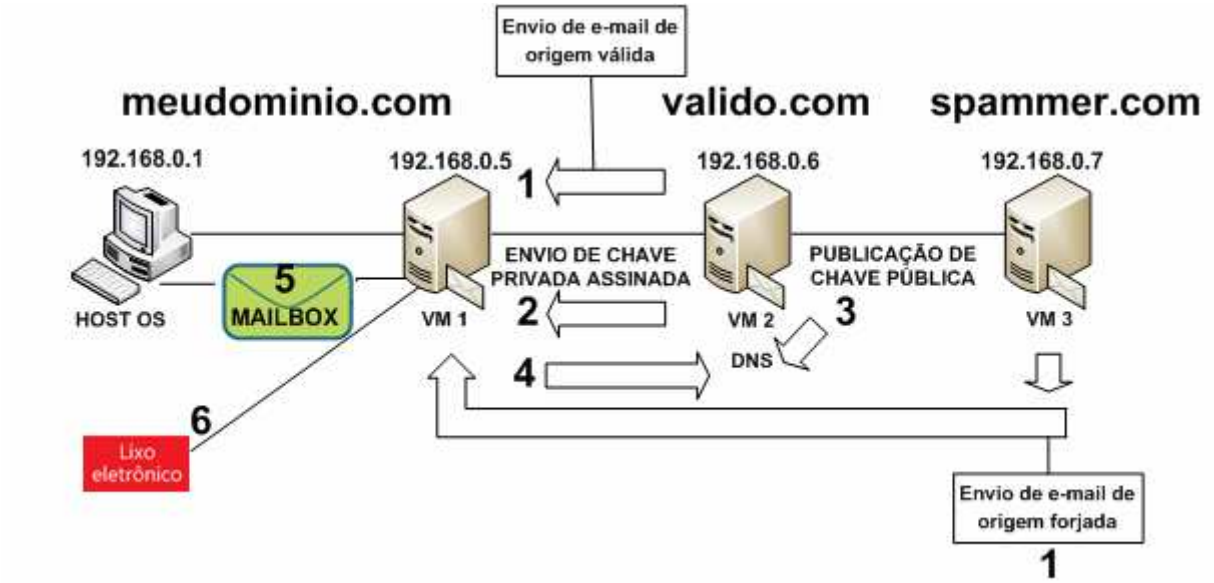

Figura 4.3 – Estrutura de verificação DKIM no ambiente virtual.

Fonte: Elaborado por Autor [2008].

Supondo registro DKIM cadastrado no DNS:

default._domainkey IN TXT "v=DKIM1; g=*; r=rsa p= MIGfMA...AB"

Supondo também registro DKIM presente no cabeçalho de e-mail:

v=1; a=rsa-sha256; c=relaxed/relaxed; d=gmail.com; s=gamma; h=domainkey-signature:received:received:message-id:date:from: subject:mime-version:content-type; bh=OpXbiNW…87paJS8=; b=rFn6r…2ulciI="

Um e-mail é enviado de uma origem válida e outro e-mail é enviado de uma origem forjada tentando se passar por esta origem válida (1). Assim, *srvlnx.valido.com* obtém uma chave privada e uma chave pública. Ele gera então um *valor hash* para o conteúdo da mensagem e utiliza sua chave privada para assinar este valor (assinatura = parâmetro *b* de registro DKIM do cabeçalho), e enviar o resultado para o servidor destinatário de correio *servidoremail.meudominio.com* (2). Em paralelo a isto, *srvlnx.valido.com* também registra uma chave pública (parâmetro *p* de registro DKIM do DNS) em seu servidor DNS, que no caso representa a própria máquina. (3). Logo, *servidoremail.meudominio.com*, ao receber a mensagem, checa o DNS responsável pelo domínio do remetente e verifica a existência de

registro DKIM pertinente a chave pública e outras informações de controle (4). Ele então recria o *valor hash* para a mensagem recebida, e utiliza a chave pública para decifrar a mensagem que foi assinada pelo emissor presente no cabeçalho de e-mail. Se este *valor hash*, obtido com a decifração desta assinatura, corresponder ao *valor hash* recriado da mensagem, significa que *srvlnx.valido.com* foi quem realmente enviou um e-mail para *servidoremail.meudominio.com* e a mensagem é depositada na caixa de correio deste servidor (5). Caso estes valores não sejam iguais, significa outro servidor tentou se passar pelo servidor *srvlnx.valido.com* (*malvado.spammer.com*) e a mensagem é bloqueada (6).

4.3 HARDWARE UTILIZADO

Como os testes foram feitos em notebook *BLUESKY* modelo *BLK-0207N*, segue configuração de *hardware* de *notebook* utilizado.

- CPU: *Intel Core Solo-1867* (1866.7 MHz);
- Memória: *DDR2-SDRM* (2048Mbytes);
- Placa-mãe:
  - Modelo: *Standard L41II3 and L41II4*;
  - Chipset: *Intel 82945GM(L/S/Z)/PM/GT (Calistoqa) + ICH7-M*;
- Disco Rígido: *HD IDE ST98823AS* (80Gbytes);
- Placa de rede: *Realtek RTL8139 Family PCI Fast Ethernet NIC*.

4.4 SOFTWARE UTILIZADO

4.4.1 Instalação e configuração dos servidores

Nesta parte serão descritos os processos de instalação e configuração dos servidores e seus respectivos serviços. Não será descrito o processo de instalação e configuração do sistema operacional e da aplicação cliente de e-mail (*Windows XP Professional* e *Mozilla Thunderbird* respectivamente) na máquina *HOST OS*, nem a instalação do sistema operacional

Mandriva Linux nas máquinas *VM 1*, *VM 2* e *VM 3* pela facilidade de instalação e configuração. Os serviços instalados foram separados por cada máquina virtual.

4.4.1.1 Máquina *VM 1*

Depois da instalação do sistema operacional, foi configurado um repositório da *Internet* para *download* de pacotes necessários para a implementação. O comando urpmi permite o *download* dos pacotes diretamente do repositório disponibilizado em *http://mandriva.c3sl.ufpr.br/official/2007.0/i586/media/,* caso os pacotes não seja encontrados nos DVDs de instalação da distribuição. Os seguintes procedimentos foram realizados nesta máquina.

- **Configuração de cliente DNS para localização do servidor DNS:**

Para que o *Postfix* possa resolver os nomes dos destinatários de e-mail, é preciso que ele se comunique com um servidor DNS (*VM 2*). Para isto, foi editado o arquivo */etc/**resolv.conf*** e especificado o servidor DNS no mesmo. Este arquivo é responsável pelo mapeamento do/s servidor/es DNS em um domínio.

[root@servidoremail /]# vi /etc/resolv.conf

#**Resolv.conf**
search localdomain
nameserver 192.168.0.6

- **Instalação do Postfix:**

Os seguintes pacotes foram instalados para configuração da máquina como servidora de e-mail.

[root@servidoremail /]# urpmi postfix-2.3.3-4mdv2007.0.i586.rpm libpostfix1-2.3.3-4mdv2007.0.i586.rpm

- **Configuração do serviço IMAP para transferência de correio:**

Para que o *Postfix* possa transferir correio para seu cliente IMAP, é preciso que ele seja configurado como servidor IMAP. Para isto, os seguintes pacotes foram instalados:

[root@servidoremail /]# urpmi courier-imap-4.1.1-8mdv2007.0.i586.rpm courier-base-4.1.1-8mdv2007.0.i586.rpm courier-authlib-0.58-9mdv2007.0.i586.rpm expect-5.43.0-7mdk.i586.rpmlibexpect-5.43.0-7mdk.i586.rpm

- **Instalação do serviço de verificação SPF:**

  Foram feitos os seguintes procedimentos para a realização da checagem SPF:

  - *Download* e instalação das bibliotecas *libspf2-1.2.5-4mdv2007.0.i586.rpm* e libspf2-devel-1.2.5-4mdv2007.0.i586.rpm:

    [root@servidoremail /]# urpmi libspf2-1.2.5-4mdv2007.0.i586.rpm libspf2-devel-1.2.5-4mdv2007.0.i586.rpm

  - *Download* e instalação do pacote *policyd-1.0.1.tar.gz*: O pacote policyd.1.0.1.tar.gz foi baixado do site *http://www.libspf2.org/patch/policyd-1.0.1.tar.gz* para o diretório */usr/lib/postfix/* e representa o *daemon* que implementa a verificação SPF na *VM 1*, quando compilado com as bibliotecas *libspf2-1.2.5-4mdv2007.0.i586.rpm* e *libspf2-devel-1.2.5-4mdv2007.0.i586.rpm*, além de outras dependências necessárias. Segue descompactação do pacote, instalação destas outras dependências e posterior compilação do programa.

    [root@servidoremail postfix]# tar xzvf policyd-1.0.1.tar.gz

    [root@servidoremail postfix]# urpmi gcc-4.1.1-3mdk.src.rpm binutils-2.16.91.0.7-3mdv2007.0.i586.rpm make-3.81-1mdv2007.0.i586.rpm libtool-1.5.20-9mdv2007.0.i586.rpm libtool-base-1.5.20-9mdv2007.0.i586.rpm libgdbm3-1.8.3-3mdv2007.0.rpm

    [root@servidoremail postfix]# ./configure

Lembrando que o comando *./configure* somente analisa o sistema e verifica as dependências necessárias, para gerar o *makefile*, *script* que de fato irá compilar o programa. Entretanto, na execução do comando *./configure* foram gerados alguns erros.

```
configure: WARNING: spf2/spf.h: present but cannot be compiled
configure: WARNING: spf2/spf.h: check for missing prerequisite headers?
configure: WARNING: spf2/spf.h: see the Autoconf documentation
configure: WARNING: spf2/spf.h: section "Present But Cannot Be Compiled"
configure: WARNING: spf2/spf.h: proceeding with the preprocessor's result
configure: WARNING: spf2/spf.h: in the future, the compiler will take precedence
configure: WARNING: ## ------------------------------------- ##
configure: WARNING: ## Report this to the AC_PACKAGE_NAME lists. ##
configure: WARNING: ## ------------------------------------- ##
checking for spf2/spf.h... yes
checking for SPF_destroy_config in -lspf2... no
libspf2 is required to build this program.
```

Figura 4.4 – Erros gerados na instalação do serviço de verificação SPF.

Fonte: Elaborado por Autor [2008].

