

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

Eduardo Juchem
Luciana Schmitz

Implantação do Webmail Seguro

Trabalho de Conclusão de Curso

Prof. Ricardo Felipe Custódio, Dr.

Florianópolis, Junho de 2004

Implantação do Webmail Seguro

Eduardo Juchem

Luciana Schmitz

Este Trabalho de Conclusão de Curso foi aprovado em sua forma final pelo Curso de Ciência da Computação da Universidade Federal de Santa Catarina.

Prof. Ricardo Felipe Custódio, Dr.

Prof. José Mazzuco Júnior, Dr.

Banca Examinadora

Prof. Júlio da Silva Dias, M.Eng.

Fabiano Castro Pereira

Marcelo Luiz Brocardo, Ms

*”Para ter sucesso neste mundo, é preciso ou ser louco ou
ser sábio”
(Barão de Montesquieu)*

Ofereço este trabalho aos meus pais João Carlos e Helga, como uma forma de agradecimento e reconhecimento por toda ajuda e apoio que recebi em minha vida e durante todo o decorrer deste curso.

(Eduardo Juchem)

Ofereço este trabalho de conclusão de curso à minha mãe Irene Martins Schmitz por sempre ter confiado e acreditado em mim em todos os momentos de minha vida. E aos meus irmãos, Vanderléia e Marcelo, pelo apoio e confiança.

(Luciana Schmitz)

Agradecimentos

Primeiramente ao grande amigo Leonardo Neves Bernardo, que nas horas de dificuldade e dúvidas esteve sempre a disposição, fosse tarde da noite ou finais de semana. Sem ele, este trabalho não teria se realizado. Amigos de verdade, temos poucos na vida, você Leonardo com certeza é um deles. Um abraço, e muito obrigado por tudo.

À minha namorada Mírian, pela imensa compreensão nos momentos de ausência, e pelo amor, amizade e companheirismo demonstrados em todos os momentos. Obrigado. Eu te amo.

À colega Luciana Schmitz, pelo esforço e dedicação na realização deste trabalho.

Ao professor Ricardo Felipe Custódio, pela amizade e orientação no projeto.

Ao professor Olinto José Varela Furtado, cujo exemplo de profissionalismo, seriedade e competência vou levar para sempre em minha vida profissional.

(Eduardo Juchem)

Ao amigo Eduardo Juchem pela parceria e dedicação no projeto.

Ao orientador, pela idéia do trabalho de conclusão de curso.

Ao meu namorado, Emerson Costa, pela compreensão nos momentos de ausência física e espiritual e pelo cuidado e atenção nos momentos de desespero.

Aos amigos e colegas do curso, que nos momentos de estudos e estresse souberam compreender determinadas atitudes e algumas vezes chamar a atenção, sem contar os momentos de lazer. Amigos e colegas que sempre levarei na lembrança com muito carinho: Sandra Alves da Mata, Elias Souza Junior, Marcio Marcelo Piffer, Carlos Parracho, Miriam Barbosa, Silvia Lanzarin, Denis Nazareno Hauffe.

Ao Professor Luís Carlos Zancanella pelo apoio mesmo antes de iniciar o curso.

E finalmente, à todos os amigos externos ao curso que adquiri antes e durante o curso.

(Luciana Schmitz)

Sumário

Lista de Figuras	x
Lista de Símbolos	xii
Resumo	xvi
Abstract	xvii
1 Introdução	1
1.1 Objetivos	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
1.2 Materiais e Métodos	2
1.3 Motivação e Justificativa	3
1.4 Trabalhos Correlacionados	4
1.4.1 Sistema de Webmail Seguro na Rede Computacional do CBPF . .	4
1.5 Organização do Texto	5
2 Correio Eletrônico	6
2.1 Introdução	6
2.2 Histórico	6
2.2.1 Definição	7
2.3 Vantagens	7
2.4 Sistema de Email	8

2.4.1	Webmail	10
2.5	Conclusão	10
3	Criptografia	11
3.1	Introdução	11
3.2	Definição	12
3.3	Criptografia	12
3.3.1	Criptografia por Chave Simétrica	13
3.3.2	Criptografia por Chave Assimétrica	14
3.3.3	Vantagens e Desvantagens	16
3.4	Certificado Digital	17
3.5	Conclusão	20
4	SMIME	21
4.1	Introdução	21
4.2	Histórico	21
4.2.1	RFC 822	21
4.2.2	MIME	24
4.2.3	S/MIME	26
4.3	Conclusão	27
5	Tecnologias Utilizadas	29
5.1	Introdução	29
5.2	Sistemas Operacionais	29
5.2.1	Fedora Core 1	29
5.2.2	OpenBSD	30
5.3	Servidor Web	30
5.4	PHP	31
5.5	SSL	31
5.6	MySQL	32
5.7	Horde	32

5.7.1	IMP	33
5.8	CVS-Control Version System	33
6	Implantação do Webmail Seguro	35
6.1	Introdução	35
6.2	Sistema Webmail	35
6.2.1	Topologia do Webmail	36
6.2.2	Manipulação de Mensagens	36
6.3	Características de Uso de um Webmail Seguro	39
6.4	Instalando o Webmail IMP com suporte a S/MIME	39
6.4.1	Um breve histórico	40
6.4.2	Configurando o Ambiente	41
6.4.3	Download do Horde, IMP e Framework	50
6.4.4	Instalar os pacotes do Horde	51
6.4.5	Configurar o Horde	52
6.4.6	Configurar o IMP	60
6.4.7	Importando as chaves S/MIME	64
6.4.8	Enviando um email assinado com S/MIME	68
6.4.9	Lendo uma mensagem assinada digitalmente pelo IMP	71
6.4.10	Atualizando a sua versão CVS do IMP	74
6.5	Problemas e Soluções	75
6.6	Conclusão	81
7	Considerações Finais	83
7.1	Trabalhos Futuros	84
	Referências Bibliográficas	85

Lista de Figuras

2.1	Mensagens	8
3.1	Chave Simétrica	14
3.2	Chave Assimétrica - Autenticação	15
3.3	Chave Assimétrica - Autenticação e Sigilo	16
3.4	Estrutura Certificado Digital	19
4.1	Envelope/Conteúdo da Mensagem	22
4.2	Estrutura da Mensagem	23
6.1	Lendo Mensagens	36
6.2	Enviando Mensagens	38
6.3	PHP no OpenBSD	48
6.4	Inicia Horde	55
6.5	Menu Administração	57
6.6	Menu Horde Completo	63
6.7	Menu Horde Configurado	64
6.8	Importando Chaves	66
6.9	Importando Chaves - Arquivos	66
6.10	Importando Chaves - Resultado	67
6.11	Compondo Email Assinado	68
6.12	Assinando Email	69
6.13	Passphrase	70
6.14	Mensagem Assinada	71

6.15 Mensagem Assinada - Aviso	72
6.16 Mensagem Assinada - Certificado	73
6.17 Resposta - Popup	79
6.18 Histórico	80
6.19 Código	81

Lista de Siglas

AC - Autoridade Certificadora

AR - Autoridade Regulamentadora

ASCII - Amsterdam Subversive Code for Information Interchange

BSD - Berkeley Software Distribution

CGI - Common Gateway Interface

CSP - Cryptographic Service Providers

CVS - Control Version System

DES - Data Encryption Standard

DLL - Dynamic-link library

DSA - Digital Signature Algorithm

DSS - Digital Signature Standard

FTP - File Transfer Protocol

HTML - Hypertext Markup Format

HTTP - Hypertext Transfer Protocol

HTTPD - Servidor Web Apache

HTTPS - Secure HyperText Transfer Protocol

ICP - Infra-estrutura de Chave Pública

IMAP - Internet Message Access Protocol

IMP - Internet Messaging Program

INE - Departamento de Informática e Estatística da Universidade Federal de Santa Catarina

IP - Internet Protocol

IPSEC - Internet Protocol Security

ITU-T - International Telecommunication Union - Telecommunication

LabSEC - Laboratório de Segurança em Computação

LCR - Lista de Certificados Revogados

MIME - Multipurpose Internet Mail Extensions

MTA - Mail Transfer Agent

PEM - Privacy Enhanced Mail

PKC - Public Key Certificate

PKCS - Public Key Cryptography Standards

PHP - Personal Hipertext Processor

POP3 - Post Office Protocol - Versão 3

RFC - Request for Comments

RC2-40 - Rivest Cipher - Versão 2 de 40 bits

RSA - Rivest-Shamir-Adelman

SET - Secure Eletronic Transaction

SMIME - Secure Multipurpose Internet Mail Extensions

SMTP - Simple Mail Transfer Protocol

SQL - Structured Query Language

SSL - Secure Socket Layer

TCP - Transmission Control Protocol

Resumo

Este projeto refere-se ao Trabalho de Conclusão de Curso sobre a implantação de um sistema de Webmail Seguro.

Inicialmente são apresentados alguns conceitos para melhor compreensão do funcionamento de um Webmail Seguro, bem como da sua necessidade para os dias de hoje. Alguns desses conceitos são as diferenças entre Webmails e Clientes de email tradicionais; criptografia de Chave Pública e Privada e suas utilizações.

Além disso, falamos também um pouco das tecnologias que serão utilizadas para a implantação deste projeto, como por exemplo: Webmail/IMP, que foi escolhido por ser de código aberto; servidor Apache, escolhido pelo mesmo motivo; o protocolo S/MIME, que estabelece as regras de segurança de uma mensagem de e-mail e outros.

Ao final é apresentada a proposta de implantação deste projeto.

Abstract

This work intends to introduce the proposal of the Course Conclusion Work about Secure Webmail.

Initially some concepts are presented for better comprehension about how Secure Webmail works, as well as its need nowadays. Some of these concepts are what is a Webmail, a mail client and the differences between them; Public and Private Key cryptography and their applications.

Beside this, we talk about the technologies that will be used in the development of this project, for example: Webmail/IMP, which was chosen by the fact that is an open source webmail; Apache server, chosen by the same reason; the S/MIME protocol, which establish the security rules for an email message; and many others.

At the end is presented the implementation proposal for this project.

Capítulo 1

Introdução

Existem muitas aplicações que são utilizadas frequentemente na Internet. Dentre estas destaca-se o correio eletrônico.

Enviar mensagens para outros usuários é um dos recursos mais utilizados na Internet. Este recurso substitui por muitas vezes o tradicional correio que as pessoas utilizam para enviar cartas, cartões de aniversários, cartões postais, etc para outras pessoas. Além de não apresentar custos, não é preciso colocar selos como no correio tradicional. É mais rápido e chega em outro país com extrema facilidade, bastando para isso que o programa de correio eletrônico esteja devidamente configurado e o endereço eletrônico da pessoa ou empresa esteja completo e correto.

Um endereço eletrônico é dividido em partes, como por exemplo **nomedousuario@provedor.tipo.país**, ou seja um determinado usuário, hospedado em um determinado provedor, de um tipo de email (governo, comercial, etc) e que é do Brasil, no caso de endereços eletrônicos brasileiros. Normalmente os endereços eletrônicos de instituições como Universidades não tem o tipo do email.

Uma das maiores preocupações na troca de mensagens usando o correio eletrônico é com relação à segurança. Neste sentido foi proposto o protocolo S/MIME (Secure Multipurpose Internet Mail Extension) para complementar as funcionalidades do MIME (Multipurpose Internet Mail Extension) nos sistemas de correio eletrônico tradicionais. O S/MIME permite agregar confidencialidade e autenticidade às mensagens através do uso

da criptografia.

O S/MIME tem sido implementado em clientes de correio eletrônico padrão, ou seja em aplicativos para utilização de endereço eletrônico como Outlook Express, Eudora, Internet Mail, etc. E mais recentemente em clientes de Webmail.

Este trabalho trata do estudo e implantação de um webmail seguro, que consiste de um sistema de webmail utilizando o a segurança SSL mais a segurança fornecida pelo protocolo S/MIME nas mensagens eletrônicas, com o objetivo de mostrar a utilização da segurança nas mensagens de correio eletrônico às pessoas interessadas em autenticação e sigilo na troca de mensagens eletrônicas via webmail.

1.1 Objetivos

1.1.1 Objetivo Geral

Mostrar a utilização de um sistema de Webmail Seguro, bem como suas vantagens, em relação a troca de mensagens eletrônicas, possibilitando ao usuário cifrar e/ou assinar digitalmente suas mensagens de correio eletrônico fazendo uso do protocolo S/MIME, como já acontece com clientes de email como Outlook Express, Eudora e outros.

1.1.2 Objetivos Específicos

- Verificar a viabilidade de implantação de um sistema de Webmail com suporte ao protocolo S/MIME;
- Mostrar a utilização dos serviços criptográficos, principalmente em relação ao armazenamento e gerência das chaves privadas;

1.2 Materiais e Métodos

As ferramentas e tecnologias utilizadas foram:

- Sistema Operacional, foi utilizado a distribuição Linux da Red Hat, o Fedora Core 1 em uma máquina local e o OpenBSD em uma máquina virtual;
- Servidor Web, utilizamos o Projeto Apache (HTTPD);
- Webmail do IMP (*Internet Messaging Program*), o qual foi escolhido por ter o código aberto. É desenvolvido em PHP baseado no protocolo IMAP, que assume uma conta de email se o servidor suporta IMAP e, após instalado pode usar o IMP para verificar os emails de qualquer lugar que tenha acesso a Internet [IMP];
- O protocolo S/MIME, que será o protocolo usado para definir a questão de segurança nas mensagens de correio eletrônico, junto aos certificados X.509;
- SSL, fornece um canal seguro entre as duas partes envolvidas, o cliente e o servidor.
- MySQL, a base de dados de armazenamento das informações usada pelo IMP.

1.3 Motivação e Justificativa

A importância de um Webmail Seguro consiste no sigilo e autenticidade de mensagens eletrônicas trocadas via web pelas pessoas através do uso do S/MIME. Hoje esta tecnologia já é utilizada em clientes de email como o Outlook Express e outros, mas em Webmail existem poucos sistemas, sendo o sistema IMP um dos pioneiros.

Para mostrar a importância dessa tecnologia em sistemas de Webmail, vamos imaginar algo hipotético como um Detetive Secreto Digital - DSD, que está investigando sobre um Bandido Digital - BD (um Super Hacker, por exemplo) em outro país, e quer enviar mensagens eletrônicas ao seu Departamento de Polícia - DP, sendo que não existe outra forma segura e rápida de se comunicar.

Como um DSD, este sabe que o BD pode interceptar suas mensagens eletrônicas via web, mas seu DP utiliza Servidor Seguro com um Webmail Seguro, que garante:

- o sigilo de sua mensagem, ou seja mesmo que o BD consiga interceptá-la, não conseguirá decifrá-la nem alterá-la.

- a autenticidade do DSD, ou seja o DP terá certeza que a mensagem estará vindo do DSD.

Além da segurança, o DSD utilizando um Webmail não precisará instalar e/ou configurar software algum onde estiver, garantindo sua eficiência, já que este é um DSD que necessita de rapidez em suas atividades.

Portanto um Webmail Seguro ajudará em muito as pessoas que necessitam de autenticidade e sigilo em suas mensagens, não só neste exemplo hipotético, mas em muitos casos onde necessita-se desse tipo de segurança.

Sabemos que hoje, qualquer pessoa pode se fazer passar por outra e enviar uma mensagem eletrônica utilizando um endereço de email qualquer, e que um hacker pode facilmente capturar uma mensagem que estiver transitando pela web.

Dessa forma o Webmail Seguro vem para suprir essas necessidades da sociedade .

1.4 Trabalhos Correlacionados

1.4.1 Sistema de Webmail Seguro na Rede Computacional do CBPF

Foi implantado um sistema de Webmail Seguro na Rede computacional do CBPF (Centro Brasileiro de Pesquisas Físicas), que acrescentou uma maior mobilidade para o usuário com a instalação de uma interface, sistema de webmail IMP, para transmissão e recepção de mensagens capaz de ser acessada de qualquer computador cliente da Internet. Garantindo não só apenas a praticidade, mas também a segurança e privacidade das mensagens através do "túnel seguro". Uma das formas utilizadas para isto é o serviço de Webmail acrescido de cifragem de dados. Este sistema de Webmail Seguro utiliza um túnel seguro para a transferência de mensagens entre usuários de um mesmo domínio. Este sistema não oferecia, na época de sua implantação, um serviço de assinatura digital de mensagens de correio eletrônico. [CN]

1.5 Organização do Texto

No capítulo 2 - Correio Eletrônico, temos um histórico do Correio Eletrônico, o que é, como e quando surgiu, e outras informações.

No capítulo 3 - Criptografia, abordamos alguns conceitos importantes para entender o que é, para que serve e como são implementadas algumas técnicas de criptografia utilizadas atualmente.

No capítulo 4 - S/MIME, falamos sobre o surgimento do protocolo S/MIME, desde a RFC 822 que define o formato de escrita de um email, passando pelo protocolo MIME e posteriormente pelo protocolo S/MIME, que acrescenta tecnologias de segurança.

No capítulo 5 - Tecnologias Utilizadas, falamos sobre cada tecnologia que foi utilizada na implantação deste projeto.

No capítulo 6 - Implantação do Webmail Seguro, falamos sobre como foi feita a implantação, os requisitos necessários, o tempo utilizado, a fase de instalação e de testes.

No capítulo 7 - Considerações Finais, são as conclusões finais deste projeto.

Capítulo 2

Correio Eletrônico

2.1 Introdução

Correio Eletrônico ou Email significa "*eletronic mail*". É uma forma de enviar mensagens de uma pessoa para outra através de um meio de comunicação eletrônico.

Inicialmente, o email era a maneira como diferentes usuários de um servidor de grande porte se comunicavam. Porém surgiu a necessidade das mensagens irem para servidores diferentes. Como estes servidores não estavam todos interligados, os emails eram encaminhados de um servidor para outro colocando no endereço de email o caminho completo que a mensagem deveria percorrer até chegar em seu destinatário. Hoje em dia, quase todos os emails são enviados diretamente via servidores conectados na Internet.

Mesmo com a popularização da Web, o email ainda é a forma mais popular de acesso à rede.

2.2 Histórico

Em 1971, Ray Tomlinson desenvolveu pela primeira vez um programa capaz de enviar pequenas mensagens eletrônicas. Utilizou um protocolo de transferência de arquivos para poder mandar mensagens para toda a rede. Na primeira mensagem enviada leu-se simplesmente "QWERTYUIOP". Como Ray estava no período de trabalho, ele

não quis dizer que foi ele quem havia mandado a mensagem.

Mais tarde em 1997, Baggott e Nichol, perceberam que as empresas tinham a necessidade de se comunicar e enviar mensagens eletrônicas entre si, criando assim uma conexão global.

2.2.1 Definição

O email é a forma de comunicação mais utilizada na Internet. São mensagens de correspondência geradas por um usuário em um computador que são enviadas sobre uma rede de computadores para um outro usuário em outro computador.

Para receber e enviar mensagens o usuário precisa de um endereço eletrônico, que o identificará na rede Internet, bem como uma senha de acesso que lhe garantirá a recuperação das mensagens destinadas a ele. A forma de endereçamento Internet usada nas correspondências eletrônicas é *nomedousuario@provedor.tipo.país*[WIR], onde cada parte significa:

- *nomedousuário*, é o nome do usuario ou login.
- @, é símbolo que se chama arroba e que significa "em". (em inglês "at")
- *provedor*, é o nome da empresa onde o usuário tem sua conta de email
- *tipo*, é um tipo de endereço eletrônico. ("com"= comercial, "gov"= governo)
- *país*, é o país ("br"= Brasil, "ar"= Argentina, por exemplo)

Supondo o seguinte endereço eletrônico: *ronaldinho@craque.com.br*. Podemos dizer que este endereço pertence ao usuário com login *ronaldinho* junto ao provedor *craque* que é do tipo comercial e fica situado no Brasil.

2.3 Vantagens

As mensagens eletrônicas funcionam da seguinte forma. Um usuário A envia uma mensagem para um usuário B, esta mensagem é armazenada em um servidor de email,

que é um computador com grande capacidade de armazenamento, até que o usuário B acesse a sua caixa postal, local onde ficam as mensagens recebidas. Cada caixa postal está associada a um único endereço, que pode pertencer a um único usuário (caixa postal privada) ou pertencer a um grupo de pessoas (caixa postal de grupo)[RSO], conforme ilustra a Figura 2.1.

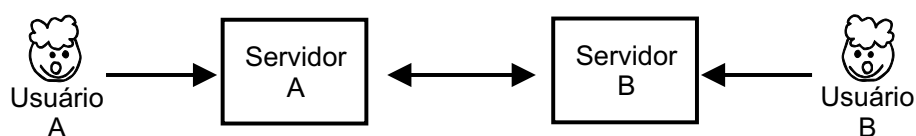


Figura 2.1: Mensagens. Funcionamento das mensagens eletrônicas trocadas entre usuários

Enviar mensagens eletrônicas tornou-se um serviço com muitas vantagens em relação ao tradicional correio. Na sequência, algumas vantagens são listadas:

- O correio eletrônico é virtualmente instantâneo, e não apresenta custos;
- Por ser um serviço de comunicação assíncrono, o usuário não precisa estar conectado a Internet no momento em que uma mensagem lhe for enviada;
- Enquanto uma carta convencional pode levar dias para chegar ao seu destino, a mesma mensagem pode chegar em alguns minutos no outro lado do mundo utilizando email;
- Mídia ecológica, as mensagens são digitais e, portanto, evita o uso de papel.

2.4 Sistema de Email

Podemos distinguir três componentes principais no sistema de correio eletrônico da Internet: cliente de email, o servidor de email e o protocolo SMTP.

O **Cliente de email** permite ao usuário ler, responder, encaminhar, salvar e compôr mensagens. No final de 1990, clientes de email com interface gráfica se tornaram populares, o que permitiu a adição de outros componentes ao email além de texto como imagens, audio e vídeo. Atualmente temos como exemplo de cliente de email com interface gráfica

o Eudora (www.qualcomm.com), o Microsoft Outlook (www.microsoft.com), o Netscape Messenger (www.netscape.com) e outros. Os baseado em interface texto temos o mutt, pine, mail, elm, etc.

Os **Servidores de email** formam a principal parte da infra-estrutura de correio eletrônico. Cada usuário que recebe correspondência tem uma caixa de entrada, local onde ficam armazenadas as mensagens recebidas, localizada em um servidor de email que gerencia estas mensagens. O caminho percorrido por uma mensagem começa a partir do cliente de email. Ela é enviada até o servidor de email de quem a enviou para então ser encaminhada para o servidor de email do destinatário. Lá a mensagem fica armazenada na caixa de entrada do destinatário para que ele possa acessá-la posteriormente. O servidor de email também é responsável por avisar a quem enviou a mensagem se não foi possível entregá-la ao seu destinatário.

O protocolo de envio de mensagens é o **SMTP** (Simple Mail Transfer Protocol), é o principal protocolo para email na Internet, está definido na RFC 821, é ele o responsável por transferir o email do servidor de quem envia para o servidor de email de quem recebe. O servidor ao receber a mensagem a deixa guardada na caixa postal do usuário. O usuário pode obter suas mensagens de duas formas dependendo das características definidas:

- Via POP (Post Office Protocol) que já se encontra na versão 3, POP3. Uma vez obtidas as mensagens estas são apagadas completamente do servidor.
- Via IMAP (Interactive Mail Access Protocol) onde podemos consultar no nosso computador uma cópia das mensagens recebidas.

Os dois protocolos para receber mensagens eletrônicas podem ser configurados em sistemas como os clientes de email desde que isso seja configurado no servidor de email, já os sistemas de webmail, que são instalados e configurados no servidor, utilizam o protocolo IMAP para recebimento das mensagens.

2.4.1 Webmail

Com o crescimento da Web foram surgindo novos recursos de email, um deles é o Webmail, que nada mais é que um serviço de email utilizado pelo Browser não mais necessitando de um cliente de email local [ICI], ou seja, utilizando um cliente de email local, como os citados acima, caso o usuário queira rever uma mensagem já baixada do seu servidor, terá que ir na máquina que estava utilizando naquele momento. Já com o Webmail o usuário pode rever suas mensagens utilizando qualquer máquina, pois as mensagens ficam em um servidor.

Outra vantagem de se utilizar um sistema de Webmail é que o usuário não precisa instalar e/ou configurar um cliente de email, basta fazer um cadastro no site que está oferecendo o serviço e pronto, é só acessar sua caixa postal de qualquer computador conectado a internet.

2.5 Conclusão

Neste capítulo conceituamos o que é email e seu princípio de funcionamento. Mostramos também as principais diferenças entre os clientes de email e serviços de web-mails.

O Webmail está se tornando cada vez mais popular. Empresas estão começando a ver as vantagens da utilização de um sistema de Webmail, facilitando a vida de seus usuários. Existe hoje uma grande quantidade de serviços de webmail oferecidos até gratuitamente por diversos sites como por exemplo: o Zipmail (www.zipmail.com.br), Bol (www.bol.com.br), Globo.com (www.globo.com), Hotmail (www.hotmail.com.br) e outros tantos.

Nossa trabalho visa a implantação de um sistema de webmail que permitirá ao usuário cifrar e assinar digitalmente suas mensagens de correio eletrônico.

Capítulo 3

Criptografia

3.1 Introdução

A internet foi projetada com o objetivo inicial de permitir uma grande interoperabilidade entre os sistemas a ela conectados. O foco era a conectividade e não a segurança. Isto criou um problema que se arrasta até os dias de hoje, que é a garantia de troca de informações de forma segura através da internet.

Como as informações que circulam na internet hoje em dia variam muito quanto ao teor e importância, e várias transações dependem destas informações para serem efetuadas, tornou-se necessária a criação de mecanismos que garantam a segurança dos dados sigilosos que transitam pela rede mundial de computadores.

Em um serviço de *Internet Banking*, por exemplo, quando acessamos o site do banco é estabelecida uma conexão segura por onde os dados que sejam trocados através deste site estejam protegidos até que a conexão seja encerrada. O protocolo SSL (*Secure Socket Layer*) é um exemplo deste tipo de conexão. Ele cria um túnel por onde passam as informações de forma segura. Já num serviço de Webmail o estabelecimento de uma conexão segura garante a privacidade somente para a conexão entre o usuário e o servidor, mas não para o envio de mensagens. Desta forma é preciso proteger a mensagem.

A criptografia pode ser definida inicialmente como uma técnica que transforma uma informação inteligível, em algo ilegível e completamente sem sentido para pessoas

não autorizadas, garantindo assim a integridade da informação. Garantir a privacidade na comunicação tem sido a ênfase da criptografia durante toda sua história. No entanto, isto é apenas uma parte do que é a criptografia hoje em dia. A autenticação, a integridade e o não-repúdio são conceitos que, juntamente com a privacidade, formam os pilares de uma comunicação segura.

Ainda neste capítulo é apresentado o conceito e estrutura de um Certificado Digital.

3.2 Definição

A palavra criptografia vem das palavras gregas *kriptos* (escondido) e *grifo* (grafia). Ela define o ato de se escrever em cifras ou códigos, através dos processos de cifragem transformando um texto em algo incompreensível de forma que apenas a pessoa desejada possa ler a mensagem através do processo inverso, a decifragem.

3.3 Criptografia

Existem duas formas básicas de se codificar uma mensagem: através de cifras ou através de códigos. A codificação consiste em ocultar uma informação através de códigos predefinidos entre o emissor e o receptor da mensagem. É talvez a forma mais antiga de ocultamento de informações. A comunicação através de sinais de fumaça pode ser usada como um exemplo de codificação. Por exemplo, dois grupos de escoteiros podem combinar entre si que fumaça branca signifique que está tudo em ordem no acampamento, enquanto que fumaça preta signifique que algo está errado. Quem não conheça este acordo entre os acampamentos irá ver apenas sinais de fumaça isolados sem imaginar que eles possam ter algum significado real. Apesar de ser um exemplo simplista, ele serve para mostrar a principal desvantagem da criptografia através de códigos, que é a troca apenas de informações previamente conhecidas. Caso um acampamento queira convidar o outro para uma confraternização, não haverá sinais de fumaça que transmitam esta informação para o segundo acampamento.

A criptografia através de cifras consiste em cifrar uma mensagem trocando ou misturando partes do conteúdo original da mensagem. Quando chegar ao destinatário a mensagem é decifrada fazendo-se o processo inverso da cifragem, resgatando a mensagem original.

O processo de cifragem, seja por troca ou mistura das partes do texto, envolvem o conhecimento da chave de cifragem por ambas as partes. Tomando como exemplo um processo de cifragem por deslocamento, podemos implementar um algoritmo que desloque cada letra da mensagem 3 posições a sua frente no alfabeto. Assim teríamos que o "a" vale "d", o "b" vale "e" e assim por diante. Como o que este algoritmo implementa é o deslocamento de N letras à frente, a chave deste algoritmo é N, e N neste caso vale 3. Do ponto de vista de um usuário comum, as chaves de criptografia são equivalentes às senhas de acesso a serviços como bancos, computadores, etc. Se a pessoa possui a chave certa terá acesso aos serviços, no caso da criptografia terá acesso à informação decifrada.

A segurança de uma chave está relacionada com sua extensão, ou seja o tamanho da chave. Quanto maior for o tamanho da chave maior será seu grau de segurança e, conseqüentemente, maior será o grau de confidencialidade da mensagem[RED]. Para usos como certificações, assinaturas digitais e confirmação de identidade, estão sendo usadas chaves com tamanhos que variam de 40 à 2048 bits.

3.3.1 Criptografia por Chave Simétrica

Quando podemos usar uma única chave para cifrar e decifrar uma mensagem, dizemos que este sistema usa a criptografia por chave simétrica ou chave secreta. Neste sistema tanto o emissor quanto o receptor compartilham a mesma chave e esta deve ser mantida em segredo.

Se uma pessoa deseja se comunicar com outra com segurança, esta deve passar a chave utilizada para cifrar a mensagem. Como esta chave é a parte fundamental da segurança, ela deve ser transmitida através de um meio seguro. Este meio, devido ao seu custo e escassez de tempo para sua utilização, deve ser usado uma única vez e não continuamente, o que justifica a transmissão apenas de uma chave uma única vez entre

duas pessoas em vez de várias trocas de mensagens através deste meio seguro.

Um exemplo, pode ser visto na figura a seguir, onde Alice deseja enviar uma mensagem cifrada para Beto, sendo que Alice já enviou, de forma segura, sua chave para Beto poder decifrar a mensagem. Dessa forma, Alice ao escrever a mensagem, vai utilizar a chave compartilhada por ambos, cifrar a mensagem e enviar. Beto, por sua vez, recebe a mensagem cifrada e utiliza a chave compartilhada com Alice para decifrar a mensagem e poder ler a mensagem recebida.

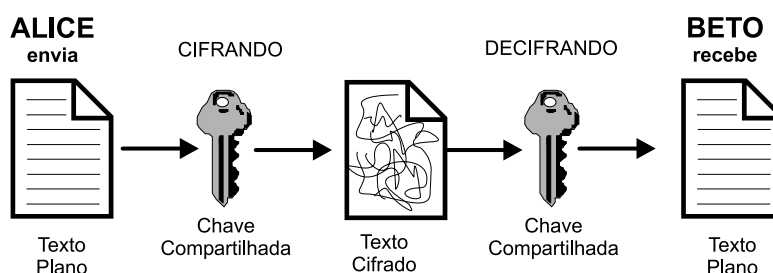


Figura 3.1: Chave Simétrica. Uma única chave é utilizada para cifrar e decifrar a mensagem.

Requerer um meio de transmissão seguro para a passagem de chaves entre as partes sem que ninguém possa tomar conhecimento delas é o principal problema da criptografia por chave simétrica. Contudo, ela tem como vantagem o fato de ser geralmente mais rápida que a criptografia por chave assimétrica ou chave pública.

3.3.2 Criptografia por Chave Assimétrica

Como forma de resolver o problema de gerenciamento de chaves e de ter que se confiar em um terceiro membro na comunicação da chave na criptografia de chave secreta, foi introduzido o conceito de criptografia de chave pública. Este conceito tem dois objetivos primários: privacidade e assinatura digital. Neste sistema cada usuário possui duas chaves, uma pública e outra privada. Estas chaves devem ser solicitadas junto a uma Autoridade Certificadora. A chave pública é divulgada e a chave privada é mantida em segredo junto ao usuário. A necessidade de compartilhamento de informações sigilosas entre emissor e receptor é eliminada, e não é mais preciso se preocupar com a segurança do meio de comunicação entre eles.

As figuras a seguir mostram como funciona o processo de somente autenticação e o processo de autenticação e sigilo de uma mensagem, respectivamente, onde "P" significa Chave Privada de Alice ou Beto e "U" significa Chave Pública de Alice ou Beto.

Um exemplo de autenticação utilizando chave pública seria: Alice deseja enviar uma mensagem a Beto, de forma que Beto tenha absoluta certeza que foi Alice quem enviou a mensagem. Neste caso Alice irá assinar digitalmente a mensagem, assim ela compõe a mensagem, utiliza sua chave privada para assinar e envia. Beto ao receber a mensagem, utiliza a chave pública de Alice, já antes divulgada por ela, para verificar a assinatura digital de Alice e ler sua mensagem, dessa forma Beto tem certeza de que quem enviou a mensagem foi Alice.