Após pesquisas sobre este problema na *Internet*, foi descoberto que as versões das bibliotecas *libspf2-1.2.5-4mdv2007.0.i586.rpm* e *libspf2-devel-1.2.5-4mdv2007.0.i586.rpm* da *Mandriva* ainda não funcionam direito com o *policyd1.0.1.tar.gz.* (*http://www.gossamer-threads.com/lists/spf/devel/20403*). Por este motivo, foram ignorados os passos descritos anteriormente, e utilizada a biblioteca *mail-spf/Mail-SPF-v2.005.tar.gz* (baixado de *http://search.cpan.org/CPAN/authors/id/J/JM/JMEHNLE/mail-spf/Mail-SPF-v2.005.tar.gz)*, um módulo programado em linguagem *Perl* em substituição as bibliotecas *libspf2-1.2.5-4mdv2007.0.i586.rpm* e *libspf2-devel-1.2.5-4mdv2007.0.i586.rpm.* Um módulo nada mais é do que um programa ou conjunto de programas que são utilizados por *scripts* para o auxílio em diversas tarefas (neste caso para a verificação de checagem de remetente SPF). O *script* que fará a checagem SPF foi baixado de *http://www.openspf.org/blobs/postfix-policyd-spf-perl-2.005.tar.gz* entrando em substituição ao *policyd-1.0.1.tar.gz.* O *script policyd-spf.pl* contigo no pacote *postfix-policyd-spf-perl-2.005.tar.gz* foi transferido para o diretório */usr/lib/postfix***.** Como o suporte a linguagem *Perl* já veio instalado no *Mandriva* por padrão, foi preciso somente instalar o módulo *mail-spf/Mail-SPF-v2.005.tar.gz* e os módulos dependentes a este utilizando o *Shell CPAN*. O *CPAN* é um conjunto de repositórios para armazenagem de *scripts*, módulos, *frameworks*, etc, escritos em linguagem *Perl*, que podem ser disponibilizados via *Internet*.

```
[root@servidoremail  postfix]#  perl -MCPAN –e shell
cpan> install Mail::SPF
      install Net::DNS
      install IO::Socket
      install  MIME::Base64
      install Digest::MD5
      install  Net::IP
      install  Netaddr::IP
```

Com o módulo instalado, falta agora configurar o servidor *Postfix* para fazer esta verificação e referenciar o *script policyd-spf.pl* na configuração do mesmo. O *main.cf* (localizado em */etc/postfix*) é o principal arquivo de configuração e o *master.cf* (localizado em */etc/postfix*) representa o *daemon* que roda o serviço de servidor de e-mail. Foram editados estes dois arquivos. A primeira figura mostra as linhas que foram acrescentadas no *main.cf.* A configuração completa deste arquivo se encontra no anexo 1 desta monografia.

```
smtpd_recipient_restrictions =
        reject_unauth_destination
        check_policy_service unix:private/policy
```

Figura 4.5 - Linhas adicionadas ao arquivo *main.cf* que implementa verificação SPF.
Fonte: Elaborado por Autor [2008].

Em seguida, foi editado o arquivo *master.cf* e referenciado o *script* SPF *policyd-spf.pl* neste arquivo. Para isto foi acrescentado no meio do arquivo o conteúdo apresentado abaixo.

```
policy  unix    -       n       n       -       0       spawn
        user=nobody  argv=/usr/lib/postfix/policyd-spf.pl
```

Figura 4.6 - Linhas adicionadas ao arquivo *master.cf* que implementa verificação SPF.
Fonte: Elaborado por Autor [2008].

O conteúdo total do *master.cf* é muito grande, e por isto, será mostrado no anexo 2 (primeiro arquivo deste anexo) somente as linhas acrescentadas que implementam a checagem SPF ou DKIM, já que o DKIM também utilizará este arquivo em seu processo de verificação. Lembrando que para a implementação SPF são necessárias duas ações:

- A instalação da ferramenta de verificação SPF no servidor de e-mail de *meudominio.com* (*VM 1*), descrita neste tópico.
- A publicação de registro SPF no DNS responsável pelo domínio *valido.com*. (*VM 2*). Isto será detalhado no tópico 4.3.1.2, referente às configurações da máquina *VM 2*.

- **Instalação do serviço de verificação DKIM:**

O processo de implementação DKIM ocorre de maneira diferente do SPF, com a utilização do par de chaves privada/pública. A chave privada é de conhecimento somente do remetente ou assinante DKIM (*srvlnx.valido.com*) e a chave pública registrada no DNS de *valido.com* de acesso a todos (Ver figura 3.9).

Para a checagem de verificação DKIM por *servidoremail.meudominio.com*, foram feitos os seguintes procedimentos:

- De forma semelhante à verificação SPF foi instalado o módulo *Mail-DKIM-0.31* (*http://search.cpan.org/CPAN/authors/id/J/JA/JASLONG/Mail-DKIM-0.31.tar.gz*) com suas dependências.

  [root@servidoremail postfix]# perl -MCPAN –e shell

  [root@servidoremail postfix]# perl -MCPAN –e shell

  cpan> install Mail::DKIM

  install Digest::SHA1

  install Digest::SHA

  install Crypt::OpenSSL::RSA

  install Mail::Address

- Após isto foi baixado o *daemon* que implementa a verificação DKIM (*http://downloads.sourceforge.net/dkimproxy/dkimproxy-1.0.1.tar.gz*) e seguidos os passos do site *http://dkimproxy.sourceforge.net/* para sua correta instalação. Após instalação do programa, o *Postfix* precisou ser reconfigurado para fazer esta verificação. Esta versão utilizada do DKIM (*DKIM Proxy* – versão 1.0.1) necessita somente da edição do arquivo *master.cf*. O conteúdo acrescentado neste encontra-se no primeiro arquivo do anexo 2.

4.4.1.2 Máquina *VM 2*

Para máquina *VM 2* foram instalados os serviços servidores de e-mail e DNS. Seguem os passos tomados.

- **Instalação do *Postfix*:**

Da mesma forma que a máquina *VM 1*, foi instalado o *Postfix* como MTA responsável pelo domínio.

- **Instalação do pacote *Bind* (serviço DNS):**

Para que esta máquina seja servidora DNS, foi instalado o seguinte pacote:

[root@localhost /]# urpmi bind - 9.3.2 – 8mdv2007.0.i586.rpm

Após a instalação do pacote *bind*, foram alterados alguns arquivos do mesmo. Estas alterações definem as novas zonas de DNS criadas. Vamos analisar os arquivos alterados:

- *Arquivo* ***named.conf*** (localizado em */var/lib/named/etc/*) – Este é o principal arquivo de configuração do DNS. É o arquivo de define as zonas de DNS para os domínios *meudominio.com* e *valido.com.* As zonas são mapas que definem a estrutura hierárquica do DNS. Em cada zona é delimitado um arquivo que fornece informações sobre determinado domínio ou máquina local, como as localidades dos servidores de e-mail, web, ftp, dns, etc. A configuração deste arquivo é mostrada no anexo 3 desta monografia e o arquivos de zonas criados são referenciados a seguir.

- *Arquivo de zona* ***db.meudominio.com.hosts*** (localizado em */var/lib/named/var/named/zone/*) – Este arquivo foi criado para se referir a zona DNS do domínio *meudominio.com.* A configuração do mesmo é disposta no anexo 4.

- *Arquivo de zona* ***db.valido.com.hosts*** (localizado em */var/lib/named/var/named/zone/*) – Este arquivo foi criado para se referir a zona DNS do domínio *valido.com.* A configuração do mesmo é disposta no anexo 5.

- **Publicação de registro SPF no DNS**

A publicação do registro SPF foi feita no arquivo *db.valido.com.hosts.* A geração deste registro foi feita manualmente, mas poderia ser feita utilizando-se um *wizard* do próprio site do SPF, disponibilizado em *http://www.openspf.org/*. A primeira marcação em negrito presente no anexo 5 representa o registro TXT referente ao SPF.

- **Publicação de registro DKIM no DNS e assinatura da mensagem**

O processo de geração das chaves pública e privada deve ser feito pelo assinante DKIM ou remetente da mensagem (*srvlnx.valido.com*), de modo que a chave privada seja de posse somente dele e a pública deve estar presente no DNS responsável pelo seu domínio. Assim, com base nas instruções do site *http://dkimproxy.sourceforge.net/* foram feitos os seguintes procedimentos:

- Geração das chaves pública e privada. O *OpenSSL* foi usado para isto:

[root@servidoremail /]# openssl genrsa -out private.key 1024

[root@servidoremail /]# openssl rsa -in private.key – pubout -out public.key

- Utilização da chave pública gerada para criação do registro DKIM do tipo TXT. Foi utilizado o seletor padrão *default*. Este nome é o padrão quando não se é especificado nenhum seletor. Como já dito em teoria, o seletor serve para diferenciar mais de uma chave publicada para o mesmo domínio. Assim, o registro ficou desta forma:

default._domainkey IN TXT "k=rsa; t=s; p=MHwwDQYJK…"
Onde em *p* foi copiado e colado o valor referente à chave pública.

O conteúdo completo deste registro encontra-se na segunda marcação em negrito do anexo 5.

- Assinatura da mensagem – O segundo arquivo *master.cf* do anexo 2 define as configurações editadas no *Postfix* para esta assinatura.

4.4.1.3 Máquina *VM 3*

Para esta máquina foram feitos os seguintes procedimentos:

- Instalação e configuração do *Postfix* sem qualquer verificação SPF/DKIM. Quando não se implementa nenhuma destas ferramentas, o único arquivo que se necessita editar é o *main.cf* (anexo 1 – parte *Minhas Configurações*).

- Configuração de cliente DNS para encontrar servidor DNS. Da mesma forma que em máquina *VM 1*, foi configurado o arquivo *resolv.conf*.

[root@malvado /]# vi /etc/resolv.conf

#Resolv.conf
search localdomain
nameserver 192.168.0.6

- Adaptação de *script* que simula a situação de forjamento do domínio *spammer.com*, tentando se passar pelo domínio *valido.com*, para verificação do comportamento das ferramentas SPF e DKIM. Este *script* foi feito com base no modelo do artigo "Sending E-mail with Perl Best Practice", de Frank Wiles (*http://www.revsys.com/writings/perl/sending-email-with-perl.html*). O *script* adaptado para a situação de forjamento dos cenários 1 e 2 encontra-se no diretório raiz */spamming.pl.* (anexo 6)

CAPÍTULO 5. TESTES E ANÁLISE DE RESULTADOS

Neste capítulo serão mostrados os resultados que foram obtidos com os testes propostos para os cenários do projeto.

5.1 PROCEDIMENTO PARA *BENCHMARK*

5.1.1 Procedimento padrão para ambos os cenários

Para o estudo de atuação das ferramentas SPF e DKIM, serão analisadas três taxas de amostras recebidas pelo servidor s*ervidoremail.meudominio.com* do servidor *malvado.spammer.com*. Para a primeira amostra, serão enviados 10 *spams*, para a segunda 100 e para a terceira 1000 (o número total de mensagens é manipulado no *script spamming.pl*). Este *script* permite falsificar o campo "FROM" do cabeçalho de e-mail que identifica o domínio de origem. Desta forma, considerando *dickvigarista* como o nome usado para se falsificar um usuário real pertencente ao servidor de e-mail *srvlnx.valido.com*, *dickvigarista@spammer.com* aparecerá neste campo como *dickvigarista@valido.com*.

Em cada amostragem serão comparados os seguintes parâmetros:

- Tempo de processamento gasto pelo servidor para verificação SPF/DKIM;
- Tempo de consulta DNS gasto pelo servidor para consulta a registro SPF/DKIM;
- Taxa de falso-positivos gerados pelas ferramentas SPF/DKIM;
- Taxa de falso-negativos gerados pelas ferramentas SPF e DKIM.

5.1.2 Procedimento para Cenário 1

Em relação ao tempo de processamento gasto com verificação SPF, foram realizados os seguintes procedimentos:

- O servidor *Postfix* precisa estar rodando em ambas as máquinas. Logo, foi iniciado o serviço que sobe a configuração do *Postfix* nas duas máquinas.

Lembrando que a configuração do *Postfix* que sobe na máquina *VM 1* é a do servidor configurado com a ferramenta de verificação SPF.

Máquina *VM 1*: *[root@servidoremail /]# service postfix start*

Máquina *VM 3*: *[root@malvado /]# service postfix start*

- Em seguida, foi apagado o arquivo de *log* de *servidoremail.meudominio.com* e recriado este arquivo com o conteúdo vazio, para que os dados retirados deste *log* permitam mostrar somente o tempo gasto com a checagem SPF.

  Máquina *VM 1*: *[root@servidoremail /]# rm –rf /var/log/mail/info*

  [root@servidoremail /]# service syslog stop

  [root@servidoremail /]# service syslog start

- E por último, iniciado o script *spamming.pl*, sendo enviados 10 *spams* para *servidoremail.meudominio.com* referente a primeira amostragem.

  Máquina *VM 3*: *[root@malvado /]# ./spamming.pl*

Deste modo, foi possível medir o tempo de processamento gasto pelo servidor *servidoremail.meudominio.com* para a verificação SPF, analisando o tempo decorrido desde o primeiro e-mail enviado e o décimo. A figura a seguir mostra um desenho com as primeiras linhas do *log* deste servidor, mostrando o tempo inicial (13:25:43), aonde são feitas as primeiras tentativas de conexão de *malvado.spammer.com* com *servidoremail.meudominio.com* até a penúltima linha, onde a conexão é encerrada após verificação SPF inválida (13:25:46). Note que estas linhas representam o tempo gasto de verificação SPF somente para o primeiro e-mail. A última linha marca o início dos mesmos processos, onde é mostrado novamente o pedido de conexão de *malvado.spammer.com* com *servidoremail.meudominio.com*, agora para o segundo e-mail e assim consecutivamente.

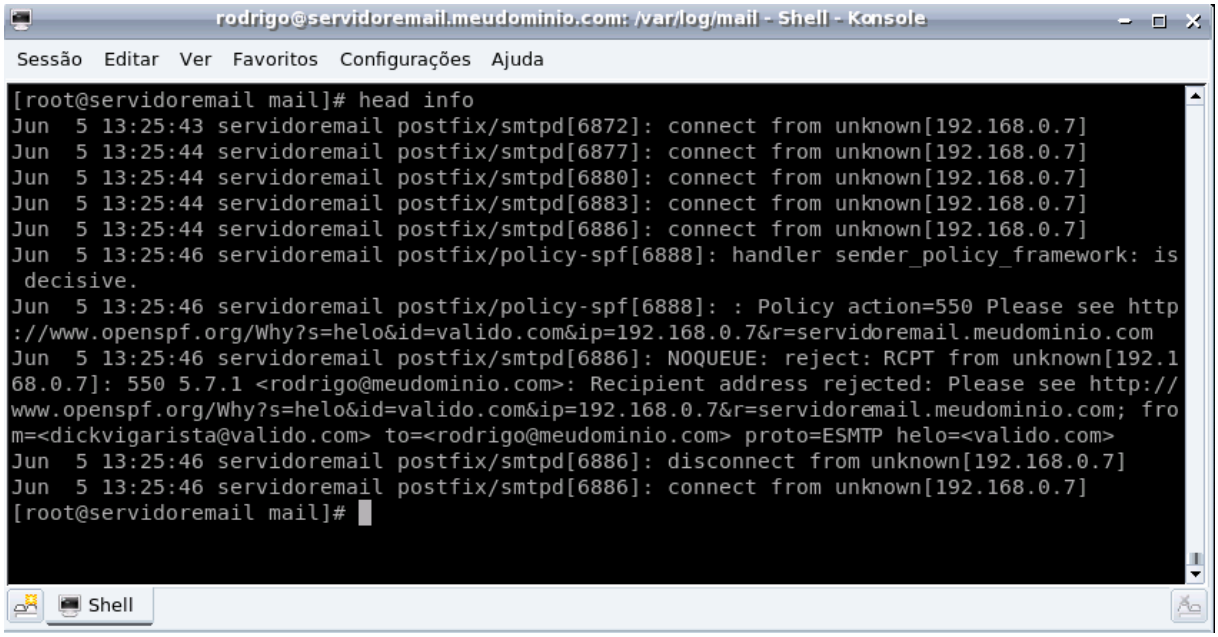

Figura 5.1 – *Log* de servidor de e-mail mostrando análise SPF.

Fonte: Elaborado por Autor [2008].

Considerando a figura acima, foi medido o tempo de processamento SPF gasto para as outras amostras. A tabela a seguir mostra todos estes dados:

Tabela 5.1 – Tempo de processamento para verificação SPF - Cenário 1.

Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	HORÁRIO DE VERIFICAÇÃO DO PRIMEIRO E-MAIL (hh:mm:ss)	HORÁRIO DE VERIFICAÇÃO DO ÚLTIMO E-MAIL (hh:mm:ss)	TEMPO DE PROCESSAMENTO PARA VERIFICAÇÃO SPF (em segundos)
ENVIO DE 10 SPAMS	13:25:43	13:25:51	8
ENVIO DE 100 SPAMS	13:39:45	13:40:13	28
ENVIO DE 1000 SPAMS	13:44:22	13:46:48	146

Em relação ao tempo de consulta DNS para pesquisa SPF, este é o tempo gasto por *servidoremail.meudominio.com* para pesquisa a um registro SPF do tipo TXT referente ao

domínio *valido.com.* Assim, o comando abaixo permite capturar o tempo relativo à pesquisa de registro referente a um e-mail (18 milissegundos).

Máquina *VM 1*: *[root@servidoremail /]# dig valido.com TXT*

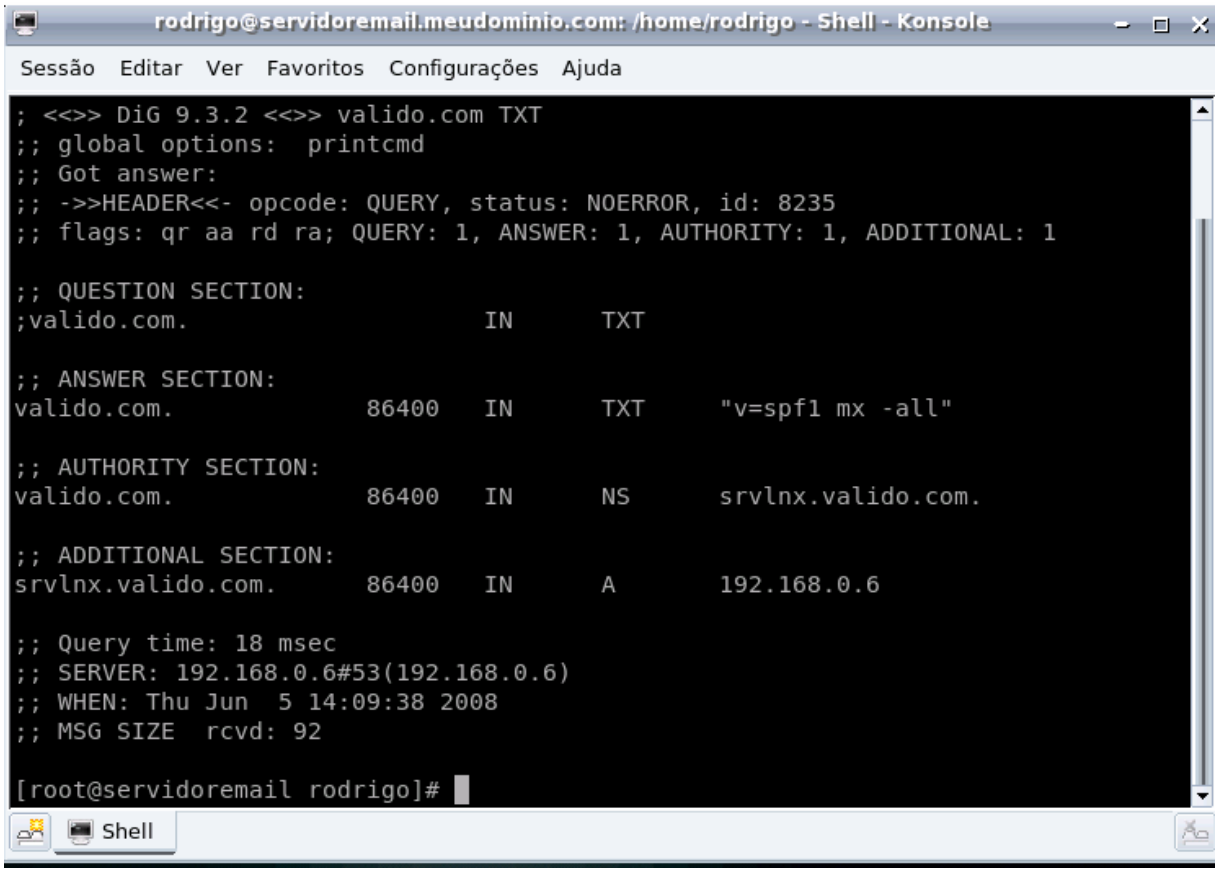

Figura 5.2 – Tempo de consulta DNS a registro SPF do tipo TXT.

Fonte: Elaborado por Autor [2008].