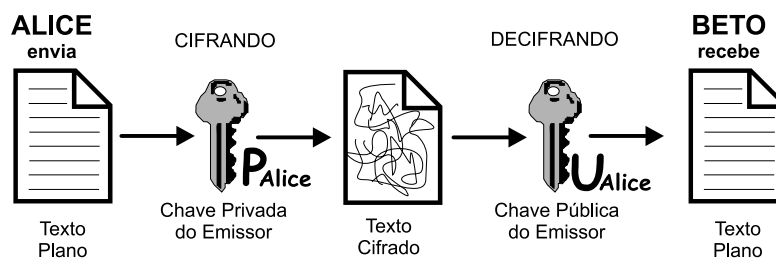


Figura 3.2: Chave Assimétrica - Autenticação. O Emissor utiliza somente sua chave privada. O receptor não necessita utilizar suas chaves.

Outro exemplo seria, além da autenticação, o sigilo da mensagem. Assim, Alice agora deseja enviar a Beto uma mensagem assinada e quer garantir que somente Beto irá ler a mensagem. Para isso, Alice compõe a mensagem, utiliza sua chave privada para assinar digitalmente, utiliza a chave pública de Beto (já antes divulgada por ele) para garantir o sigilo e envia a mensagem. Beto, recebendo a mensagem, utiliza a chave pública de Alice para ter a certeza de que foi ela quem enviou, e posteriormente utiliza sua chave privada para decifrar a mensagem e poder ler.

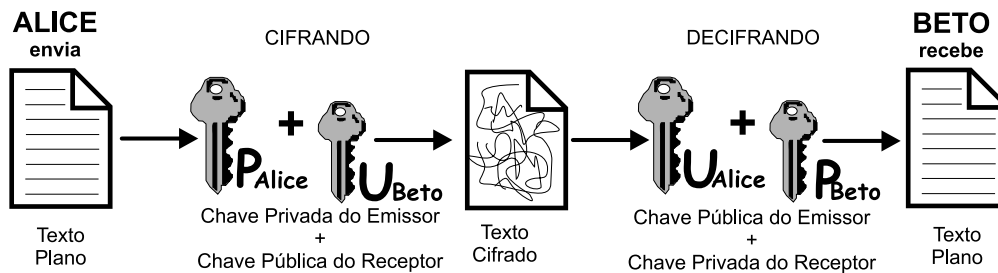


Figura 3.3: Chave Assimétrica - Autenticação e Sigilo. O Emissor utiliza sua chave privada e a chave pública do Receptor. O Receptor necessita utilizar sua chave privada e a pública do Emissor.

A chave pública é usada para cifrar a mensagem e geralmente ela é divulgada pelo usuário, de forma que qualquer pessoa possa lhe enviar mensagens cifradas. Já a chave privada é de conhecimento apenas de seu possuidor, e deve ser mantida em sigilo, pois apenas ela é capaz de decifrar uma mensagem enviada para um destinatário utilizando sua chave pública.

A chave pública é ligada matematicamente à chave privada, de maneira que sempre é possível obter a chave privada a partir da chave pública. O que é feito para impedir que isto aconteça é tornar esta obtenção computacionalmente impossível de se efetivar devido ao enorme volume de cálculos envolvidos no processo [LAB a] .

3.3.3 Vantagens e Desvantagens

A principal vantagem da criptografia de chave pública sobre a de chave secreta é o fato de não ser preciso compartilhar a chave privada com ninguém. A partir do momento que a criptografia por chave secreta exige que esta chave seja transmitida para outra pessoa, abre-se uma brecha para que esta chave caia em mãos erradas.

Outra vantagem do sistema com chave pública é a possibilidade de prover assinaturas digitais que não podem ser repudiadas. Esta propriedade da autenticação por chave pública é chamada de não-repúdio. Uma autenticação por chave secreta requer o compartilhamento de informações secretas e também a confiança em uma terceira parte no processo de autenticação, já que as chaves secretas de cada usuário são guardadas em uma central que mantém cópias destas chaves. Um ataque nesta base de dados e ninguém

poderá saber se quem está enviando uma mensagem é realmente quem diz ser. Já no sistema de chave pública isto não acontece, pois a chave privada é de uso e conhecimento apenas de seu possuidor. Desta forma a autenticação funciona como uma espécie de recibo criptográfico, de maneira que o autor de uma mensagem não possa negar falsamente que a tenha enviado.

A desvantagem do sistema de chave pública em relação ao de chave secreta está na velocidade. Existem muitos métodos de criptografia por chave secreta que são mais rápidos que qualquer método de criptografia por chave pública existente hoje em dia.

Existem situações em que a criptografia por chave pública não é necessária. Geralmente são ambientes onde a distribuição de chaves secretas não é um problema, seja devido as partes poderem encontrar-se a sós ou seja pela presença de uma autoridade que centralize e gerencie todas as chaves. Em um sistema bancário, por exemplo, o próprio banco tem a capacidade de gerenciar e centralizar de forma segura as chaves secretas (senhas) de seus clientes, e já que a autoridade tem conhecimento de todas as chaves não há necessidade de umas tornarem-se públicas e outras privadas. Geralmente a criptografia de chave pública tem mais utilidade em ambientes abertos e multi-usuários.

3.4 Certificado Digital

ICP é a sigla para Infra-estrutura de Chaves Públicas no Brasil, que é quem determina as técnicas e procedimentos para utilizar e suportar um sistema criptográfico baseado em certificados digitais. Os certificados digitais, se responsabilizam por tornar possível as transações e comunicações sem que haja riscos entre as partes envolvidas. Portanto, a ICP-Brasil tem por função coordenar e estabelecer política para as normas para o licenciamento de Autoridades Certificadores (AC), Autoridades de Registro (AR), e outros serviços de suporte em todas as camadas da cadeia de certificação.

As ACs, tem por função emitir certificados digitais, vinculando os pares de chaves criptográficas ao titular, expedir, distribuir, revogar e gerenciar os certificados, além de disponibilizar aos seus usuários uma Listas de Certificados Revogados (LCR). Deve também manter o registro de suas operações.

As ARs, são entidades que estão vinculadas as ACs, tem por função identificar e cadastrar usuários na presença dos mesmos, encaminhar solicitações de certificados às ACs e manter registros de suas operações. [CER]

Em 1978 foi proposto a criação de uma estrutura de dados assinada contendo uma identificação e a chave pública que hoje é chamado de Certificado Digital de Chave Pública ou Public Key Certificate (PKC).

Os Certificados Digitais são como um identificador digital semelhante a uma carteira de motorista ou passaporte onde você pode apresentar para se identificar digitalmente ou acessar informações e serviços on-line.

Um exemplo de utilização de certificados digitais pode ser visto no site do Banco do Brasil (www.bb.com.br), onde o certificado digital é utilizado para comprovar eletronicamente a identidade do usuário. Digamos que o usuário queira acessar o site e fazer suas transações eletrônicas como transferência de dinheiro, pagamento, verificação de saldo, etc. Ao acessar o site através de seus dados de usuário e senha, o Banco precisa certificar-se da melhor forma de que o usuário é quem diz ser, fazendo uma verificação de seus dados com o certificado, garantindo assim a autenticidade do usuário e de que não houve alteração do conteúdo da mensagem entre o momento de emissão e o de recebimento da mensagem.

Os Certificados Digitais unem uma identidade a um par de chaves eletrônicas, que podem ser usadas para cifrar e assinar informações digitais.

É emitido por uma AC(Autoridade Certificadora) e assinada com a chave privada da AC. Normalmente é composto por:

- Chave Privada do dono;
- Nome do dono;
- Data de validade da chave privada;
- Nome do emissor (a autoridade certificadora que emitiu o certificado);
- Número de série do Certificado;

- Assinatura digital do emissor;

Além dessas informações outras podem ser adicionadas para atender alguma exigência de uma determinada empresa.

O formato mais aceito dos Certificados Digitais é definido pelo padrão internacional ITU-T o padrão X.509, que define a estrutura do certificado e o protocolo de autenticação que é utilizado em diversas aplicações. Veja figura a seguir sobre estrutura do padrão X.509 [BUR].

Versão
Número de Série do Certificado
Identificador do Algoritmo de Assinatura
Nome do emissor(AC)
Periodo de Validade
Nome do Sujeito
Informações da Chave Pública do Sujeito
Identificador Unico do Emissor(AC)
Identificador Unico do Sujeito
Extensões
Assintaura

Figura 3.4: Estrutura Certificado Digital. Estrutura do padrão X.509 de Certificado Digital.

O certificado X.509 é utilizado pelo S/MIME, IPSEC, SSL e SET, mas podem ser lidos e escritos por qualquer aplicativo compatível com o X.509 como os padrões PKCS.

Neste trabalho é citado dois padrões de PKCS o PKCS#7 e PKCS#12, que servem respectivamente para geração de envelopes digitais (este será melhor explicado no capítulo sobre S/MIME) e repositório de chaves e certificados em arquivos[RSA].

A extensão de importação ou exportação de um certificado digital varia conforme o sistema operacional no qual o usuário está trabalhando. Para o sistema operacional Windows é utilizado o formato PKCS#12 que gera a extensão .pfx. E no sistema operacional

Linux ou Unix é utilizado o formato PKCS#7 que gera a extensão .pem.

O certificado utilizado neste trabalho foi gerado na Autoridade Certificadora do LabSEC, neste site[LAB b] você encontra um manual para utilização do certificado digital, ensinando como verificar e conhecer seu certificado, criar e recuperar cópia de segurança e proteger sua chave privada.

3.5 Conclusão

A criptografia de chave pública não veio em substituição à de chave secreta, mas sim para complementá-la e torná-la mais segura.

Neste capítulo também é falado como funciona os Certificados Digitais desde sua criação e estrutura até sua definição atual como formato padrão. Além disso foi mostrado como funciona uma ICP com suas responsabilidades, quais as responsabilidades de entidades como Autoridade Certificadora, Autoridade Regulamentadora e a Lista de Certificados Revogados.

Capítulo 4

SMIME

4.1 Introdução

Para que as pessoas passem a utilizar uma tecnologia de segurança nos emails como uma solução de interação dinâmica entre elas, é necessário definir como serão tratadas as atividades de autenticação e criptografia. Existem padrões estabelecidos para segurança do nível de transporte (SSL) e segurança de mensagens (S/MIME).

Este trabalho consiste em mostrar a utilização e importância da segurança nas mensagens. Portanto neste capítulo será mostrado a evolução e funcionalidade do padrão S/MIME.

4.2 Histórico

4.2.1 RFC 822

Na internet existem padrões de formatos para que os computadores possam conversar entre si. Esses formatos de padrões são desenvolvidos pela IETF (*Internet Engineering Task Force*) que escreve documentos para cada tipo de comunicação realizada na Internet. Esses documentos são chamados de RFC, que significa *Request for Comments*, sendo numeradas seqüencialmente.

A RFC que determina o formato padrão de mensagens de e-mail é a 822, escrita

por D. Crocker em Agosto de 1982. Ela foi a base do protocolo SMTP (*Simple Mail Transfer Protocol*), que foi desenvolvido para transportar as mensagens textuais pela Internet.

A estrutura da RFC 822 é muito simples. Uma mensagem consiste de um envelope e um conteúdo. No envelope estão as informações necessárias para se fazer com que o e-mail chegue ao destino. O conteúdo é o objeto a ser entregue ao destinatário do e-mail, conforme ilustra a Figura 4.1.

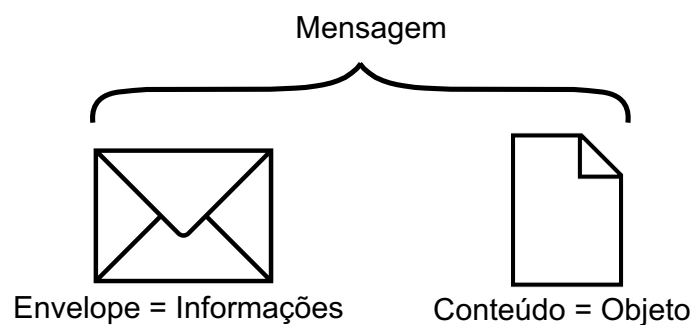


Figura 4.1: Envelope/Conteúdo da Mensagem. Idéia principal do formato de uma mensagem eletrônica.

A Figura 4.2 ilustra a estrutura de uma mensagem eletrônica.



Figura 4.2: Estrutura da Mensagem. Estrutura básica de uma mensagem eletrônica conforme a RFC 822.

A mensagem possui linhas de cabeçalho seguidas do corpo, que consiste de texto, sem restrições. Uma linha em branco separa o cabeçalho do corpo.

Uma linha de cabeçalho consiste de palavras-chave seguidas por dois pontos. As palavras-chave mais frequentes são:

- *Date: (Data)*
- *From: (De)*
- *To: (Para)*
- *Subject: (Assunto)*
- *Cc: (Cópia com)*
- *Cco: (Cópia oculta com)*

A seguir um exemplo de uma mensagem padronizada pela RFC 822.

```
Date: Tue, 16 Jan 2004 12:17:45 (EST)
From: "Luciana Schmitz"luciana@inf.ufsc.br
To: juchem@inf.ufsc.br
Subject: Sintaxe segundo RFC 822
```


Ola. Aqui inicia o corpo da mensagem, separado por uma linha em branco do cabeçalho.

Outro campo que faz parte do cabeçalho de mensagem especificado pela RFC 822, é o campo *Message-ID*, que contém o identificador único associado com a mensagem.

4.2.2 MIME

MIME significa *Multipurpose Internet Mail Extension*, ou seja Extensões Multipropósito do Internet Mail. É uma extensão da RFC 822, criado para solucionar os problemas de limitação do protocolo SMTP [STA].

Um exemplo de utilização do padrão MIME seria quando você envia um email contendo imagens e animações para outro usuário, e o cliente de email deste usuário destinatário aceita somente mensagens de email contendo apenas texto. Desta maneira a mensagem aparece como um texto sem formatação, e as imagens e animações aparecem em um arquivo hipertexto (HTML) anexado. Ou seja, o cliente de email do usuário não está nos padrões do MIME portanto só consegue ler a mensagem em formato texto, sendo assim somente programas de email que oferecem suporte ao MIME podem ler a formatação em HTML.

Algumas das limitações do protocolo SMTP/RFC822 são:

- Não pode transmitir arquivos executáveis ou objetos binários;
- Aceita apenas caracteres ASCII, que são limitados em 7-bits;
- Alguns servidores rejeitam mensagens muito grandes;
- Tem problemas com mapeamento em outras representações;
- Faz alterações indevidas nas mensagens.

MIME foi criado para resolver problemas desse tipo mantendo compatibilidade com as implementações existentes com a RFC 822. A especificação MIME provém das RFCs 2045 a 2049 e define novos elementos, como [CUS]:

- Cinco novos campos de cabeçalho de mensagem para prover informação sobre o corpo da mesma;
- Um outro número de formatos de conteúdo é definido, padronizando as representações de suporte a conteúdos multimídia;
- Codificações de transferências são definidos permitindo a transmissão de qualquer conteúdo sem perdas ou alterações dos dados e informações.

Os cinco novos campos de cabeçalhos de mensagem para prover informações sobre o corpo da mensagem são:

- *MIME-Version*: indica que a mensagem está conforme as RFCs 2045 e 2046;
- *Content-type*: descreve o conteúdo de maneira que o agente receptor possa mostrar de forma adequada ao usuário;
- *Content-Transfer-Encoding*: indica a transformação que foi utilizada no corpo da mensagem;
- *Content-ID*: identifica entidades MIME em contextos múltiplos. É um campo opcional;
- *Content-Description*: uma breve descrição do conteúdo da mensagem. Também é um campo opcional.

A seguir os sete tipos de conteúdo MIME especificado na RFC 2046 e seus quinze subtipos, onde os tipos são uma representação genérica dos dados e seus subtipos especificam um formato particular para os tipos de dados.

- Text: Plain e Enriched.

- Multipart: Mixed, Parallel, Alternative e Digest.
- Message: rfc822, Partial e External-body.
- Image: jpeg e gif.
- Video: mpeg.
- Audio: Basic.
- Application: PostScript e octet-stream.

As codificações para transferência são:

- 7-bit: os dados são todos representados por linhas curtas de caracteres ASCII.
- 8-bit: as linhas são curtas, mas podem conter caracteres não-ASCII.
- binary: as linhas podem ser grandes demais e dificultar o envio das informações via protocolo SMTP.
- quoted-printable: codifica de forma que a maioria dos caracteres se mantém legíveis.
- base64: codifica os dados mapeando blocos de 6-bits de entrada para blocos de 8-bits de saída, possibilitando abrir arquivos binários em modo texto.
- x-token: uma codificação qualquer não-padronizada

4.2.3 S/MIME

Em 1995 através de um grupo de empresas como a RSA, Microsoft, Lotus e Novel surgiu o *Secure Multipurpose Mail Extension S/MIME* que é definida pela IETF representada pela RFC 2311[IETF], com o intuito de garantir às mensagens eletrônicas enviadas pela Internet a:

- autenticidade: por meio dos certificados de chave pública X.509;

- confidencialidade: por meio do uso de algoritmos criptográficos especificados pelo usuário;
- integridade: pelo uso de funções criptográficas;
- não repúdio: pelas mensagens assinadas criptograficamente.

O S/MIME utiliza os certificados no padrão X.509 que dependem de uma autoridade certificadora. No S/MIME a confidencialidade das mensagens é garantida pelo uso de algoritmos de criptografia simétricos. No caso da integridade e a autenticidade das mensagens o S/MIME utiliza assinaturas digitais, baseados em algoritmos de chave pública como o RSA com chaves diferentes para cifrar e decifrar. Com o S/MIME é possível:

- cifrar uma mensagem, quando o objetivo é somente o sigilo ou confidencialidade;
- assinar uma mensagem, quando o objetivo é somente a autenticidade e a integridade;
- assinar e cifrar uma mensagem, quando os objetivos são confidencialidade, autenticidade e integridade.

4.3 Conclusão

Com o início da padronização de mensagens, feita pela RFC 822, não era possível enviar mensagens com arquivos anexos, sejam eles quais forem, e muito menos utilizar segurança digital. Com o desenvolvimento do S/MIME isto se tornou possível, trazendo a facilidade da troca de arquivos e principalmente a confidencialidade, autenticidade e integridade que as tecnologias de segurança fornecem.

Baseado no padrão MIME, o S/MIME oferece serviços criptográficos seguros para mensagens eletrônicas como: autenticidade, integridade e não-repúdio (utilizando assinatura digital) e privacidade de dados seguros (utilizando criptografia).

O S/MIME provê uma maneira consistente de enviar e receber dados seguros no padrão MIME. Por exemplo, ao tentarmos enviar uma mensagem de email com uma imagem anexada, ou algum arquivo de som, estaremos utilizando o padrão MIME, pois somente após seu surgimento que isto se tornou possível. No caso de quisermos aplicar segurança à esta mensagem, garantindo sua autenticidade, através da assinatura digital, e sigilo, através de algoritmos criptográficos, estaremos fazendo uso do padrão S/MIME, já que o S/MIME é o padrão que garante a segurança das mensagens de email.

Capítulo 5

Tecnologias Utilizadas

5.1 Introdução

Este capítulo destina-se a um estudo sobre as ferramentas e tecnologias que foram utilizadas para o desenvolvimento de nosso projeto, incluindo linguagens de programação; ferramentas e aplicativos; servidor Web e sistema de gerenciamento de banco de dados. Fizemos um estudo sobre cada uma das tecnologias e ferramentas que foram empregadas procurando dar uma síntese do que é e qual a finalidade da aplicação dessas ferramentas e tecnologias no projeto.

5.2 Sistemas Operacionais

5.2.1 Fedora Core 1

A Red Hat [RED], uma das principais empresas de distribuição Linux, após o lançamento da versão Red Hat 9 resolveu se dedicar ao mercado corporativo e por isso montou o Fedora Project, que é a continuação da versão 9 do Red Hat.

No Brasil há uma equipe de desenvolvedores do Fedora totalmente voltados ao público Brasileiro e muitas soluções desenvolvidas pela equipe não se aplicam à distribuição Internacional [BRA]. Todos os pacotes traduzidos e modificados pela equipe brasileira são submetidos aos servidores do Fedora Internacional para serem incluídas em futuras

versões de distribuição.

Foi utilizada a distribuição Fedora Core-1 do sistema operacional Linux. A escolha por esta distribuição está justificada no capítulo 7, onde será exposto os detalhes de implementação deste projeto.

5.2.2 OpenBSD

BSD significa Berkeley Software Distribution e surgiu com o objetivo de aprimorar e acrescentar funcionalidades ao sistema operacional Unix, virando um sistema operacional independente do Unix.

Tratando-se de segurança os sistemas operacionais BSDs são os melhores. Sua estrutura é totalmente desenvolvida considerando os aspectos de estabilidade, integridade, segurança e confiabilidade tornando-se um sistema robusto, eficiente e excelente para aplicações segurança.

Existem três sistemas operacionais mais importantes e conhecidos em BSD: o FreeBSD, OpenBSD e o NetBSD. Destes foi escolhido o OpenBSD por ser, dos três citados, o melhor no que diz respeito a segurança.

Já no processo de instalação o sistema desativa recursos que podem se tornar perigosos ao sistema. No desenvolvimento o código está em constante avaliação para que não haja erros de desenvolvimento e quando ocorre erros imediatamente é criado e aplicado uma solução.

Sua instalação é um pouco mais complexa e exige do usuário conhecimentos de segurança para poder configurar.

5.3 Servidor Web

Como servidor Web, foi utilizado o Apache. Funciona como um servidor de páginas web permitindo que outros computadores acessem remotamente um determinado site que está em uma pasta deste servidor. Pode ser utilizado para testar e hospedar sites, hospedar uma intranet ou um sistema de Webmail que é o caso deste projeto.

Segundo a HTTPD Project, o Apache oferece suporte a proxy HTTP, a sistemas de segurança SSL e a ferramentas de publicação web mais sofisticadas para a criação de páginas para internet. Além de rodar em diferentes sistemas operacionais como Windows, Unix, Linux, etc.

O Apache foi escolhido por ser um projeto de código aberto e por ter todas as funcionalidades necessárias à implantação deste projeto.

5.4 PHP

O PHP (Personal Hypertext Processor) é uma linguagem de script para servidor que facilita a criação de páginas Web dinâmicas, embutindo códigos PHP em documentos HTML. PHP combina muitas das melhores características de Perl, C e Java e adiciona seus próprios elementos à esta combinação para dar aos programadores Web grande flexibilidade e poder no desenho e implementação de páginas dinâmicas e orientadas a conteúdo para Web.

PHP é uma linguagem cuja interpretação é feita integralmente pelo servidor. Apenas a resposta é enviada para o cliente. Ao contrário de scripts CGI escritos em outras linguagens, que possuem uma grande quantidade de comandos com saídas para HTML, o código PHP é inserido dentro das tags HTML.

Estaremos lidando diretamente com PHP já que o Webmail que adotamos para desenvolver nosso trabalho é o Horde, um Webmail escrito em PHP.

5.5 SSL

O Secure Socket Layer (SSL), é o protocolo para criptografia e autenticação baseada em seção atualmente em uso na internet. Ele fornece um canal seguro entre as duas partes envolvidas, o cliente e o servidor. O SSL fornece uma autenticação de servidor e uma autenticação opcional de cliente para obstruir adulterações e falsificações de mensagens em aplicativos cliente-servidor. Um exemplo de aplicação que utiliza o protocolo SSL são os sistemas de *Internet-Banking*. Quando entramos no site do nosso banco, antes

de informarmos o número de nossa conta, agência e senha, somos direcionados para outro local do site onde é estabelecida uma conexão segura utilizando o protocolo SSL. Uma conexão "http"criptografada pelo SSL é indicada pelo prefixo "https:"

Este protocolo foi desenvolvido pela Netscape, e se tornou popular inicialmente por causa do navegador. Desde então o SSL já foi incorporado em vários outros navegadores de forma que hoje em dia não é mais uma vantagem competitiva e sim uma necessidade[OPE].

5.6 MySQL

A escolha pelo MySQL se deu por ser este o banco de dados mais utilizado em distribuições linux. Nele serão armazenados os dados de cada usuário do sistema de Web-mail, como por exemplo preferências de visualização de mensagens, filtros e as próprias chaves pública e privada, necessárias para a assinatura das mensagens de email.

5.7 Horde

Horde é ao mesmo tempo uma aplicação e um projeto. O Horde Project (Projeto Horde) reúne uma série de aplicações baseadas na web que vão desde gerenciamento de projetos e produtividade até aplicações para troca de mensagens eletrônicas. O Horde Framework (Framework do Horde) é uma base de código comum às aplicações do Projeto Horde. Ele é uma espécie de cola que mantém juntas todas as características em comum entre as aplicações do Projeto Horde. Dentre estas coisas em comum estão padrões de codificação, códigos em comum, e a inter-comunicação das aplicações. O código compartilhado proporciona maneiras em comum para a manipulação de aspectos como preferências, permissões, detecção do browser, dentre outros. Dentre os vários módulos que integram o Projeto Horde podemos citar:

- Horde: é o módulo que possui as bibliotecas do Horde e o Horde Framework;
- Framework: é o módulo que possui as bibliotecas do Horde Framework;

- IMP: é o servidor de Webmail do Projeto Horde;
- Chora: é o visualizador web de repositórios CVS do Horde;
- Turba: é o programa de catálogo de endereço e gerenciador de contatos do Horde. Ele surgiu da necessidade de um catálogo de endereços com mais funcionalidade do que o que vinha sendo incluído no IMP;
- Kronolith: é um calendário web robusto para usuários que possuem repetidos eventos. Possui um algoritmo que mostra um dia inteiro de eventos mesmo que alguns se sobreponham a outros.

De todos os servidores de Webmail livres pesquisados, nenhum apresentou tantas funcionalidades e avanços quanto o IMP. No decorrer deste projeto foi verificado que o Webmail IMP, em sua versão CVS, já trazia a funcionalidade de assinatura e cifragem de mensagens de email.

5.7.1 IMP

IMP são as iniciais de Internet Messaging Program, que quer dizer Programa de Mensagens pela Internet. É um sistema de Webmail e um componente do Projeto Horde. O IMP foi a primeira aplicação do Projeto Horde e, na verdade, foi a semente que deu origem a este projeto. O IMP provê acesso IMAP ou POP3 à caixas de correio eletrônico, oferecendo muitas das características presentes em programas de email convencionais, incluindo anexos, catálogo de endereços, pastas múltiplas e suporte a vários idiomas.

5.8 CVS-Control Version System

Todos os módulos do Projeto Horde possuem uma versão estável e uma versão em desenvolvimento. As versões em desenvolvimento de um programa começam a ser testadas e colocadas à disposição em repositórios CVS, onde a cada dia, e muitas vezes, mais de uma vez ao dia, são colocados melhoramentos e correções de problemas encontrados.

As versões de programas que estão em repositórios CVS são sempre as mais atualizadas, visto que são feitas correções todos os dias. Em contrapartida, por ainda estarem em desenvolvimento, as versões CVS acabam apresentando alguns problemas para a sua instalação ou configuração. Quando um programa já possui uma certa estabilidade, é colocada a disposição a versão Alpha do programa. Esta ainda não é a versão definitiva que será colocada a disposição do usuário final, o programa continua sendo atualizado e melhorado no repositório CVS. Após a liberação da versão Alpha, o software continua passando por alterações e melhoramentos até ser colocada a disposição a sua versão Beta. Alguns programas chegam a ter até versões Gama antes de ser colocada a disposição a sua versão estável. No entanto quando um software chega a este ponto geralmente a versão já é chamado de Release-Candidate, para em seguida ser liberada a versão estável do programa. As versões do Webmail IMP, bem como do Horde, utilizadas neste projeto foram as disponíveis no CVS, por serem as mais atualizadas e contarem com a funcionalidade para assinatura e cifragem de mensagens eletrônicas, objetivo deste trabalho.

Capítulo 6

Implantação do Webmail Seguro

6.1 Introdução

Para o desenvolvimento deste projeto será utilizado o Webmail IMP, sua escolha deu-se por ser um projeto *Open Source* (de código aberto, desenvolvido pela comunidade da Internet.). É inteiramente desenvolvido na linguagem PHP.

Completando o sistema de Webmail foi escolhido o Servidor Apache, que junto com o Webmail IMP e a linguagem PHP formam o Servidor de Webmail, onde usuários pertencentes ao domínio em questão, poderão fazer envio e recebimento de mensagens eletrônicas pela internet sem a necessidade de ter que configurar um cliente de e-mail em cada máquina que for querer acessar seus e-mails.

O Servidor Webmail IMP, já trata da questão de segurança em relação a assinatura digital.

6.2 Sistema Webmail

Nesta seção, vamos procurar entender a topologia de um Servidor de Webmail Seguro, segundo o "Túnel Seguro" já implementado e utilizado por alguns servidores de Webmail.

6.2.1 Topologia do Webmail

Como falamos no início desta seção, o Servidor de Webmail é um conjunto composto por um sistema de Webmail, um servidor e uma linguagem de script.

Este servidor utiliza o protocolo IMAP para transporte (Agente de Transporte) de mensagens de um mesmo domínio e é justamente neste transporte que fica o "Túnel Seguro" implementado pela tecnologia SSL, a seguir vamos exemplificar a manipulação de mensagens através de usuários hipotéticos.

6.2.2 Manipulação de Mensagens

- Lendo e organizando mensagens:

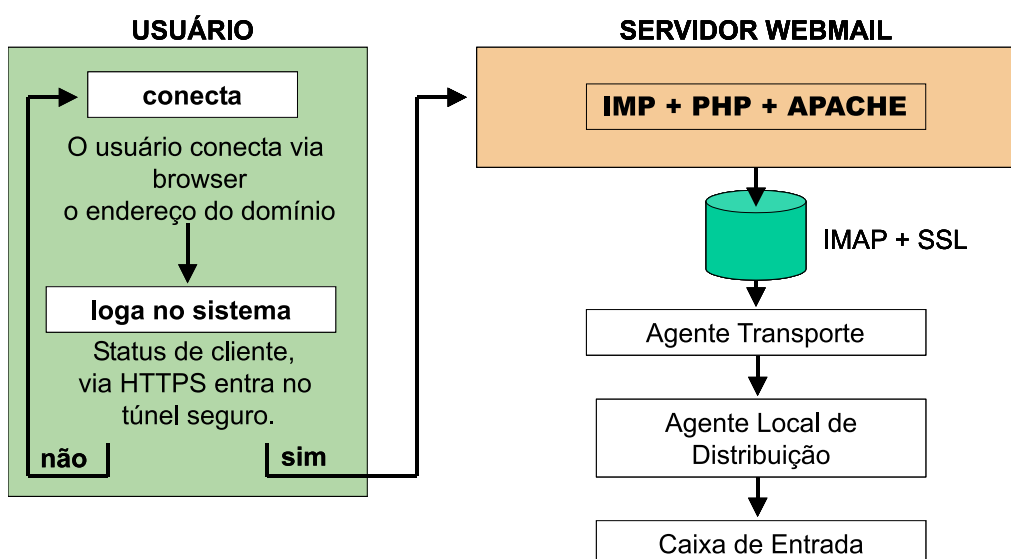


Figura 6.1: Lendo Mensagens. Lendo uma mensagem num sistema de Webmail.

Vamos supor que um determinado usuário deseja ler suas mensagens eletrônicas e organizá-las em seu *Inbox* (Caixa Postal). O usuário conecta-se ao serviço de Webmail utilizando um web browser, Internet Explorer ou Netscape, e acessa o endereço do domínio ao qual está cadastrado.

Ao logar-se no sistema, o usuário adquire o status de cliente e interage com o sistema através do protocolo HTTPS (HTTP sobre SSL). O servidor Apache, recebe os co-

mandos do cliente, passando-os ao IMP, que é interpretado como script PHP pelo próprio servidor e responde ao cliente através de páginas HTML.

Com o sucesso da conexão, o servidor de Webmail (IMP, Apache e Scripts PHP) comunica-se com o Agente de Transporte. Esta comunicação é cifrada por SSL, ou seja, é estabelecido o "Túnel Seguro" por onde os dados trafegam. Estabelecida esta conexão, o Agente de Transporte entra em contato com o Agente Local de Distribuição de mensagens, que é acionado para que o servidor possa acessar o conteúdo do *Inbox* (caixa de entrada) do usuário e manipulá-lo como quiser [CN].