Assim, para se obter o tempo total de pesquisa DNS à registro SPF para as amostragens de 10, 100 e 1000 *spams*, basta multiplicar este tempo pelo valor das amostras. A tabela a seguir mostra estes resultados:

Tabela 5.2 – Tempo de consulta DNS a registro SPF.

Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	TEMPO DE PESQUISA DNS A REGISTRO SPF DO TIPO TXT (em milissegundos)
ENVIO DE 10 SPAMS	180
ENVIO DE 100 SPAMS	1800
ENVIO DE 1000 SPAMS	18000

Em relação à taxa de falso-positivos gerados, foi alterado o *script spamming.pl* e no campo "FROM" deste script foi colocado o endereço de um usuário real de e-mail (*rodrigo@valido.com*) pertencente ao servidor *srvlnx.valido.com*, e, logicamente, foram enviados e-mails reais deste servidor para o usuário de e-mail *rodrigo@meudominio.com* pertencente ao servidor *servidoremail.meudominio.com*. implementado com a ferramenta de verificação SPF. Este novo *script* (*envia.pl*) se encontra no diretório */home/rodrigo* da máquina *VM 2* e é mostrado no anexo 6 .

Sendo assim, a taxa de falso-positivos gerados pela verificação SPF foi nula, já que *servidoremail.meudominio.com* aceitou e armazenou todas as mensagens recebidas de *srvlnx.valido.com* referente as amostras de 10, 100 e 1000 e-mails reais. A figura 5.3 mostra o cliente de e-mail visualizando as mensagens recebidas pelo servidor para a amostra de 1000 e-mails:

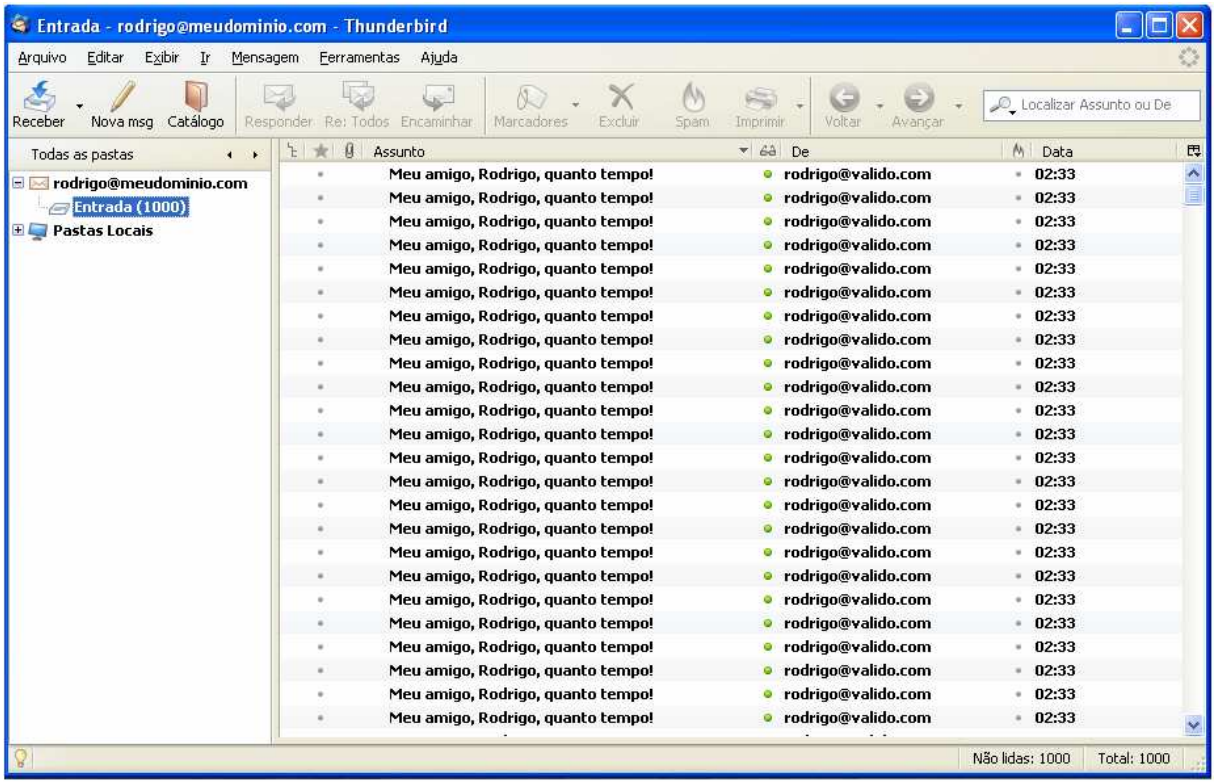

Figura 5.3 – Cliente de e-mail visualizando e-mails reais verificados com SPF.

Fonte: Elaborado por Autor [2008].

Esta próxima figura mostra a leitura do primeiro e-mail. Note que a marcação em negrito reflete o resultado da verificação SPF (*pass*), o que significa que este servidor aceitou todos os e-mails que de fato vieram de *srvlnx.valido.com*, comprovando a não ocorrência de falso-positivos.

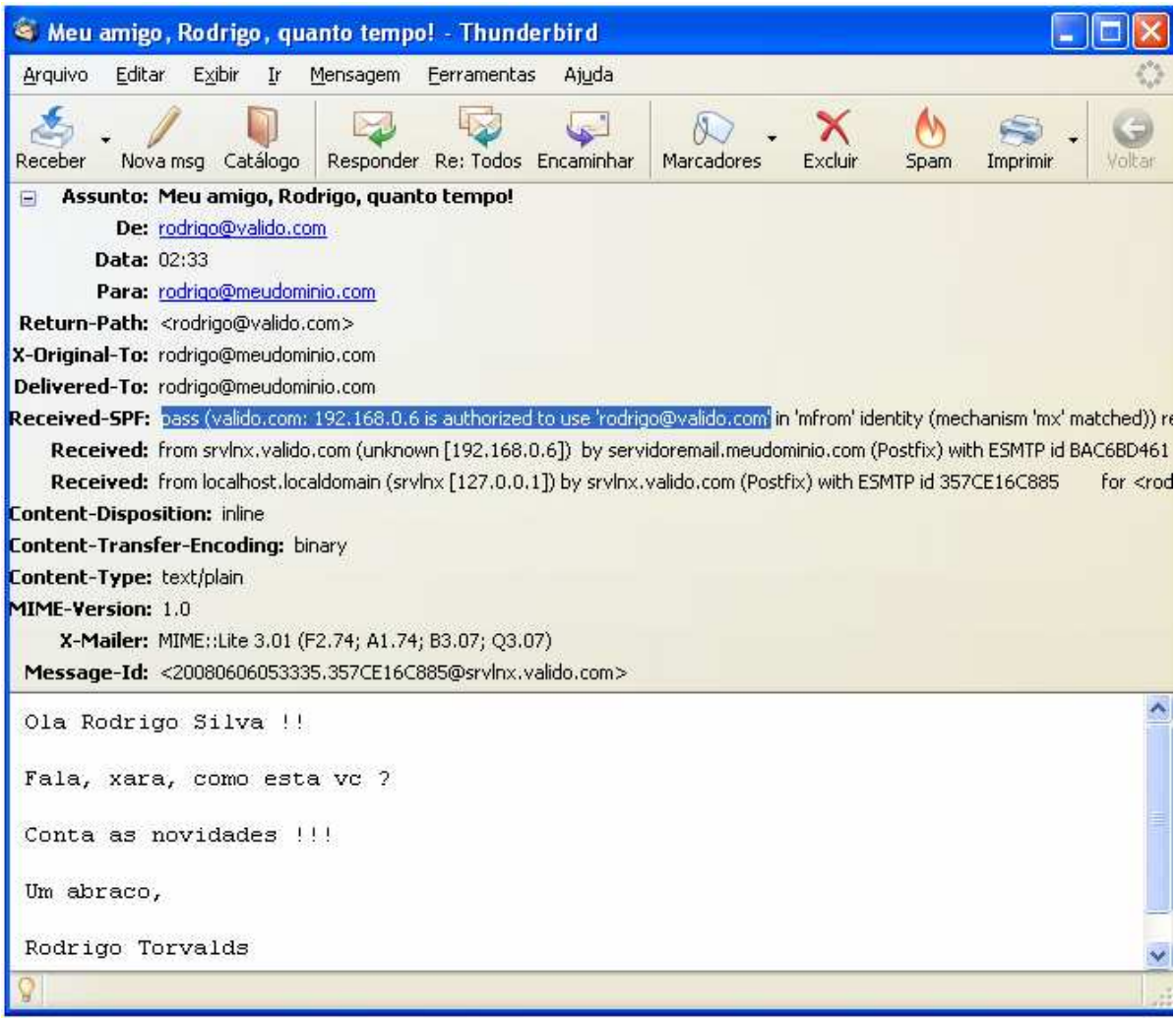

Figura 5.4 – Visualização de cabeçalho e-mail com resultado da verificação SPF.

Fonte: Elaborado por Autor [2008].

Para a análise de falso-negativos, a figura 5.1 comprova que a ferramenta SPF não gerou nenhum falso-negativo para as amostras de 10, 100 e 1000 *spams* enviados. O texto sublinhado da figura 5.1 reflete o comportamento do servidor de e-mail (*reject*) ao receber o primeiro *spam*, o que permite concluir que todos os *spams* enviados por *malvado.spammer.com* realmente foram classificados como *spams* pela ferramenta de verificação SPF presente em *servidoremail.meudominio.com*, e por isto foram rejeitados.

```
Jun  5 13:25:46 servidoremail postfix/smtpd[6886]: NOQUEUE: reject: RCPT from unknown[192.1
68.0.7]: 550 5.7.1 <rodrigo@meudominio.com>: Recipient address rejected: Please see http://
www.openspf.org/Why?s=helo&id=valido.com&ip=192.168.0.7&r=servidoremail.meudominio.com; fro
m=<dickvigarista@valido.com> to=<rodrigo@meudominio.com> proto=ESMTP helo=<valido.com>
```

Figura 5.5 – Comportamento do servidor ao recebimento do primeiro *spam* - Cenário 1.

Fonte: Elaborado por Autor [2008].