- Envio de mensagens eletrônicas:

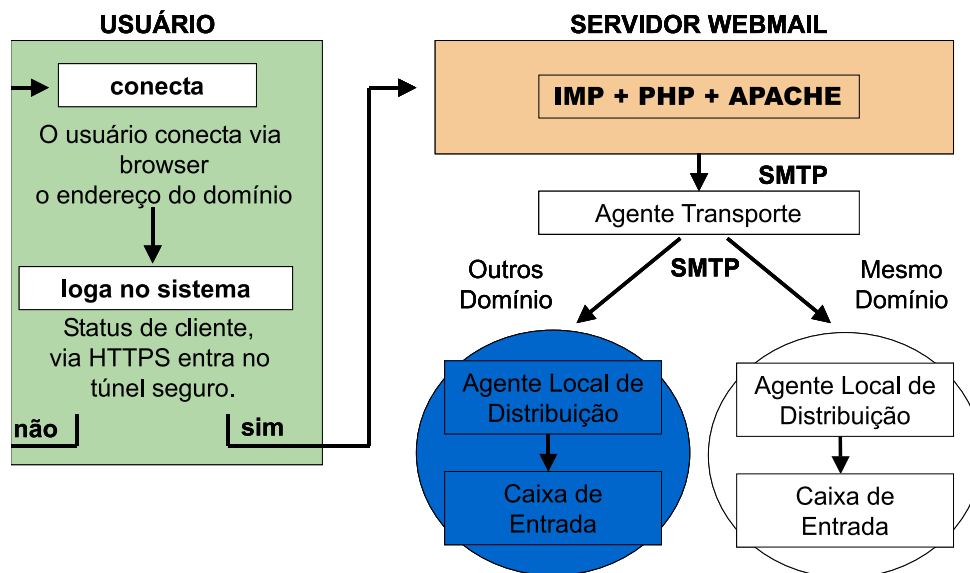


Figura 6.2: Enviando Mensagens. Enviando uma mensagem de um sistema de Webmail.

Há dois tipos de envio de mensagens: à usuários pertencentes ao mesmo domínio e à usuários que pertencem a outros domínios. Os dois tipos de envio de mensagens são similares e dá-se através do protocolo SMTP, responsável pelo envio de mensagens tanto para sistemas de Webmail como para clientes de e-mail. A diferença está em pertencerem ou não ao mesmo domínio, e no segundo caso o agente de transporte do domínio de quem está enviando comunica-se, via SMTP, com o agente de transporte do domínio do destinatário.

Simulando um envio de mensagens temos: digamos que uma usuária Alice deseja enviar uma mensagem ao usuário **Beto**. Alice irá acessar a interface de composição da mensagem preenchendo todos os campos necessários e irá executar o envio da mensagem.

O servidor entrará em contato com um outro agente de transporte via protocolo SMTP, que irá acionar o agente de distribuição para depositar a mensagem na caixa postal do usuário Beto. Este agente de distribuição poderá ser o local, caso Beto pertença ao mesmo domínio de Alice, ou não-local, caso Beto pertença a outro domínio.

6.3 Características de Uso de um Webmail Seguro

Este projeto propõe-se a implantação de um sistema de webmail que possa garantir a segurança da mensagem eletrônica do usuário, no momento em que este usuário esteja acionando o comando de envio do Webmail.

Esta segurança consiste em poder assinar e/ou cifrar uma mensagem de correio eletrônico, garantindo assim a autenticidade e o sigilo da mensagem como já acontece com os clientes de email como o Outlook Express, Microsoft Outlook, e outros.

Quando Alice for compôr uma mensagem terá duas opções extras no sistema de Webmail do IMP, um para cifrar a mensagem caso queira somente esconder os dados e outro para assinar a mensagem caso queira somente se autenticar. Haverá também a possibilidade de selecionar as duas opções caso o usuário queira esconder os dados e autenticar-se.

Para a utilização do sistema, o usuário precisará ter um par de chaves (pública e privada) e fazer sua importação assim como a habilitação ou não do uso de segurança no webmail. Tendo isto, poderá então enviar mensagens seguras pela internet.

Ao abrir a mensagem recebida, o usuário não verá nada além de textos dizendo que a mensagem está cifrada, somente após a habilitação de sua chave privada é que poderá visualizar o texto, pois como a mensagem foi enviada cifrada precisará da chave privada do receptor para decifrar a mensagem, revelando então a mensagem.

6.4 Instalando o Webmail IMP com suporte a S/MIME

Nesta seção será discutido o processo de instalação do Webmail IMP, bem como sua configuração para uso de assinaturas digitais, pacotes necessários para sua instalação e forma de obtenção destes pacotes. O ambiente de teste que serviu como base para a instalação do Webmail IMP foi uma máquina com a seguinte configuração, e sistema operacional Fedora Core-1:

- Processador Athlon 2 Ghz;

- Memória RAM de 256 Mb DDR;
- HD particionado com Linux instalado em uma partição de 20 Gb;
- Placa mãe Soyo Dragon Plus KT333, com placa de som CMI8738 e dispositivo de rede VIA Rhine II ambos on-board.

Posteriormente também foi feita uma instalação em um sistema OpenBSD, numa máquina com a seguinte configuração:

- Processador Celeron 1.1Ghz;
- Memória RAM de 128Mb;
- HD com no mínimo 2Gb de espaço livre.

6.4.1 Um breve histórico

Procuramos utilizar uma distribuição Linux que trouxesse o maior número de pacotes em suas últimas versões. Isto porque, como estamos trabalhando com um software em sua versão CVS, talvez seja preciso algum pacote em sua versão mais atualizada, em especial os pacotes do PHP, já que todos os scripts do Webmail IMP são escritos nesta linguagem. Das distribuições linux analisadas, o Mandrake-10 era, na época, a distribuição que trazia o maior número de pacotes em suas últimas versões, como por exemplo a versão 4.3.4 do PHP, Apache2, além de já vir com a opção de instalação do kernel 2.6. Após optarmos por utilizar a distribuição Mandrake-10, ela foi instalada na máquina com a configuração especificada anteriormente. Ao final da instalação, a máquina foi reiniciada e a interface eth0, que identifica e ativa a placa de rede, falhou ao ser ativada. Foi tentado configurar a placa de rede com os utilitários de configuração do Mandrake mas não houve êxito nas tentativas. Foi instalada então outra placa de rede para testar se havia algum problema com a placa de rede anterior. Tivemos o cuidado de escolher uma placa de rede que estava funcionando em um computador que também estivesse com o Mandrake-10 instalado, para não haver dúvida quanto ao funcionamento da placa.

A mesma placa que no outro computador não apresentava problemas, falhou da mesma forma que a anterior na tentativa de ativação da interface eth0 durante a inicialização do sistema. Concluímos então que se tratava de algum conflito detectado pelo Mandrake-10 que impossibilitava a ativação da interface eth0, impossibilitando assim a utilização de qualquer placa de rede instalada na máquina com a configuração acima. Desta forma optamos por utilizar a distribuição Fedora Core-1 que, da mesma forma que a distribuição Mandrake-10, trazia vários pacotes em suas últimas versões. As diferenças verificadas entre as versões de pacotes contidas nestas duas distribuições estão nos pacotes PHP, que na distribuição Fedora Core-1 vem na versão 4.3.3, e no kernel 2.6 disponível apenas na distribuição Mandrake-10, a distribuição Fedora Core-1 vem com o kernel 2.4.22-1. Apesar de não termos testado a instalação do Webmail IMP com a versão 4.3.3 do PHP, para sabermos se haveria alguma incompatibilidade, resolvemos, por precaução, atualizar o PHP e o servidor Web Apache. A distribuição Fedora Core-1 disponibiliza dois utilitários para atualização de pacotes: o "up2date" e o "yum". Qualquer um destes utilitários pode ser usado tanto para a atualização de algum pacote quanto para a atualização de todo o sistema. Em nosso trabalho foi utilizado o utilitário "yum" para atualizar os pacotes PHP e Apache já instalados no computador.

6.4.2 Configurando o Ambiente

As configurações descritas aqui servem para deixar pronto o ambiente onde será instalado o Horde e o Webmail IMP. Estarão descritos os pacotes necessários tanto para a distribuição Mandrake-10 quanto para a distribuição Fedora Core-1 e para o sistema OpenBSD.

Existe diferença na quantidade de pacotes necessários para cada distribuição. No Mandrake-10, por ser uma distribuição mais modular que a Fedora Core-1, há a necessidade de um número maior de pacotes a serem instalados. Estes pacotes a mais, presentes na distribuição Mandrake-10, podem ser encontrados e feito download na página www.rpmfind.net. A instalação no OpenBSD também segue os mesmos padrões de necessidade de arquivos. A diferença em relação à instalação num sistema Linux está na

forma pela qual alguns arquivos são obtidos e instalados. Para a maioria dos pacotes no OpenBSD, ao se instalar o pacote principal, suas dependências são verificadas e instaladas automaticamente.

As configurações descritas aqui foram feitas no ambiente de teste especificado no início deste capítulo. As diferenças de configuração em relação ao sistema OpenBSD serão descritas ao final desta seção.

A) DNS

O DNS foi configurado para localhost, mas em um ambiente de produção pode ser utilizado o já configurado, como foi o caso da instalação no OpenBSD, pois a máquina já estava com a rede configurada. Para a instalação no ambiente de teste foi utilizado o bind:

- bind-9.2.3-4
- bind-utils-9.2.3-4

Foi verificado com o gerenciador de pacotes "yum" se estas versões estão instaladas e se precisam ser atualizadas. No caso do ambiente de testes estes dois pacotes estavam atualizados.

B) MTA (Mail Transfer Agent)

No ambiente de teste foi utilizado o postfix, mas poderia ser usado qualquer outro MTA, como Sendmail ou Qmail. A escolha pelo postfix se deu por ser o MTA utilizado pela rede do INE, além do fato de ser um MTA leve e modular.

- libpostfix1-2.1.0-0
- postfix-2.1.0-0

Foi verificado com o gerenciador de pacotes "yum" se estas versões estão instaladas e se precisam ser atualizadas. No caso do ambiente de teste, apenas o pacote postfix, versão 2.0.11, estava presente pois o pacote libpostfix não existe para a distribuição Fedora Core-1.

Apenas para reforçar, a inexistência de um pacote em uma distribuição não quer dizer que a configuração do ambiente esteja comprometida. Nos casos verificados neste trabalho tal fato sempre foi devido a maior modularidade de uma distribuição em relação

a outra. No caso o Fedora Core-1 por ser menos modular em relação ao Mandrake-10, necessita de menos pacotes a serem instalados.

C) IMAP

No ambiente de teste foi utilizado IMAP:

- imap-2002d-3

Para habilitar este serviço deve-se editar o arquivo `/etc/xinetd.d/imap`:

alterar:

`disable = yes`

para:

`disable = no`

Após esta mudança, executar o comando:

```
# service xinetd restart
```

No OpenBSD o IMAP foi instalado indo até o diretório `/usr/ports/mail/imap-uw` e digitando o comando:

```
# make install clean
```

Será iniciado o download do pacote e das dependências caso sejam verificadas. Ao final da instalação deve-se efetuar a configuração do arquivo `/etc/inetd.conf`. Neste, deve existir as linhas dos protocolos `pop2`, `pop3` e `imap` conforme orientação ao final da instalação do pacote.

D) MySQL

No ambiente de teste o banco de dados utilizado foi o MySQL, por ser o mais utilizado em distribuições linux. Os pacotes abaixo são para a distribuição Mandrake-10:

- libmysql12-4.0.18-1mdk

- MySQL-common-4.0.18-1mdk

- MySQL-4.0.18-1mdk

- MySQL-client-4.0.18-1mdk

Na distribuição Fedora Core-1 os pacotes disponíveis são:

- mod_auth_mysql-20030510-3
- mysql-server-3.23.58-4
- mysql-3.23.58-4

Verificar com o gerenciador de pacotes "yum" se estas versões estão instaladas e se precisam ser atualizadas.

Para o OpenBSD foi feito o download do servidor de banco de dados MySQL. Para isto digite no prompt o comando:

```
# pkg-add ftp://ftp.das.ufsc.br/pub/OpenBSD/3.4/packages/i386/mysql-server-4.0.18.tgz
```

Para iniciar o servidor digite o commando:

```
# mysqld_safe &
```

É preciso escolher uma senha para o administrador do banco. Para isto digite o comando:

```
# /usr/local/bin/mysqladmin -u root password suassenha
```

Para que o MySQL seja habilitado quando o sistema for iniciar você deverá criar um script de inicialização no arquivo /etc/rc.local

```
if[ -x /usr/local/bin/mysqld_safe]; then
echo -n `mysql`; /usr/local/bin/mysqld_safe &
fi
```

Na última linha do arquivo /etc/rc.conf deve ser trocada a referencia ao /etc/rc.conf.local para rc.local.

Para que o php possa acessar o mysql deve-se alterar o modo de acesso. Troca-se a referencia à localhost pelo ip da maquina. Lembrando, o arquivo é o /var/www/conf/php.ini

D) Servidor Web Apache (HTTPD)

Para a distribuição Mandrake-10, os pacotes necessários são:

- apache2-modules-2.0.48-5mdk
- apache2-mod_cache-2.0.48-5mdk
- apache2-mod_disk_cache-2.0.48-5mdk
- apache2-common-2.0.48-5mdk
- apache2-2.0.48-5mdk
- apache2-manual-2.0.48-5mdk
- apache2-mod_php-2.0.48_4.3.4-1mdk
- apache2-mod_suexec-2.0.48-1mdk

Na distribuição Fedora Core-1, os pacotes disponíveis são:

- httpd-manual.i386 0:2.0.48-1.2
- httpd.i386 0:2.0.48-1.2
- httpd-devel.i386 0:2.0.48-1.2

Verificar com o gerenciador de pacotes "yum" se estas versões estão instaladas e se precisam ser atualizadas.

E) SSL

Para a distribuição Mandrake-10, os pacotes necessários são:

- libopenssl0.9.7-0.9.7c-2mdk
- openssl-0.9.7c-2mdk
- apache2-mod_ssl-2.0.48-5mdk

Na distribuição Fedora Core-1, os pacotes disponíveis são:

- mod_ssl-2.0.47-10
- openssl-0.9.7a-23
- openssl-devel-0.9.7a-23
- docbook-style-dsssl-1.78-2

Verificar com o gerenciador de pacotes "yum" se estas versões estão instaladas e se precisam ser atualizadas.

F) PHP

Na distribuição Mandrake-10 foram encontrados alguns arquivos previamente instalados, mas que não podemos ter certeza se são realmente necessários para a correta configuração do ambiente PHP. Desta forma, será descrito abaixo todos os arquivos encontrados nesta distribuição e, ao lado de cada um, será indicado seu status como necessário ou desnecessário.

- php-xmlrpc-4.3.2-3mdk //necessário
- php-pear-4.3.2-3mdk //necessário
- php-xml-4.3.2-3mdk //necessário
- php-manual-en-4.3.3-1mdk //desnecessário
- php-ldap-4.3.4-1mdk //desnecessário
- php-tclink-4.3.2_3.3.1-15mdk //necessário
- php-rrdtool-4.3.3_1.0.45-1mdk //desnecessário
- libphp_common432-4.3.4-3mdk //necessário
- php-ini-4.3.4-1mdk //necessário
- mod_php-4.3.4-1mdk //necessário
- php-imap-4.3.4-1mdk //necessário
- php-mysql-4.3.4-1mdk //necessário
- php-dba_bundle-4.3.2-4mdk //necessário
- php-cli-4.3.4-3mdk //necessário
- php-readline-4.3.4-1mdk //necessário
- php-domxml-4.3.4-2mdk //necessário
- php-xslt-4.3.2-3mdk //necessário
- php-cgi-4.3.4-3mdk //desnecessário
- php-gd-4.3.4-1mdk //necessário

Já para a distribuição Fedora Core-1, a ferramenta "yum" oferece uma opção de mostrar as opções de pacotes disponíveis para serem instalados ou atualizados. Os pacotes disponíveis e que devem ser instalados são os seguinte:

- php-imap.i386 0:4.3.4-1.1
- php-xmlrpc.i386 0:4.3.4-1.1
- php-odbc.i386 0:4.3.4-1.1
- php-domxml.i386 0:4.3.4-1.1
- php.i386 0:4.3.4-1.1
- php-ldap.i386 0:4.3.4-1.1
- asp2php.i386 0:0.76.2-6
- php-pgsql.i386 0:4.3.4-1.1
- php-mysql.i386 0:4.3.4-1.1
- php-devel.i386 0:4.3.4-1.1
- asp2php-gtk.i386 0:0.76.2-6

É aconselhável que todos os pacotes sejam instalados. No caso de estar faltando algum destes pacotes, baixar através da ferramenta "yum" os que estejam faltando e atualizar os já instalados.

No OpenBSD foi preciso inicialmente disponibilizar o PHP no sistema. Para isto deve-se ir até o diretório `/usr/ports/www/php4/core` e digitar o comando:

```
# make install clean
```

Ao final desta compilação, surgirá uma tela com as seguintes instruções:

- digite o comando: `php -s` - edite o arquivo `/var/www/conf/httpd.conf` - crie o arquivo `php.ini` (a tela diz de onde tirar o modelo)

Agora o sistema deve estar apto a processar arquivos `php`. Para fazer um teste, crie o arquivo `/var/www/htdocs/teste.php` com o seguinte conteúdo: `<? phpinfo(); ?>`

Abra um browser e digite a url:

```
http://xxxxxxxx/teste.php
```

Onde `xxxxxxxx` é o ip da sua máquina OpenBSD. O resultado será algo parecido com a figura a seguir.