A tabela a seguir mostra o resumo de todos os parâmetros checados com a implementação SPF:

Tabela 5.3 – Resumo dos parâmetros analisados com checagem SPF.
Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	TEMPO DE PROCESSAMENTO PARA VERIFICAÇÃO SPF (em segundos)	TEMPO DE CONSULTA DNS A REGISTRO SPF (em segundos)	TAXA DE FALSO-POSITIVOS	TAXA DE FALSO-NEGATIVOS
ENVIO DE 10 E-MAILS	8	0.18	0	0
ENVIO DE 100 E-MAILS	28	1.8	0	0
ENVIO DE 1000 E-MAILS	146	18	0	0

5.1.3 Procedimento para Cenário 2

Para a captura dos resultados referentes à implementação DKM, foram realizados os mesmos procedimentos para a coleta de resultados SPF. Logo, foi iniciado o *Postfix* com suporte a verificação DKIM na máquina *VM 1*, iniciado o *Postfix* com suas configurações padrões na máquina do *spammer* (*VM 3*) e iniciado o serviço *dkimproxy*.

Máquina *VM 1*: *[root@servidoremail /]# service postfix start*

Máquina *VM 1*: *[root@servidoremail /]# service dkimproxy start*

Máquina *VM 3*: *[root@malvado /]# service proxy start*

Esta versão do DKIM exige, além do *Postfix* configurado, o serviço *dkimproxy*, responsável pela verificação da assinatura dos e-mails que chegam, bem como a assinatura dos e-mails que saem de um servidor. No caso acima, ele está configurado para verificar a assinatura dos e-mails que chegam a *servidoremail.meudominio.com.*

Em seguida, foram enviados os *spams* da máquina *VM 3* utilizando o *script spamming.pl*. Em relação ao tempo de processamento para verificação DKIM, a tabela a seguir mostra os valores capturados:

Tabela 5.4 – Tempo de processamento para verificação DKIM - Cenário 2

Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	HORÁRIO DE VERIFICAÇÃO DO PRIMEIRO E-MAIL (hh:mm:ss)	HORÁRIO DE VERIFICAÇÃO DO ÚLTIMO E-MAIL (hh:mm:ss)	TEMPO DE PROCESSAMENTO PARA VERIFICAÇÃO DKIM (em segundos)
ENVIO DE 10 SPAMS	19:50:45	19:50:49	4
ENVIO DE 100 SPAMS	19:56:08	19:56:18	10
ENVIO DE 1000 SPAMS	20:01:21	20:02:25	64

Em relação ao tempo de consulta DNS a registro DKIM, o comando abaixo permite capturar este valor para um e-mail (figura 5.6):

Máquina *VM 1*: *[root@servidoremail /]# dig default._domainkey.valido.com TXT*

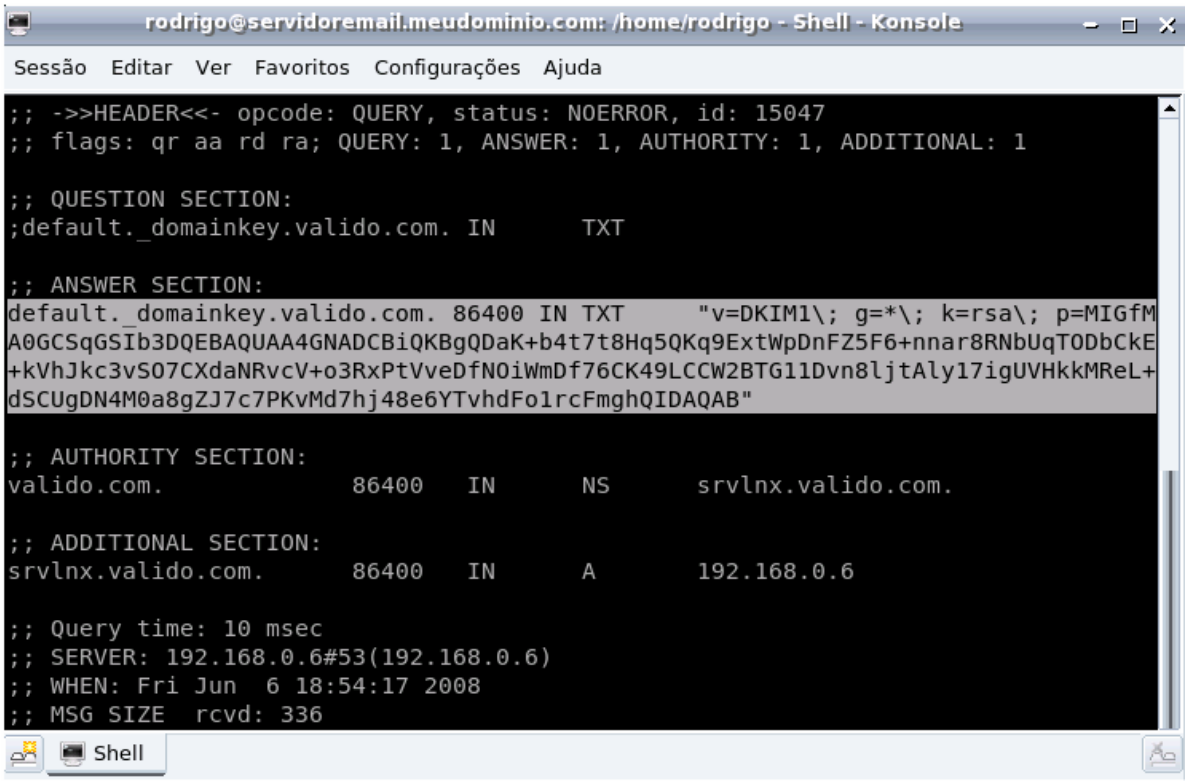

Figura 5.6 – Tempo de consulta DNS a registro DKIM do tipo TXT.

Fonte: Elaborado por Autor [2008].

A tabela abaixo mostra este tempo para as amostras de 10, 100 e 1000 *spams*:

Tabela 5.5 – Tempo de consulta DNS a registro DKIM.
Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	TEMPO DE PESQUISA DNS A REGISTRO DKIM DO TIPO TXT (em milissegundos)
ENVIO DE 10 SPAMS	100
ENVIO DE 100 SPAMS	1000
ENVIO DE 1000 SPAMS	10000

Em relação à taxa de falso-positivos, foram mandados e-mails reais utilizando o *script envia.pl* de *srvlnx.valido.com* para *servidoremail.meudominio.com*. Para isto, as duas máquinas precisam estar configuradas com suporte a DKIM. Veja os comandos abaixo:

Máquina *VM 1*: *[root@servidoremail /]# service postfix start*

Máquina *VM 1*: *[root@servidoremail /]# service dkimproxy start*

Máquina *VM 2*: *[root@servidoremail /]# service postfix start*

Máquina *VM 2*: *[root@servidoremail /]# service dkimproxy start*

Note que o serviço *dkimproxy* na máquina *VM 2* realiza a assinatura dos e-mails e o serviço *dkimproxy* na máquina *VM 1* realiza a conferência desta assinatura. A figura 5.7 mostra o cliente de e-mail visualizando as mensagens recebidas para a amostra de 1000 e-mails reais:

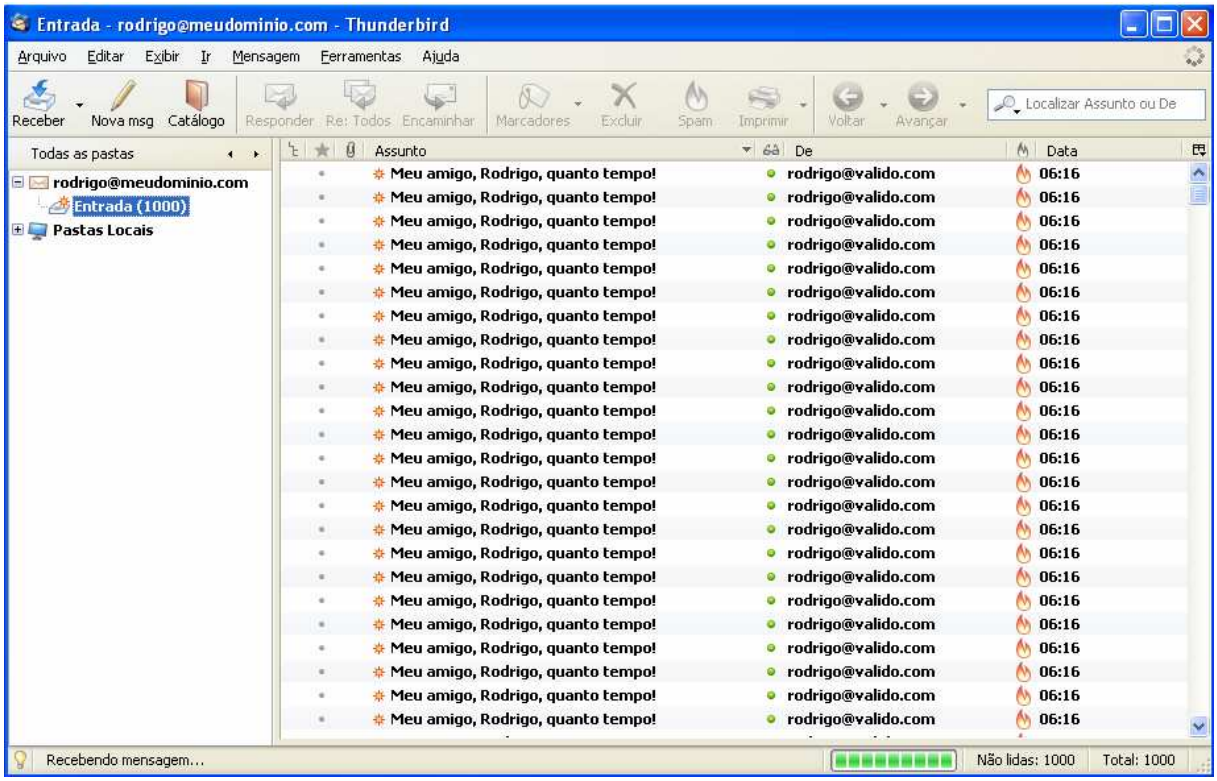

Figura 5.7 – Cliente de e-mail visualizando e-mails reais verificados com DKIM.

Fonte: Elaborado por Autor [2008].

Esta próxima figura mostra a leitura do primeiro e-mail. Note que o texto realçado reflete o resultado da verificação DKIM (*pass*), o que significa que este servidor aceitou todos os e-mails que de fato vieram de *srvlnx.valido.com*, comprovando a não ocorrência de falso-positivos.