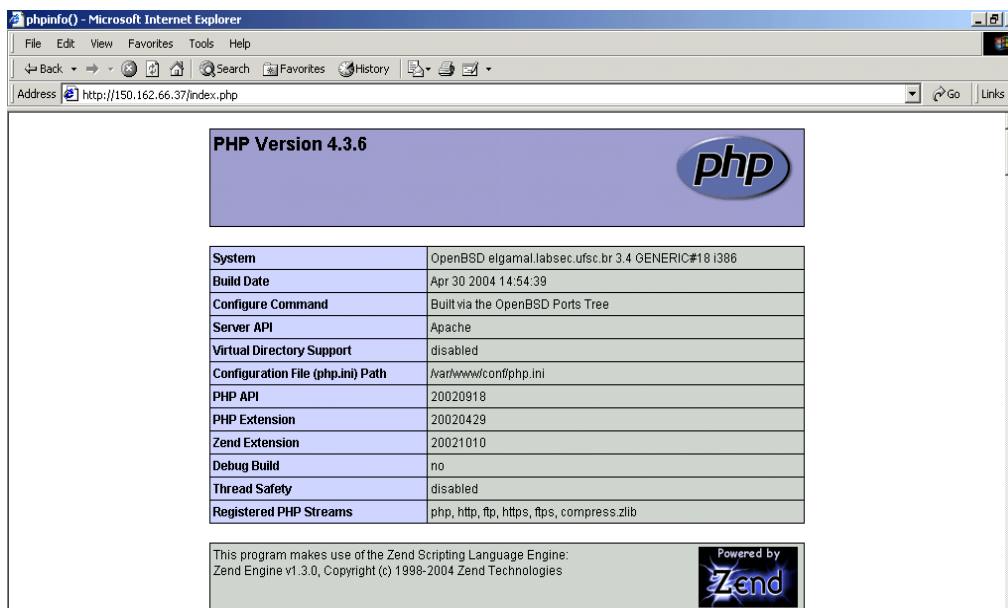


Figura 6.3: PHP no OpenBSD. Resultado de abertura da página do PHP no OpenBSD.

Depois destes passos foi preciso instalar as extensões do php necessárias ao perfeito funcionamento do Horde. Para isto, ir até o diretório `/usr/ports/www/php4/extensions` e digitar o comando:

```
# make package
```

Este comando fará com que sejam compilados os pacotes das extensões do php. Nesta etapa também serão adquiridos e/ou atualizados os pacotes necessários ao funcionamento das extensões do PHP. Esta etapa leva um bom tempo, no caso da máquina especificada onde estava instalado o OpenBSD levou cerca de 4 horas.

Após esta etapa, segue-se com a instalação e ativação dos pacotes de extensão do PHP que foram gerados no passo anterior. Vá até o diretório `/usr/ports/packages/i386/all` e digite o comando:

```
# ls php
```

Este comando listará todos os pacotes iniciados com php contidos neste diretório, conforme mostra a lista a seguir:

- php4-bz2-4.3.6.tgz
- php4-core-4.3.6.tgz
- php4-curl-4.3.6.tgz
- php4-dba-4.3.6.tgz
- php4-dbase-4.3.6.tgz
- php4-dbx-4.3.6.tgz
- php4-domxml-4.3.6.tgz
- php4-extensions-4.3.6.tgz
- php4-filepro-4.3.6.tgz
- php4-gd-4.3.6.tgz
- php4-gmp-4.3.6.tgz
- php4-imap-4.3.6.tgz
- php4-ldap-4.3.6.tgz
- php4-mcrypt-4.3.6.tgz
- php4-mhash-4.3.6.tgz
- php4-mysql-4.3.6.tgz
- php4-ncurses-4.3.6.tgz
- php4-odbc-4.3.6.tgz
- php4-pdf-4.3.6.tgz
- php4-pear-4.3.6.tgz
- php4-pgsql-4.3.6.tgz
- php4-ispell-4.3.6.tgz
- php4-shmop-4.3.6.tgz
- php4-snmp-4.3.6.tgz
- php4-sybase_ct-4.3.6.tgz
- php4-xmlrpc-4.3.6.tgz
- php4-xslt-4.3.6.tgz

Será necessário garantir que cada um dos pacotes listados, esteja instalado e ativado.

Para instalar um pacote digite `pkg_add nome_do_pacote.tgz`. Por exemplo, para instalarmos o pacote `php4-bz2-4.3.6.tgz` devemos digitar o comando:

```
# pkg_add php4-bz2-4.3.6.tgz
```

Após a instalação é gerada uma instrução de ativação conforme mostrado abaixo, siga-a corretamente.

```
Digite: pkg_add php4-bz2-4.3.6.tgz
```

```
Resultado: pkg_add(php4-bz2-4.3.6): package already recorded as installed
```

```
Digite: pkg_add php4-core-4.3.6.tgz
```

```
Resultado: pkg_add(php4-core-4.3.6): package already recorded as installed
```

```
Digite: pkg_add php4-curl-4.3.6.tgz
```

```
Resultado: Enable this module in php.ini using the following command:
```

```
/usr/local/sbin/phpxs -a curl
```

```
Digite: /usr/local/sbin/phpxs -a curl
```

```
Resultado: Activating extension: curl
```

6.4.3 Download do Horde, IMP e Framework

Para fazer o download de forma correta dos pacotes Horde, IMP e Framework, deve-se inicialmente estar logado como super-usuário. Comando:

```
# su
```

Após isto é preciso ir para o diretório base do http (DocumentRoot):

```
# cd /var/www/html/ (Para sistemas Linux)
```

```
# cd /var/www/html/ (Para sistemas OpenBSD)
```

Neste diretório deve-se digitar o seguinte comando:

```
# export CVSROOT=:pserver:cvsread@anoncvs.horde.org:/repository
```

```
# cvs login
```

Após estes comandos será solicitado que seja digitada a senha para logar no repositório CVS como usuário anônimo (anonymous user). A senha é 'horde'. Estas instruções também estão disponíveis na página do Horde (www.horde.org). Uma vez completado os passos acima, já é possível fazer o download de qualquer módulo do repositório CVS. Os módulos utilizados neste trabalho foram "horde", "imp" e "framework".

Para baixar um módulo do repositório CVS deve-se usar o seguinte comando:

```
# cvs co ;nome_do_modulo
```

No caso, para baixar os módulos utilizados neste trabalho deve ser seguido uma ordem. Primeiramente, dentro do diretório /var/www/html/ digitar o comando:

```
# cvs co horde
```

Este comando irá criar um diretório chamado "horde" dentro de /var/www/html/. Agora devemos entrar neste diretório:

```
# cd horde
```

Dentro do diretório /var/www/html/horde/ (ou /var/www/htdocs/horde no caso do OpenBSD) usar os comandos:

```
#cvs co imp
```

```
#cvs co framework
```

Estes comandos irão baixar os pacotes "imp" e "framework" respectivamente.

6.4.4 Instalar os pacotes do Horde

Uma vez feito o download dos pacotes "horde", "imp" e "framework" deve ser usado o comando de instalação dos pacotes. Para isto devemos entrar no diretório /horde/framework/. Como já estávamos dentro do diretório /horde/ para instalar os pacotes, basta entrar no diretório /horde/framework/ com o comando:

```
# cd framework
```

E em seguida digitar o comando:

```
# php install-packages.php
```

Isto configura os pacotes em /usr/share/pear. Isto pode variar, sendo importante verificar a variável `include_path` do arquivo `php.ini`. No sistema Linux, caso falte algum pacote, executar o comando:

```
# pear install ;nome_do_pacote>
```

Por exemplo: `# pear install Log`

O comando `pear` se conecta na internet e baixa uma versão atualizada do pacote `Log`.

6.4.5 Configurar o Horde

A) O horde, tem como padrão distribuir apenas os arquivos `.php.dist` para serem gerados os `.php`. Para gerar os arquivos `.php` deve-se entrar no diretório `/horde/config/` e executar o seguinte comando:

```
# for foo in *.dist; do cp $foo 'basename $foo .dist'; done
```

É importante notar que não são usadas aspas simples, mas sim sinais de crase!

B) Independente de como ficará a configuração final, é interessante ter algum tipo de autenticação para logar no horde e fazer a configuração inicial. Isto é feito no diretório `/horde/config/`, no arquivo `conf.php`. No sistema Fedora Core-1 foram feitas as seguintes mudanças para que seja feita autenticação pelo `imap`:

(Editar a linha)

```
$conf['auth']['driver'] = 'imap';
```

(Adicionar as linhas)

```
$conf['auth']['params']['folder'] = 'INBOX';
```

```
$conf['auth']['params']['hostspect'] = 'localhost';
```

```
$conf['auth']['params']['port'] = 143;
```

```
$conf['auth']['params']['protocol'] = 'imap/notls/novalidate-cert';
```

Configurar a linha do conf.php:

```
$conf['log']['enabled'] = true;
```

para

```
$conf['log']['enabled'] = false;
```

C) Devemos agora criar o banco de dados MySQL que o Horde irá usar para guardar as configurações dos usuários. Para isso inicialmente entramos no diretório /horde/scripts/db/ com o comando:

```
# cd /var/www/html/horde/scripts/db/
```

ou, conforme o sistema onde você estará instalando:

```
# cd /var/www/htdocs/horde/scripts/db/
```

Dentro deste diretório digitar o comando:

```
# mysql < mysql_create.sql
```

O script mysql_create.sql cria um database chamado "horde" e um usuário chamado "horde" com senha "horde" que possui direitos totais sobre o database. Agora devemos modificar as permissões, usando os seguintes comandos:

```
# mysql horde
```

Com este comando entramos no prompt do MySQL. Devemos agora nos referenciar ao database "horde" para efetuar as modificações nas permissões. Para isto usa-se o comando:

```
mysql> use horde;
```

Notar que estamos agora no ambiente MySQL, por isso estamos com o prompt "mysql>" a frente do comando "use horde;". Agora podemos modificar as permissões com o comando:

```
mysql> grant all privileges on *.* to horde@localhost identified by "";
```

Sair do MySQL com o comando:

```
mysql> quit;
```

E agora editar o arquivo conf.php em /horde/config/ alterando as linhas:

```
$conf['prefs']['driver'] = 'none';
```

para

```
$conf['prefs']['driver'] = 'sql';
```

e

```
$conf['sql']['password'] = '';
```

para

```
$conf['sql']['password'] = 'horde';
```

Por padrão o horde vai se conectar com usuário horde@localhost e senha em branco, então não é necessário alterar mais o arquivo conf.php.

D) Para conferir se está ok, logar com um usuário/senha do sistema em <http://localhost/horde> ou em <http://xxxxxxxxx/horde> no caso de seu sistema estar com a rede configurada de outra forma que não como localhost. Uma tela como a seguinte deverá aparecer:

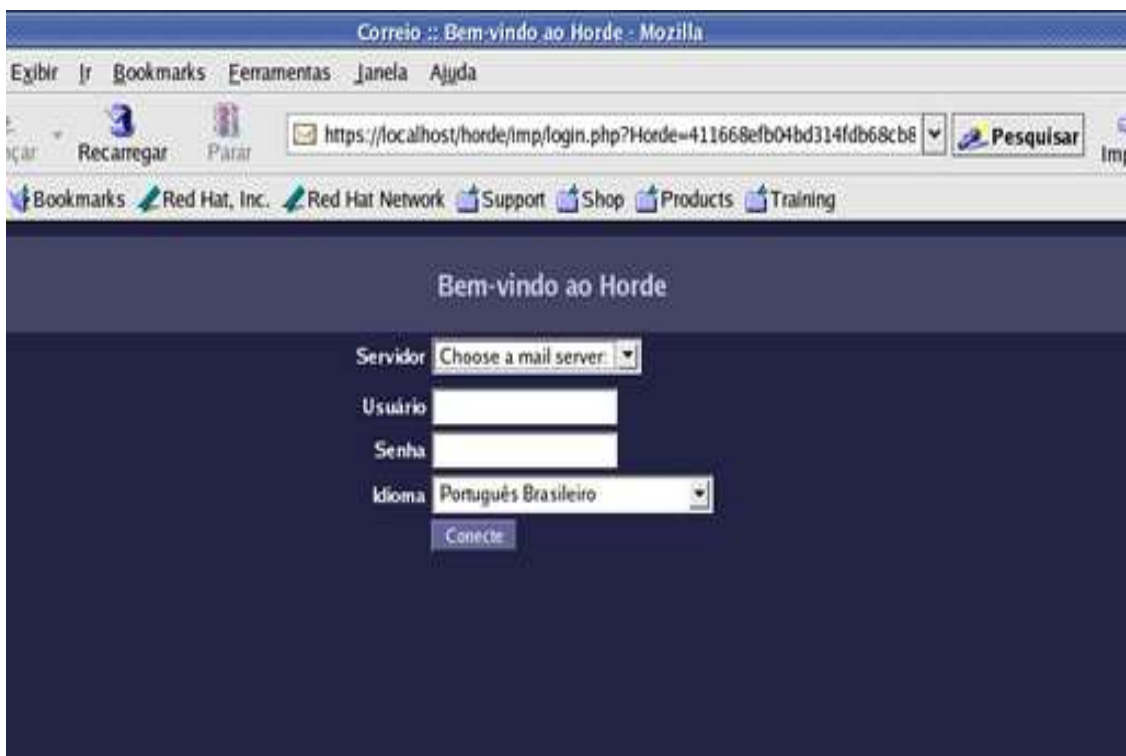


Figura 6.4: Inicia Horde. Tela inicial do Horde.

Ao acessar <http://localhost/horde> talvez apareçam algumas mensagens de erro, fazendo com que alguns ajustes ainda tenham que ser feitos. No caso deste trabalho, durante a instalação no sistema Linux, a seguinte mensagem apareceu antes de podermos logar no horde normalmente:

MENSAGEM:

Warning: main(Log.php): failed to open stream: Arquivo ou diretório não encontrado in /usr/share/pear/Horde.php on line 3

Warning: main(): Failed opening 'Log.php' for inclusion (include_path='.:usr/share/pear') in /usr/share/pear/Horde.php on line 3

Notice: Use of undefined constant PEAR_LOG_EMERG - assumed 'PEAR_LOG_EMERG' in /usr/share/pear/Horde.php on line 195

Fatal error: Undefined class name 'log' in /usr/share/pear/Horde.php on line 74

Este erro ocorreu devido a ausência do pacote "Log" no sistema. Para instalar este pacote usou-se o comando:

```
# pear install Log
```

E) Após conseguir visualizar a tela de entrada do horde, algumas configurações ainda precisam ser feitas. Para que se possa fazer algumas configurações no horde através de sua própria interface, não sendo preciso editar os arquivos diretamente, é necessário que apareça a opção "Administration" no menu do Horde. Para isto deve-se alterar o arquivo conf.php em /horde/config/ na seguinte linha:

```
$conf['auth']['admins'] = array();
```

para

```
$conf['auth']['admins'] = array('usuário_local');
```

O 'usuário_local' colocado como parâmetro, deve ser um usuário do sistema. Este usuário será o administrador do Horde, e somente ele terá a opção "Administration" em seu ambiente horde. A opção "Administration" é mostrada abaixo:

F) Agora que podemos acessar o menu "Administration", devemos configurar o Horde para que ele guarde as configurações dos usuários nas tabelas SQL criadas na item C). Para isto devemos logar no Horde com o usuário administrador do horde. Uma vez logado, ir em Administration-¿Configuration-¿Horde. Lá existe uma ordem certa para configurar cada guia. Primeiro devemos ir na guia "Database". Nesta guia devemos fazer as seguintes alterações:

- Em "If Horde uses a database, which database backend are we using? If you are not using a database for anything, you can ignore this entire section.", colocar a opção "MySQL".
- Em "Request persistent connections?", deixar em branco.
- Em "What protocol will we use to connect to the database?", colocar "TCP/IP".
- Em "Port the DB is running on, if non-standard", colocar "3306".

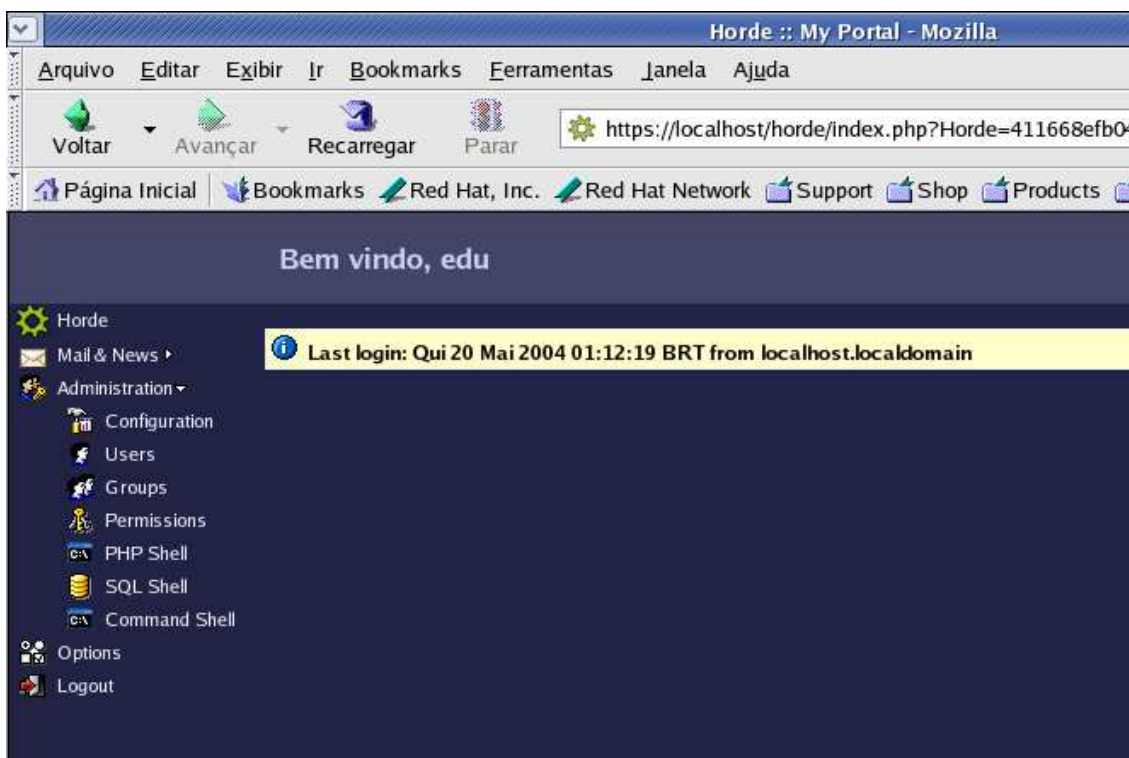


Figura 6.5: Menu Administração. Mostra o menu de Administração do Horde.

- Em "What hostname is the database server running on, or what is the name of the system DSN to use?", colocar "localhost".
- Em "What username do we authenticate to the database server as?", colocar "horde".
- Em "What password do we authenticate to the database server with?", colocar "horde".
- Em "What database name/tablespace are we using?", colocar "horde".
- Em "What charset does the database use internally?", colocar "iso-8859-1"

Agora devemos ir para a guia "Preference System". Nela configurar o seguinte:

- Na primeira opção, deixar em branco.
- Na opção "What preferences driver should we use?", colocar "SQL Database".

- Em "Driver configuration", colocar "Horde defaults".
- Deixar a última opção em branco.

Da mesma forma que foi configurada a guia "Preference System", ir na guia "DataTree System" e configurar:

- Em "What backend should we use for Horde DataTree storage?", colocar "SQL Database".
- Em "Driver configuration", colocar "Horde defaults".
- Em "The name of the data table in the database [horde_datatree]", deixar em branco. Mas observar a tabela que se encontra entre as chaves. Ela deve ser a tabela "horde_datatree".

A última guia a ser configurada será a "Authentication". Lá modificar o seguinte:

- Em "Which users should be treated as administrators (root, super-user) by Horde?", deixar o usuário já configurado.
- Deixar a segunda opção selecionada.
- Em "What backend should we use for authenticating users to Horde?", colocar "Let a Horde application handle authentication".
- Em "The application which is providing authentication", colocar "imp".

Depois de feitas estas alterações, gerar o arquivo conf.php clicando no botão "Generate Horde Configuration", copiar o conteúdo mostrado e colar no arquivo conf.php já existente, substituindo seu conteúdo pelo recém gerado. Salvar o arquivo e logar novamente para testar se as configurações estão sendo guardadas. Um simples teste que pode ser feito, é logar no Horde, deslogar, e logar novamente. Na segunda vez que for logar, deverá aparecer a data e hora do último login. Caso isto não aconteça, irá continuar a mostrar como último login a palavra "Never" que significa "nunca", ou seja, o Horde continua

a não guardar as opções nas tabelas SQL criadas no item C). Neste caso os passos anteriores devem ser feitos novamente, com a diferença que a cada guia modificada deverá ser gerado o novo arquivo conf.php, colar seu conteúdo no conf.php existente, fazer o logout do Horde, e logar novamente. Isto para cada aba modificada. Devemos lembrar que isto só será necessário caso a geração do arquivo conf.php feita após a modificação de todas as guias não surta efeito, fazendo com que o Horde continue a não guardar as configurações. Isto pode variar pois estamos lidando com uma versão em desenvolvimento, ou seja, alguns bugs podem aparecer.

G) Para configurar o Horde para funcionar apenas em um ambiente seguro, https (necessário para importação de chave privada S/MIME e também para melhorar a segurança geral do sistema), deve-se configurar as seguintes linhas do conf.php:

```
$conf['use_ssl'] = 2;
```

para

```
$conf['use_ssl'] = 1;
```

e

```
$conf['server']['port'] = $_SERVER['SERVER_PORT'];
```

para

```
$conf['server']['port'] = 443;
```

H) Agora, antes de usar o horde, temos que verificar se há algum pacote ainda a ser instalado. Acessando o endereço <http://localhost/horde/test.php> verificamos a configuração do horde. Na implementação deste trabalho no sistema Linux, faltaram os seguintes pacotes:

- MCAL
- Mcrypt
- MIME Magic

Foram instalados apenas os pacotes Mcrypt e MIME Magic, pois o pacote MCAL não era necessário para o objetivo específico deste trabalho.

A instalação do Mcrypt, pacote php-mcrypt-4.3.4-1mdk.i586.rpm possui dependência com os seguintes pacotes:

- libmcrypt
- php432

O pacote php432 não foi encontrado para ser instalado. Desta forma ao instalar o pacote php-mcrypt foi usada a opção "--nodeps".

Outra dependência que não está habilitada é opção "memory_limit disable" do PHP. Se a opção de limite de memória interna do PHP estiver habilitada e não estiver com um valor alto o suficiente, o Horde não vai conseguir lidar com grandes volumes de dados, como por exemplo arquivos anexados em mensagens do IMP. Caso não seja possível recompilar o PHP sem a opção "--enable-memory-limit" então o valor em "memory_limit" dentro do arquivo "php.ini" deve estar setado para um valor suficientemente alto. Neste trabalho optamos por não recompilar o PHP e utilizar um valor alto o suficiente no "memory_limit". Por padrão o valor setado é de 8 Mb, mas no ambiente de teste este valor foi aumentado para 64 Mb.

6.4.6 Configurar o IMP

A) Da mesma forma que o Horde, o IMP também traz os arquivos .php no formato .php.dist. Para gerar os arquivos .php devemos ir /horde/imp/config/ e executar o comando:

```
# for foo in *.dist; do cp $foo 'basename $foo .dist'; done
```

Novamente, atentar para o detalhe de que não são aspas simples que estão neste comando, mas sim sinais de crase!

B) Agora devemos gerar o arquivo conf.php do IMP. Para isto devemos logar novamente no horde e ir em Administration/Configuration/Mail(imp), gerar o arquivo de configuração e colocar o conteúdo em /horde/imp/config/ criando um arquivo conf.php.

C) Novamente temos que verificar se há algum pacote ainda a ser instalado. Acessando o endereço <http://localhost/horde/imp/test.php> verificamos a configuração do IMP.

D) Para configurar o IMP para acessar um servidor IMAP ou POP3 deve-se editar o arquivo `servers.php` dentro do diretório `/horde/imp/config/` nas seguintes linhas:

```
$servers['imap'] = array(  
  
    'name' => 'IMAP Server',  
    'server' => 'localhost',  
    'hordeauth' => false,  
    'protocol' => 'imap/notls',  
    'port' => 143,  
    'folders' => 'mail/',  
    'namespace' => '',  
    'maildomain' => 'localhost',  
    'smtp host' => 'localhost',  
    'realm' => '',  
    'preferred' => '',  
    'dotfiles' => false,  
    'hierarchies' => array()  
  
);
```

Esta foi a configuração padrão usada para acesso local na máquina Linux. Para acessar outro servidor de email, como por exemplo o servidor da INF, foi feita a seguinte configuração, abaixo da configuração feita acima:

```
$servers['imap2'] = array(  
  
    'name' => 'INE',  
    'server' => 'imap.inf.ufsc.br',  
    'hordeauth' => false,  
    'protocol' => 'imap/notls',  
    'port' => 143,  
    'folders' => 'mail/',
```

```

'namespace' => "",
'maildomain' => 'inf.ufsc.br',
'smtp host' => 'smtp.inf.ufsc.br',
'realm' => "",
'preferred' => "",
'dotfiles' => false,
'hierarchies' => array()
);

```

Com estas duas configurações estaremos habilitando o login tanto localmente quanto na rede INF. No caso de se logar na rede INF, com o computador fora desta rede, é possível ler os emails e deleta-los da caixa de entrada, mas não será possível enviar emails para endereços @inf.ufsc.br. Isto porque a rede INF utiliza SMTP autenticado para o envio de emails para endereços @inf.ufsc.br. Ou seja, quando o computador não estava dentro da rede INF era possível ler os emails na caixa de entrada mas não era possível enviar emails. Nas vezes que o computador estava dentro da rede INF o envio de emails não teve problemas.

E) Para habilitar o uso do S/MIME para todos os usuários do sistema, devemos editar o arquivo prefs.php dentro do diretório /horde/imp/config/ alterando as seguintes linhas:

```

$_prefs['use_smime'] = array(

    'value' => 0,
    'locked' => false,
    'shared' => false,
    'type' => 'implicit'

);
para
$_prefs['use_smime'] = array(

```

```
'value' => 1,
'locked' => false,
'shared' => false,
'type' => 'implicit'
);
```

Desta forma o S/MIME estará habilitado por padrão para todos os usuários. Os usuários que não quiserem utilizar S/MIME ainda podem configurar em Options->Mail->S/MIME Options.

F) Em sua configuração inicial, o Horde vem com os menus de todos os módulos que podem ser instalados. A aparência pode ser vista na figura abaixo:

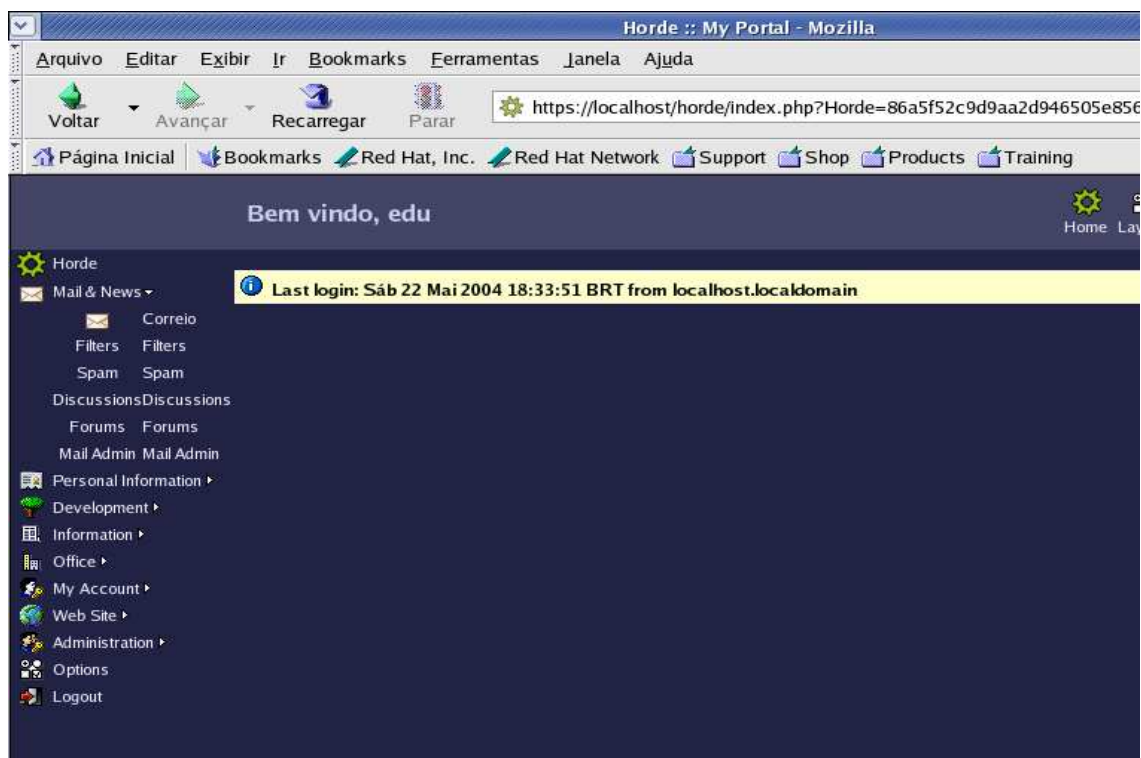


Figura 6.6: Menu Horde Completo. O Horde por padrão vem com todos os menus.

Como mostra a figura, cada item do menu possui submenus que representam diferentes aplicações, disponíveis caso o módulo desta aplicação estiver instalado e configurado corretamente. Como para este trabalho foi instalado apenas o módulo "imp" do

Horde, é aconselhável desabilitar os outros menus que não possuem módulos correspondentes instalados. Para isto devemos editar o arquivo `registry.php` presente no diretório `/horde/config/` e comentar todas as linhas dos aplicativos que não estiverem instalados. Fazendo isso a aparência do Horde apenas com o IMP instalado ficará como mostrado na figura abaixo:

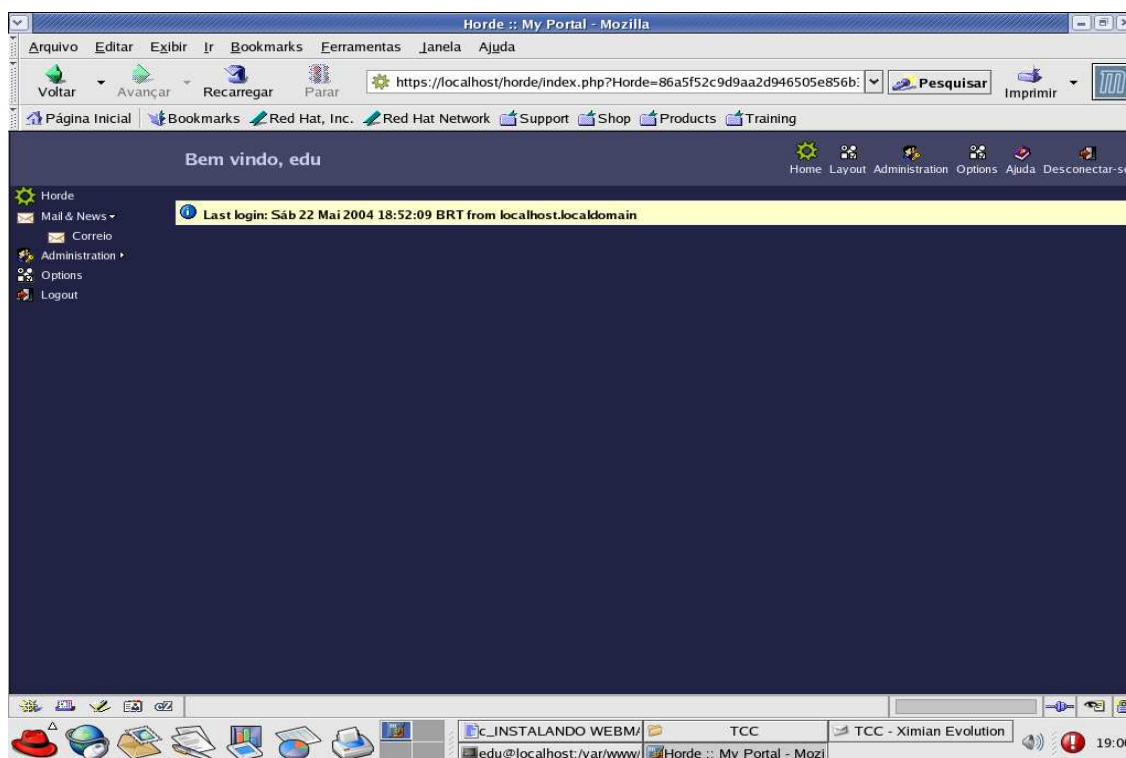


Figura 6.7: Menu Horde Configurado. Após a configuração do Horde para mostrar somente os menus utilizados.