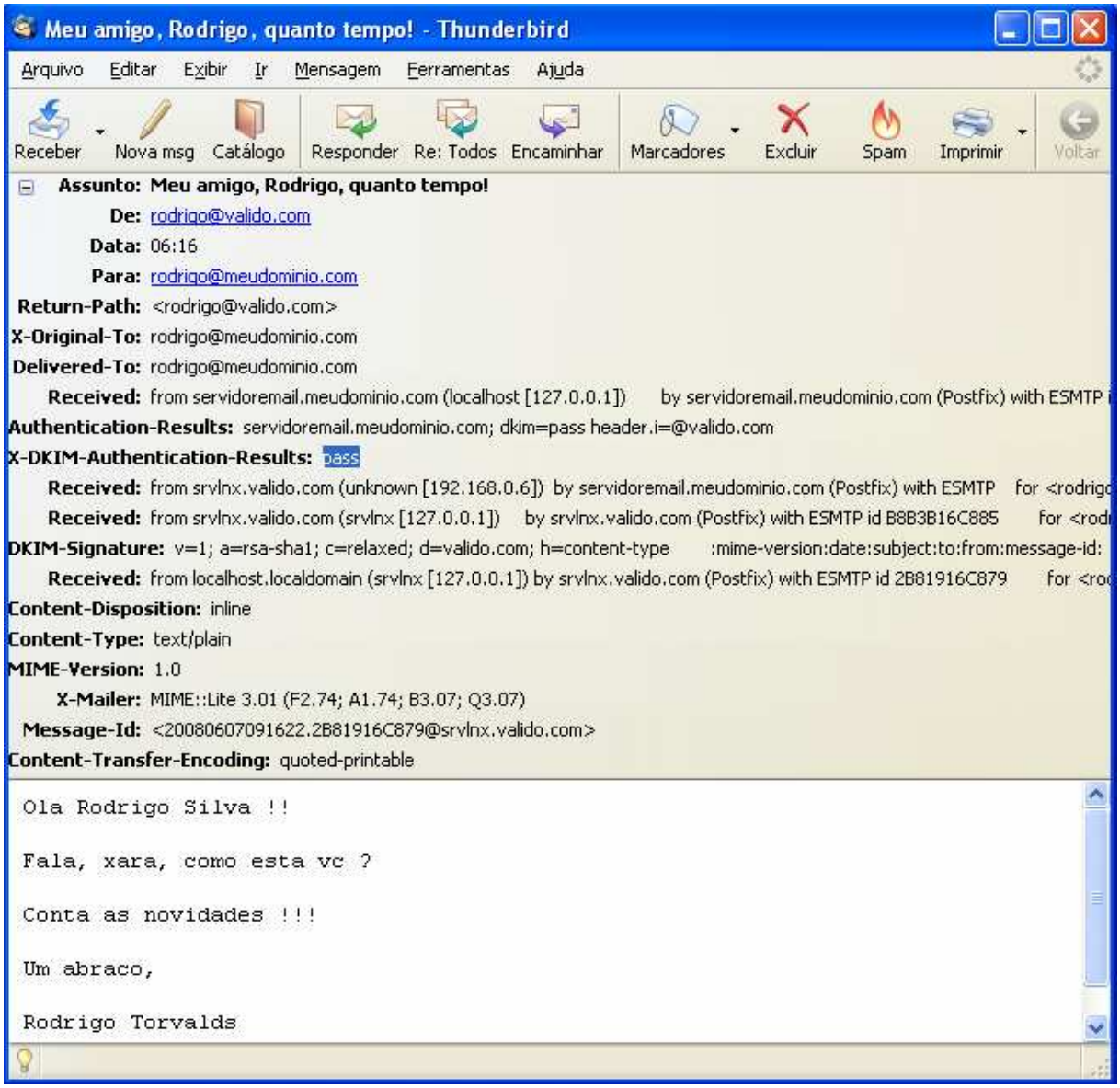

Figura 5.8 – Visualização de cabeçalho e-mail com resultado da verificação DKIM.

Fonte: Elaborado por Autor [2008].

Em relação à taxa de falso-negativos, esta também foi nula. A figura 5.9 mostra o *log* do servidor para o primeiro *spam* enviado. Note que o e-mail enviado de *dickvigarista@valido.com* não foi emitido de *srvlnx.valido.com*, mas sim de *malvado.spammer.com*. O texto em realce reflete o resultado do teste de verificação DKIM (*none)* e o texto sublinhado relata o comportamento do servidor (*discard*), o que significa que a mensagem foi descartada já quem realiza a assinatura das mensagens de *valido.com* é a máquina *srvlnx.valido.com* de IP 192.168.0.6 e não *malvado.spammer.com* de IP 192.168.0.7. Caso o *spammer* estivesse realizado alguma assinatura, o resultado da verificação seria *fail*.

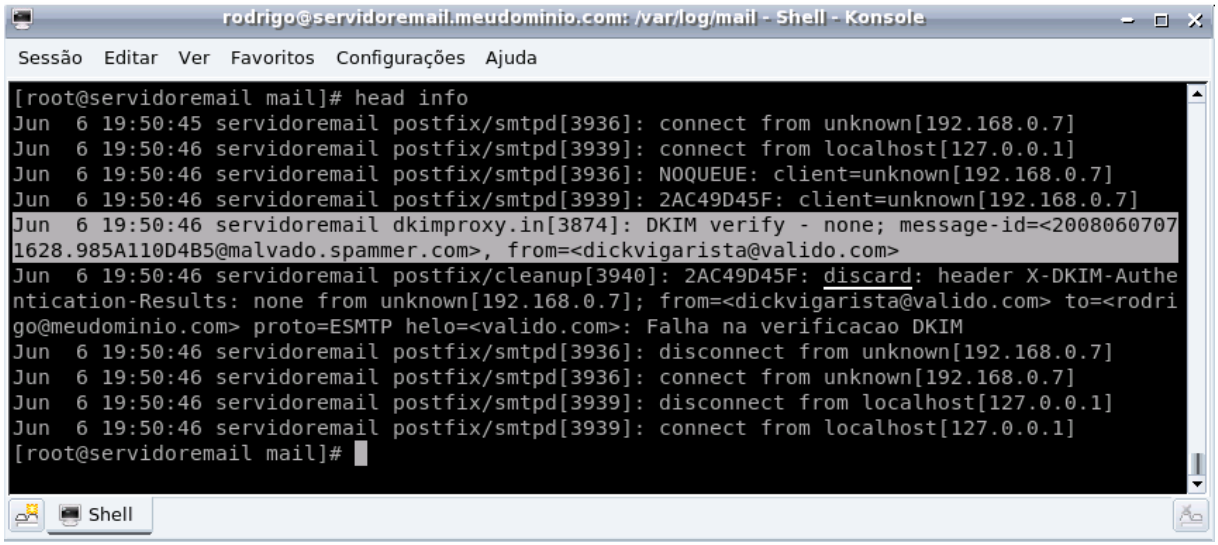

Figura 5.9 – Comportamento do servidor ao recebimento do primeiro *spam* - Cenário 2.

Fonte: Elaborado por Autor [2008].

A tabela a seguir mostra o resumo de todos os parâmetros checados com a implementação DKIM:

Tabela 5.6 – Resumo dos parâmetros analisados com checagem DKIM.

Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	TEMPO DE VERIFICAÇÃO DKIM (em segundos)	TEMPO DE CONSULTA DNS A REGISTRO DKIM (em segundos)	TAXA DE FALSO-POSITIVOS	TAXA DE FALSO-NEGATIVOS
ENVIO DE 10 E-MAILS	4	0.10	0	0
ENVIO DE 100 E-MAILS	10	1	0	0
ENVIO DE 1000 E-MAILS	64	10	0	0

5.2 RESULTADOS OBTIDOS

Comparando as tabelas 5.3 e 5.6, foi possível tirar as seguintes conclusões, de acordo com o resultado de cada teste realizado:

5.2.1 Tempo de processamento para verificação SPF/DKIM

Em relação ao tempo de processamento gasto para verificação SPF/DKIM, é possível concluir que o DKIM apresentou um melhor resultado.

Obtendo-se a soma das três amostragens, nota-se que o tempo de processamento gasto com a verificação SPF (182 segundos) foi superior ao tempo gasto para a verificação DKIM (78 segundos).

Entretanto, o tempo de processamento gasto pelo servidor sem qualquer mecanismo de checagem de origem de remetente implementado foi de 1, 6 e 62 segundos para as amostras de 10, 100 e 1000 *spams* respectivamente (ver tabela 5.7). Note que este é o tempo gasto para o servidor aceitar a mensagem e armazenar a mesma em disco.

Tabela 5.7 – Tempo de processamento gasto pelo servidor sem implementação SPF/DKIM.
Fonte: Elaborado por Autor [2008].

TAXA DE AMOSTRAGEM	**TEMPO INICIAL (PRIMEIRO SPAM RECEBIDO – em hh:mm:ss)**	**TEMPO FINAL (ÚLTIMO SPAM RECEBIDO – em hh:mm:ss)**	**TEMPO DE PROCESSAMENTO GASTO PELO SERVIDOR SEM MECANISMOS DE VERIFICAÇÃO SPF/DKIM (em segundos)**
ENVIO DE 10 SPAMS	18:03:21	18:03:22	1
ENVIO DE 100 SPAMS	18:06:55	18:07:01	6
ENVIO DE 1000 SPAMS	18:10:33	18:11:35	62
TEMPO TOTAL	-	-	69

Em relação ao SPF verificou-se um acréscimo de 113 segundos (182 – 69) do tempo de processamento gasto por um servidor que utilize esta verificação e um que não utilize, que permite concluir que este tempo adicional gasto é compensando pela economia de espaçamento em disco ocupado pelo acúmulo dos *spams* quando não utilizada esta ferramenta. Para as amostras de 10, 100 e 1000 *spams* o espaçamento em disco ocupado foi de aproximadamente 10470 bytes, 104700 bytes e 1047000 bytes (aproximadamente 1 Mbyte), respectivamente. O comando abaixo permite ver o espaço ocupado para o recebimento de um *spam (1047 bytes).*

> Máquina *VM 1*: *[root@servidoremail /]# ls –la /home/rodrigo/Maildir/new/121 3719154.V801 112c813M210349 .servidoremail.meudominio.com*

Em relação ao DKIM verificou-se um acréscimo de 9 segundos do tempo de processamento gasto por um servidor que utilize esta verificação (78 segundos) e um que não utilize (69 segundos), o que permite concluir sua melhor atuação em relação ao SPF, pois pequeno este pequeno acréscimo de tempo, economizou o mesmo espaçamento em disco para as amostras de 10 (10470 bytes), 100 (104700 bytes) e 1000 *spams* (1047000 Kbytes), quando não utilizada esta ferramenta.

5.2.2 Tempo de consulta DNS

O tempo gasto de pesquisa a um registro do DNS (TXT, A, MX, NS ou PTR) feita por um servidor de e-mail destinatário depende da proximidade ou não deste servidor ao servidor DNS responsável pelo domínio pesquisado. Portanto, este tempo não serve como parâmetro de comparação. No caso da verificação SPF ou DKIM, é feita a pesquisa a um registro do tipo TXT, e como o serviço de DNS foi instalado de maneira local para os cenários do *benchmark*, este tempo foi relativamente pequeno (18 milissegundos para pesquisa a um registro SPF contra 10 milissegundos para pesquisa a um registro DKIM para a análise de um e-mail).

5.2.3 Taxa de falso-positivos

A taxa de falso-positivos para os dois cenários foi nula. A ausência de falso-positivos indica que as ferramentas não bloquearam de fato nenhum e-mail real, o que comprova a eficácia destas duas tecnologias. Na *Internet*, a ocorrência de falso-positivos

gerados por estas ferramentas seria considerada somente se um domínio de origem utilizasse um servidor de e-mail que não estaria autorizado mediante política SPF ou DKIM, o que não faria sentido, já que um domínio real não precisaria disfarçar a identidade do seu servidor de e-mail.

5.2.4 Taxa de falso-negativos

A ausência de falso-negativos indica que as ferramentas bloquearam todos os e-mails que efetivamente são *spams*. Entretanto, no cenário da *Internet*, *spammers* podem registrar domínios válidos e implementar o SPF ou DKIM de maneira correta. Admitindo esta possibilidade, a ocorrência de falso-negativos seria considerável, dependendo do número de *spammers* que possuíssem seus próprios domínios e utilizassem corretamente estas ferramentas. Infelizmente não há como lançar alguma estatística com relação a isto, mas pode-se concluir que estas ferramentas implementadas por domínios reais corretamente contribuem para reduzir estes índices.

5.3 PROBLEMAS

Em relação à implementação SPF, o *policyd1.0.1.tar.gz*, pacote inicialmente baixado para realização da instalação do serviço de verificação SPF, não apresentou paridade com as bibliotecas disponibilizadas pelo *Mandriva* (*libspf2-1.2.5-4mdv2007.0.i586.rpm* e *libspf2-devel-1.2.5-4mdv2007.0.i586.rpm*). Para contornar o problema, foi instalado a versão 2.005 desenvolvida em linguagem *Perl* do SPF (*postfix-policyd-spf-perl-2.005.tar.gz*).

Em relação à implementação DKIM, encontrou-se uma dificuldade muito grande em encontrar pacotes já compilados (com extensão .rpm) disponibilizados pelos espelhos de *downloads* da *Mandriva Linux* que fizessem a análise DKIM. Os pacotes encontrados referiam-se ao *dkim-milter* e foram desenvolvidos para serem suportados em outras distribuições, como por exemplo *(//www.c-corp.net/linux/centos/5/general/RPMS/i386/dkim-milter-2.2.1-1.i386.rpm* e *http://rpm.pbone.net/index.php3/stat/4/idpl/6620576/com/dkim-milter-2.5.1-5.fc9.i386.rpm.html*). A primeira URL refere-se ao *download* de uma versão do *dkim-milter* projetada para trabalhar no *CentOS 5* e a segunda uma versão suportada pelo *Fedora Core 9*. Tendo em vista este problema, tentou-se instalar o pacote *dkim-milter-2.5.0.tar.gz* diretamente do código fonte, para que o mesmo fosse compilado e instalado no *Mandriva Linux*. Esta nova

tentativa também não foi bem sucedida, visto que esta ferramenta não realizava corretamente a autenticação DKIM após a sua compilação e instalação ao sistema. A solução encontrada foi instalar a versão desenvolvida em *Perl* do DKIM, o *dkimproxy-1.0.1.tar.gz*, como o nome do próprio pacote diz, em sua versão 1.0.1.

CAPÍTULO 6. CONCLUSÕES

Baseado na simulação de forjamento, levando em consideração as amostras de e-mails enviados pelo servidor de spams (*malvado.spammer.com*) para o servidor de recebimento de correio (*servidoremail.meudominio.com*), baseado também na simulação de envio de e-mails reais feita pelo servidor *srvlnx.valido.com* para o servidor *servidoremail.meudominio.com* (admitindo as mesmas taxas de amostragem) e considerando que os testes foram feitos em ambiente virtualizado, foi possível tirar as seguintes conclusões :

Em suma, estas ferramentas não demandam muito processamento em disco, pois permitem bloquear corretamente um *spam* sem que haja a necessidade de verificação da carga útil desta mensagem, substituindo mecanismos de listas-negras e dispensando a necessidade de uma verificação heurística completa para filtragem da mensagem.

A expectativa do resultado deste estudo era que o SPF apresentasse uma performance superior, já que não envolve nenhum método de criptografia para prover sua autenticação. Entretanto, tendo como base as duas versões implementadas em *Perl* (*policyd-spf-perl* versão 2.005 e *dkimproxy* versão 1.0.1), o DKIM, mais complexo em seu método de autenticação e até de sua própria configuração, apresentou um rendimento melhor, se mostrando mais atraente para sua futura utilização por uma empresa que queria aumentar seu nível de segurança de correio eletrônico sem qualquer custo adicional.

Todavia, as duas ferramentas se mostraram vantajosas, e o quesito que se deve analisar não é o tempo de processamento adicional gasto com a implementação destas ferramentas quando comparado ao tempo de processamento gasto quando não implementadas as mesmas, mas sim, a economia de espaço em disco que as mesmas proporcionaram evitando o acúmulo dos *spams*, além dos outros malefícios que estes e-mails indesejáveis podem causar. A economia de espaço em disco para as amostras de *spams* deste projeto foi pequena, porém, em uma situação de forjamento real, o espaço economizado representa fator importante considerando um volume efetivamente maior de *spams.*

A ausência de falso-positivos em ambos os cenários permitiu concretizar a eficiência e utilidade das tecnologias SPF e DKIM.

A análise de falso-negativos exige um estudo mais minucioso dos mecanismos de filtragem da mensagem, e se mostra relevante em um ambiente de rede real. As ferramentas e

estratégias que vêm sendo desenvolvidas para o combate aos *spams* estão cada vez mais avançadas, mas infelizmente ainda estão um passo atrás dos *spammers.* A possibilidade de *spammers* implementarem corretamente tecnologias como o SPF e o DKIM, e usarem estas próprias ferramentas para incentivar a falsificação de remetentes representa até uma ironia, já que as mesmas foram criadas para combater os *spams*, e não propagá-los. Todavia, o estudo cada vez mais detalhado de ferramentas como estas, elaboradas para suprir as vulnerabilidades do protocolo SMTP, que quando desenvolvido não se preocupou em garantir a autenticidade de uma mensagem, irá ajudar muito na redução do *spam.*

Em suma, a conclusão geral é que a checagem correta destes mecanismos depende da adequação da política SPF ou DKIM por parte dos dois domínios interessados no processo de envio/recebimento de correio, o que significa que não adianta nada um domínio destinatário de uma mensagem de correio implementar um mecanismo de verificação SPF ou DKIM se o domínio de origem não se adequou a política ou não publicou nenhum registro SPF ou DKIM no seu servidor DNS. Milhões de empresas e detentores de domínios já implementam as ferramentas SPF e/ou DKIM em seu sistema de correio. Assim, se uma empresa fica em dúvida na escolha de uma delas como solução alternativa de segurança de e-mail, a solução é implementar as duas para se obter uma melhoria no desempenho de um servidor de e-mail proporcionado pela redução dos casos de forjamento, já que uma ferramenta é independente da outra, o que significa que a implementação de uma não influência na implementação da outra, permitindo deduzir que o SPF resolverá casos em que *spammers* implementem corretamente o DKIM, mas não implementem o SPF em suas tentativas de ataque, e vice-versa.

Como sugestão para o desenvolvimento de projetos futuros, poderia ser realizado um estudo de caso de outras ferramentas que atuam no combate à falsificação de remetentes de correio, as quais utilizam de outros métodos para o processo da autenticação de origem. Por exemplo:

- Comparar a atuação da ferramenta SPF com a ferramenta GREYLISTING mediante simulações de envio de e-mails reais e forjados;
- Comparar a atuação da ferramenta DKIM com a ferramenta GREYLISTING mediante simulações de envio de e-mails reais e forjados;
- Comparar a atuação da ferramenta SPF com a ferramenta SENDER-ID mediante simulações de envio de e-mails reais e forjados;

- Comparar a atuação da ferramenta DKIM com a ferramenta SENDER-ID mediante simulações de envio de e-mails reais e forjados;
- Comparar a atuação da ferramenta GREYLISTING com a ferramenta SENDER-ID mediante simulações de envio de e-mails reais e forjados.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLMAN, E. **RFC4871 – DomainKeys Identified Mail (DKIM) Signatures.** Maio 2007. *http://www.ietf.org/rfc/rfc4871.txt* - Jan.2008.

ANTISPAM.br. **Administradores – Listas de Bloqueio.** 2008b. Disponível em <*http:www.antispam.br/admin/listas-de-bloqueio/#2*>. Acesso em: Jan.2008.

ANTISPAM.br. **O que é spam?** 2008a. Disponível em <*http://www.antispam.br/conceito/*>. Acesso em: Jan.2008.

BÄCK, Magnus. **Servidor de E-mail Linux – Guia de Instalação, Configuração e Gerenciamento para Pequenos Escritórios**. Editora Pearson Prentice Hall. 1ª Edição - 2007 - 284 pág.

CROCKER, David. **RFC0822 – Standard for the format of Arpa Internet Text Messages.** Agosto 1982. *http://www.ietf.org/rfc/rfc0822.txt* - Jan.2008.

DKIM.org. **DomainKeys Identified Mail**. 2008. Disponível em: <*www.dkim.org*>. Acesso em: abr. 2008.

JUNIOR, Fred. **MUTT – Guia Definitivo.** 2007. *http://www.dicas-l.com.br/dicas-l/20050601.php* - Dez.2007

LINEX – **Assessoria Especializada em Segurança da Informação**. 2008. Disponível em <*http://www.lynex.com.br/protocolosmtp*>. Acesso em: Mar. 2008.

MARCELO, Antonio. **Postfix – Guia rápido do administrador de redes**. Editora Brasport. 1ª Edição - 2004 - 89 pág.

NUNES, Danton. **Spam e Fraudes –** Técnicas de mitigação para administradores de redes. Novembro 2006. *www.cert.br/docs/palestras/certbr-ssi2006-2.pdf* - Mar.2008.

SPF. **Sender Policy Framework – Project Overview.** 2008. Disponível em: <*www.openspf.org*>. Acesso em: abr. 2008.

WIKIPEDIA, a enciclopédia livre. **E-mail.** 2007. Disponível em <*http://pt.wikipedia.org/wiki/E-mail*>. Acesso em: Set.2007.

WIKIPEDIA, a enciclopédia livre. **Engenharia Social – Segurança da Informação.** 2008a. Disponível em < *http://pt.wikipedia.org/wiki/Engenharia_social_%28seguran%C3%A7a_da_informa%C3%A7%C3%A3o%29*>. Acesso em: Jan.2008.

WIKIPEDIA, a enciclopédia livre. **RSA.** 2008b. Disponível em <*http://pt.wikipedia.org/wiki/RSA*>. Acesso em: Mar.2008.

WRONG, M. **RFC4408 – Sender Policy Framework (SPF) for Authorizing Use of Domains in E-mail, Version 1.** Abril 2006. *http://www.ietf.org/rfc/rfc4408.txt* - Jan.2008.

ANEXO 1 – MAIN.CF

O *main.cf* é o principal arquivo de configuração do *Postfix*. As linhas referentes à *Minhas configurações* representam o conteúdo deste arquivo sem qualquer implementação SPF ou DKIM. (Máquinas *VM 2* e *VM 3*). As linhas negritadas representam a configuração deste arquivo para a checagem de verificação SPF feita pela máquina *VM 1*. Logicamente as linhas referentes à *Minhas configurações* são pré-requisitos para o funcionamento desta checagem, e por tal motivo, devem permanecer configuradas em máquina *VM 1*.

```
#Arquivo main.cf

#Minhas Configurações
inet_interfaces = all
mynetworks_style = subnet
mynetworks = 192.168.0.0/24, 127.0.0.0/8
smtpd_banner = ESMTP - SERVIDOR DE EMAIL
unknown_local_recipient_reject_code = 450
smtp-filter_destination_concurrency_limit = 2
lmtp-filter_destination_concurrency_limit = 2
mydestination = $myhostname, localhost.$mydomain, $mydomain
myorigin = $mydomain
myhostname = servidoremail.meudominio.com
relay_domains = $mydomain
mydomain = meudominio.com
home_mailbox = Maildir

#Implementação SPF
smtpd_recipient_restrictions
      reject_unauth_destination
      check_policy_service unix:private/policy
```

ANEXO 2 – MASTER.CF

O *master.cf* é o arquivo *daemon* responsável por gerenciar o servidor de e-mail Postfix. Segue abaixo as linhas adicionadas à configuração do *master.cf* para o processo de checagem SPF e DKIM feita pelo servidor destinatário de correio (máquina *VM 1*):

```
#Arquivo master.cf
#
#Verificação SPF
#
policy   unix -      n      n      -      0      spawn
          user=nobody    argv=usr/lib/postfix/policy-spf.pl
#
#Verificação DKIM
#
smtp      inet  n      -      n      -      -      smtpd
   -o smtpd_proxy_filter=127.0.0.1:10025
   -o smtpd_client_connection_count_limit=10
127.0.0.1:10026 inet n  -      n      -      -      smtpd
   -o smtpd_authorized_xforward_hosts=127.0.0.0/8
   -o smtpd_client_restrictions=
   -o smtpd_helo_restrictions=
   -o smtpd_sender_restrictions=
   -o smtpd_recipient_restrictions=permit_mynetworks,reject
   -o smtpd_data_restrictions=
   -o mynetworks=127.0.0.0/8
   -o receive_override_options=no_unknown_recipient_checks
```

Segue abaixo as linhas adicionadas à configuração do *master.cf* para o processo de assinatura DKIM feita pelo servidor remetente de correio (máquina *VM 2*):

```
#Arquivo master.cf
#
#Assinatura DKIM da mensagem
#
smtp           inet  n      -      y      -      -      smtpd
   -o smtpd_etrn_restrictions=reject
   -o smtpd_sasl_auth_enable=yes
   -o content_filter=dksign:[127.0.0.1]:10027
   -o receive_override_options=no_address_mappings
   -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject

dksign         unix -      -      n      -      10     smtp
   -o smtp_send_xforward_command=yes
   -o smtp_discard_ehlo_keywords=8bitmime,starttls

127.0.0.1:10028 inet n  -      n      -      10      smtpd
   -o content_filter=
   -o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
   -o smtpd_helo_restrictions=
```

```
-o smtpd_client_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o smtpd_authorized_xforward_hosts=127.0.0.0/ 8
```

ANEXO 3 – NAMED.CONF

O *named.conf* é o principal arquivo de configuração do DNS, responsável por definir as zonas ou mapeamentos de domínios. A parte negritada refere-se às zonas criadas referentes aos domínios *meudominio.com* e *valido.com* presentes em máquina *VM 2*:

```
#Arquivo named.conf
#
key mykey {
   algorithm hmac-md5;
   secret "";
};

controls {
   inet 127.0.0.1 port 953
   allow { 127.0.0.1; } keys { mykey; };
};

options {
   directory "/var/named";
   pid-file "/var/named/named.pid";
   version "Wizard drake";
   allow-query { any; };
   allow-transfer { any; };
};

zone "localhost" {
   type master;
   file "zone/db.localhost";
};

zone "0.0.127.in-addr.arpa" {
   type master;
   file "zone/db.127.0.0.1";
};

zone "." {
   type hint;
   file "zone/root.hints";
};

zone "0.168.192.in-addr.arpa" {
   type master;
   file "zone/db.0.168.192.hosts";
   forwarders { };
};

zone "valido.com" {
   type master;
   file "zone/db.valido.com.hosts";
```

```
   forwarders { };
};

zone "meudominio.com" {
   type master;
   file "zone/db.meudominio.com.hosts";
   forwarders { };
};
```

ANEXO 4 – DB.MEUDOMINIO.COM.HOSTS

O *db.meudominio.com.hosts* refere-se ao arquivo de zona para o domínio *meudominio.com*:

```
#Arquivo db.meudominio.com.hosts
$TTL 1D
@       IN      SOA     srvlnx.meudominio.com. root.servidoremail.meudominio.com. (
              2008050801      ; Serial
                        8H  ; Refresh
                        2H  ; Retry
                        4W  ; Expire
                        1D) ; Minimum TTL
                        IN      NS          srvlnx.meudominio.com.
                        IN      MX   00   servidoremail.meudominio.com.
localhost                       A           127.0.0.1
dnsmaster                           IN      CNAME   srvlnx.meudominio.com.
servidoremail.meudominio.com.       IN      A       192.168.0.5
srvlnx.meudominio.com.              IN      A       192.168.0.6
```

ANEXO 5 – DB.VALIDO.COM.HOSTS

O *db.valido.com.hosts* refere-se ao arquivo de zona para o domínio *valido.com.* As marcações em negrito representam os registros SPF e DKIM do tipo TXT:

```
#Arquivo db.valido.com.hosts
$TTL 1D
@       IN      SOA     srvlnx.valido.com. root.servidoremail.meudominio.com. (
            2008050801      ; Serial
                        8H  ; Refresh
                        2H  ; Retry
                        4W  ; Expire
                        1D) ; Minimum TTL
                        IN      NS          srvlnx.valido.com.
                        IN      MX   00   srvlnx.valido.com.
                                TXT     "v=spf1 mx -all"
localhost                       A           127.0.0.1
dnsmaster                           IN      CNAME   srvlnx.valido.com.
srvlnx.valido.com.                  IN      A       192.168.0.6
default._domainkey IN TXT "v=DKIM1; g=*; k=rsa;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDaK+b4t7t8Hq5QKq9ExtWpDnF
Z5F6+nnar8RNbUqTODbCkE+kVhJkc3vSO7CXdaNRvcV+o3RxPtVveDfNOiWmDf76CK49L
CCW2BTG11Dvn8ljtAly17igUVHkkMReL+dSCUgDN4M0a8gZJ7c7PKvMd7hj48e6YTvhdFo1
rcFmghQIDAQAB" ; ----- DKIM default for valido.com
```

ANEXO 6 – SPAMMING.PL

O *spamming.pl* representa o script que simula o envio de *spams* pela máquina *malvado.spammer.com* para a máquina *servidoremail.meudominio.com*, tentando forjar a identidade do servidor de e-mail do domínio *valido.com*:

```
#Arquivo spamming.pl
#!/usr/bin/perl
use strict;
use warnings;
use MIME::Lite::TT;

my $numero_de_emails = 1000;
my %params;
$params{nome} = 'Rodrigo Silva';

my $template = <<TEMPLATE;

Ola [% nome %] !!

Você foi premiado!! Favor colocar seu CPF, conta
bancaria e senha no site http://www.acreditesequiser.com/
E receba R\$ 1.000.000,00.

Atenciosamente
Caixa Econômica Federal

   TEMPLATE

my %options;
$options{INCLUDE_PATH} = '/root/templates';
$options{EVAL_PERL} = 1;

my $msg = MIME::Lite::TT->new(
             From        =>  'dickvigarista@valido.com',
             To          =>  'rodrigo@meudominio.com',
             Subject     =>  'Você foi sorteado!',
             Template => \$template
             TmplOptions =>  \%options,
             TmplParams  =>  \%params,
             );

for (my $i = 0; $i < $numero_de_emails; $i++) {
             $msg->send('smtp', 'localhost', Timeout => 60);
};
```

ANEXO 7 – ENVIA.PL

O *envia.pl* representa o script *spamming.pl* adaptado para uma situação real, onde são enviados e-mails legítimos da máquina *srvlnx.valido.com* (*VM 2*) para a máquina *servidoremail.meudominio.com* (*VM 1*):

```
#Arquivo spamming.pl
#!/usr/bin/perl

 use strict;
 use warnings;
use MIME::Lite::TT;

 my $numero_de_emails = 1000;
 my %params;
 $params{nome} = 'Rodrigo Silva';

 my $template = <<TEMPLATE;

Ola [% nome %] !!

Fala, xará, como está você ?
Conta as novidades !!!
Um abraço,

Rodrigo Torvalds

TEMPLATE

 my %options;
 $options{EVAL_PERL} = 1;

 my $msg = MIME::Lite::TT->new(
              From        =>  'rodrigo@valido.com',
              To          =>  'rodrigo@meudominio.com',
              Subject     =>  'Meu amigo, Rodrigo, quanto tempo!',
              Template => \$template,
              TmplOptions =>  \%options,
              TmplParams  =>  \%params,
            );

for (my $i = 0; $i < $numero_de_emails; $i++) {
              $msg->send('smtp', 'localhost', Timeout => 60);
};
```