Configurar para usar o mesmo login do horde e não precisar logar sempre duas vezes quando logamos no horde e depois no imp.

6.4.7 Importando as chaves S/MIME

Para assinar digitalmente uma mensagem, você deve primeiramente importar seu par de chaves pública e privada. No IMP para você usar assinatura S/MIME, você deverá importar as duas chaves separadamente. É importante notar que o IMP não suporta arquivos no formato pkcs#12, somente arquivos no formato pkcs#7 são suportados. Portanto

ao gerar os arquivos com as suas chaves, assegure-se de que estará gerando arquivos neste formato. A geração das chaves pública e privada usadas para teste neste trabalho, foram geradas utilizando as funcionalidades do OpenSSL. Para gerar um arquivo contendo as chaves privada e pública foi usado o comando:

```
# openssl genrsa -des3 -out chave.pem 1024
```

O arquivo chave.pem conterà as duas chaves, sendo que a chave privada será cifrada usando o des3. Uma senha será solicitada na execução do programa. Em seguida deverá ser gerada uma requisição de certificado digital. Esta requisição foi gerada usando o seguinte comando:

```
# openssl req -new -key chave.pem -out requisicao.pem
```

O arquivo requisicao.pem contém a requisição do certificado em formato pkcs#10 a ser submetido a uma autoridade certificadora. No caso, esta requisição foi submetida à Autoridade Certificadora do LabSEC na UFSC. (www2.labsec.ufsc.br/certsrv). Após este processo será recebido da Autoridade Certificadora um arquivo contendo o certificado digital. Agora você poderá instalar o certificado e usar a chave pkcs#7. O arquivo recebido da AC para teste neste trabalho foi o juchem-certificado.pem.

Acesse o menu Options->Mail->S/MIME Options. Você verá a tela abaixo:

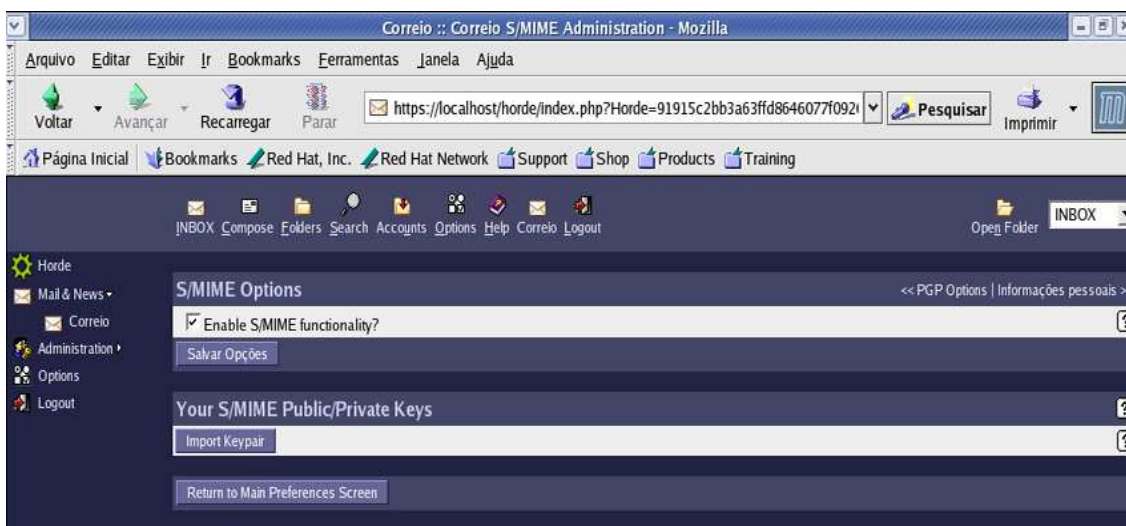


Figura 6.8: Importando Chaves. Mostrando a importação da chaves.

Antes de prosseguir assegure-se de que seu browser não está configurado para bloquear janelas popup. Deixe esta opção desabilitada, caso contrário a janela para a importação das chaves não aparecerá. Verificado este detalhe, clique no botão "Import Keypair". A janela abaixo deverá aparecer:



Figura 6.9: Importando Chaves - Arquivos. Inserindo os arquivos do par de chaves.

Aqui você deverá importar primeiro a sua chave pública. Uma vez selecionado o arquivo que contém sua chave pública, você deverá importar o arquivo que contém a sua chave privada. Após ter selecionado os arquivos corretamente você deverá se deparar

com a seguinte tela:

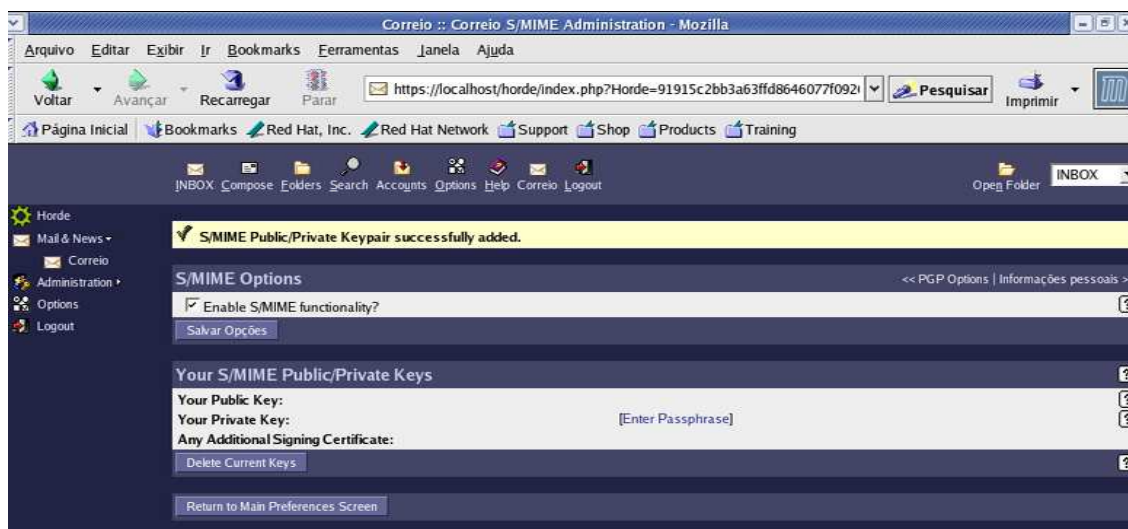


Figura 6.10: Importando Chaves - Resultado. Resultado da importação.

Pronto, as suas chaves S/MIME pública e privada estarão adicionadas ao seu perfil. A cada seção do IMP, na primeira vez que você for enviar uma mensagem assinada digitalmente, você deverá informar uma senha (Passphrase), que é a mesma utilizada na geração do arquivo que contém a sua chave privada.

6.4.8 Enviando um email assinado com S/MIME

Uma vez que você conseguiu importar as suas chaves pública e privada, você poderá utilizá-las para assinar as suas mensagens. Entrando no IMP você irá visualizar as suas mensagens em sua caixa de entrada. Clique em "Compose" para abrir uma janela onde você irá escrever sua mensagem:



Figura 6.11: Compondo Email Assinado. Esta figura mostra como deve-se compôr o email assinado.

Depois de ter endereçado corretamente sua mensagem, colocado o assunto e escrito o texto da mensagem, você deverá selecionar a opção "S/MIME Sign Message" na caixa de seleção "Encryption Options:" mostrada na figura abaixo:

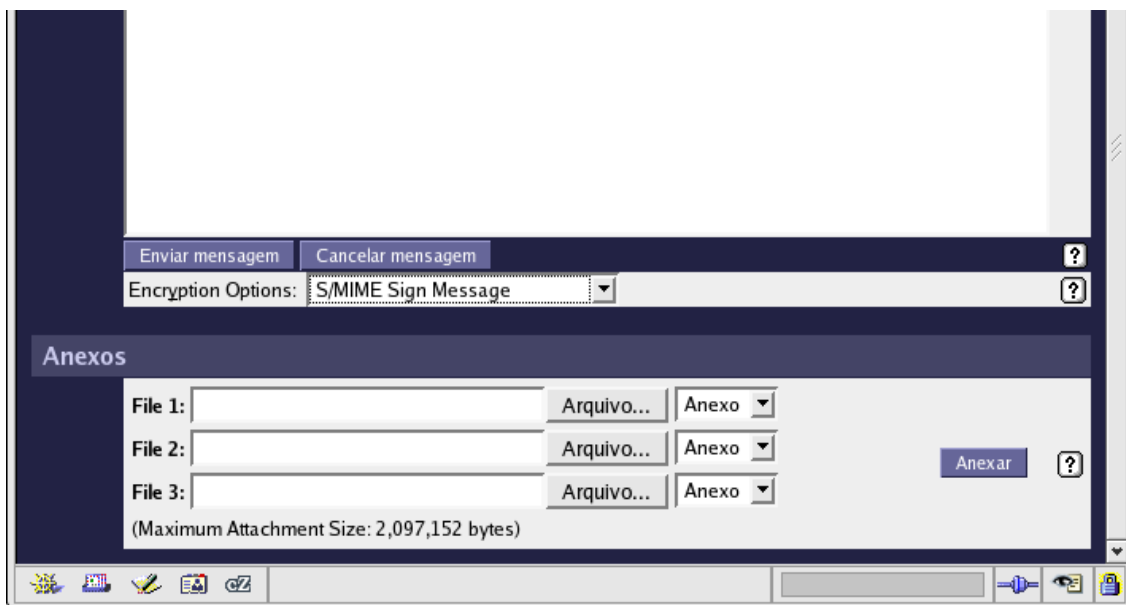


Figura 6.12: Assinando Email. Esta figura mostra como deve-se assinar a mensagem.

Em seguida você poderá clicar em "Enviar Mensagem". Se este for o primeiro email assinado que você estiver enviando nesta seção do IMP, irá aparecer para você a janela abaixo para que você coloque a senha (Passphrase) que originou o arquivo com a sua chave privada:

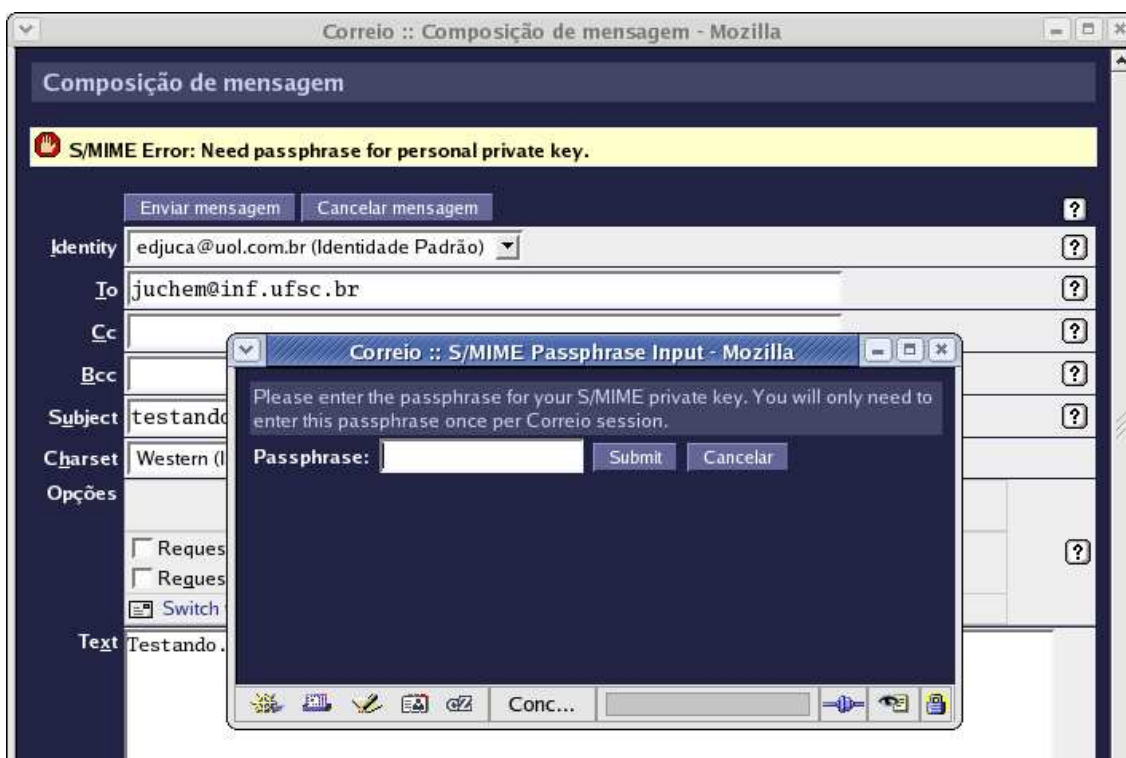


Figura 6.13: Passphrase. Colocando sua senha ao enviar seu primeiro email assinado.

Pronto, você enviou sua mensagem assinada digitalmente com S/MIME pelo Web-mail IMP.

6.4.9 Lendo uma mensagem assinada digitalmente pelo IMP

A figura abaixo mostra a mensagem assinada digitalmente sendo lida pelo IMP.

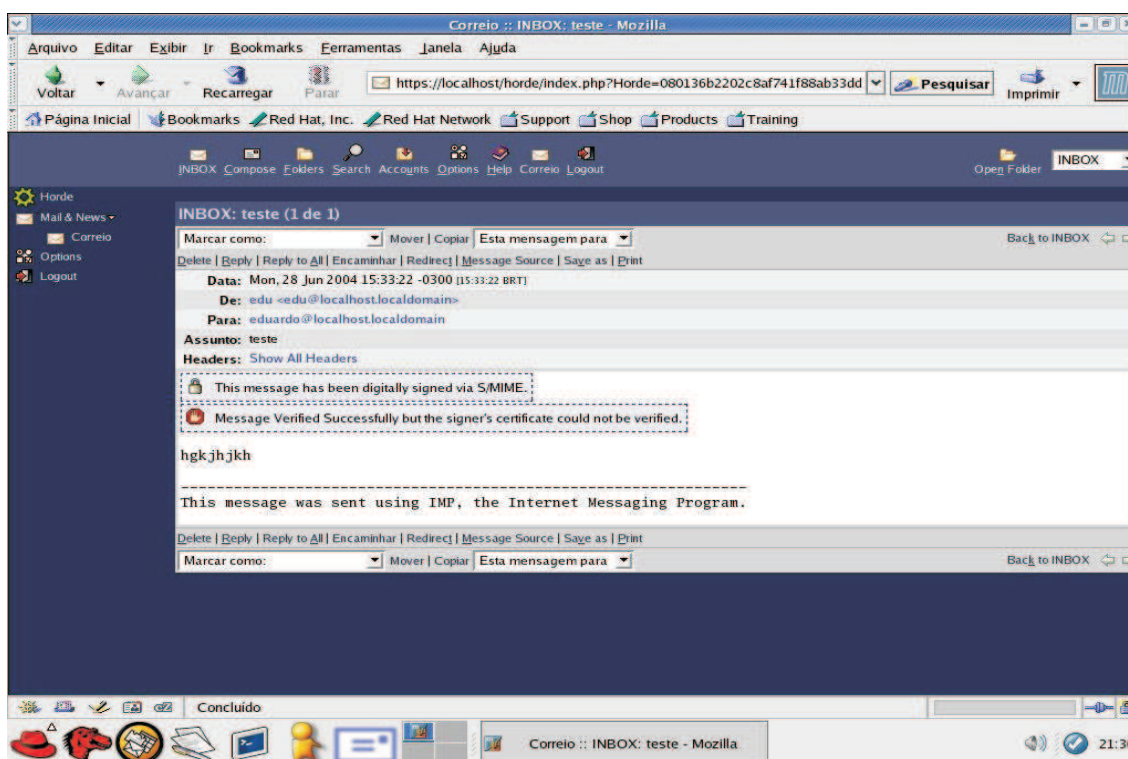


Figura 6.14: Mensagem Assinada. Recebendo uma mensagem assinada.

Nesta mensagem está sendo mostrado dois avisos. O primeiro informa que a mensagem foi assinada com S/MIME. O segundo aviso informa que a assinatura foi verificada com sucesso, mas que o certificado do emissor não pôde ser verificado. Isto aconteceu pois o email do remetente difere do email do signatário da mensagem.

Esta advertência é melhor visualizada em um cliente de email comum. No Outlook Express, por exemplo, ao abrirmos esta mensagem nos é mostrado uma tela de advertência atentando para este fato.

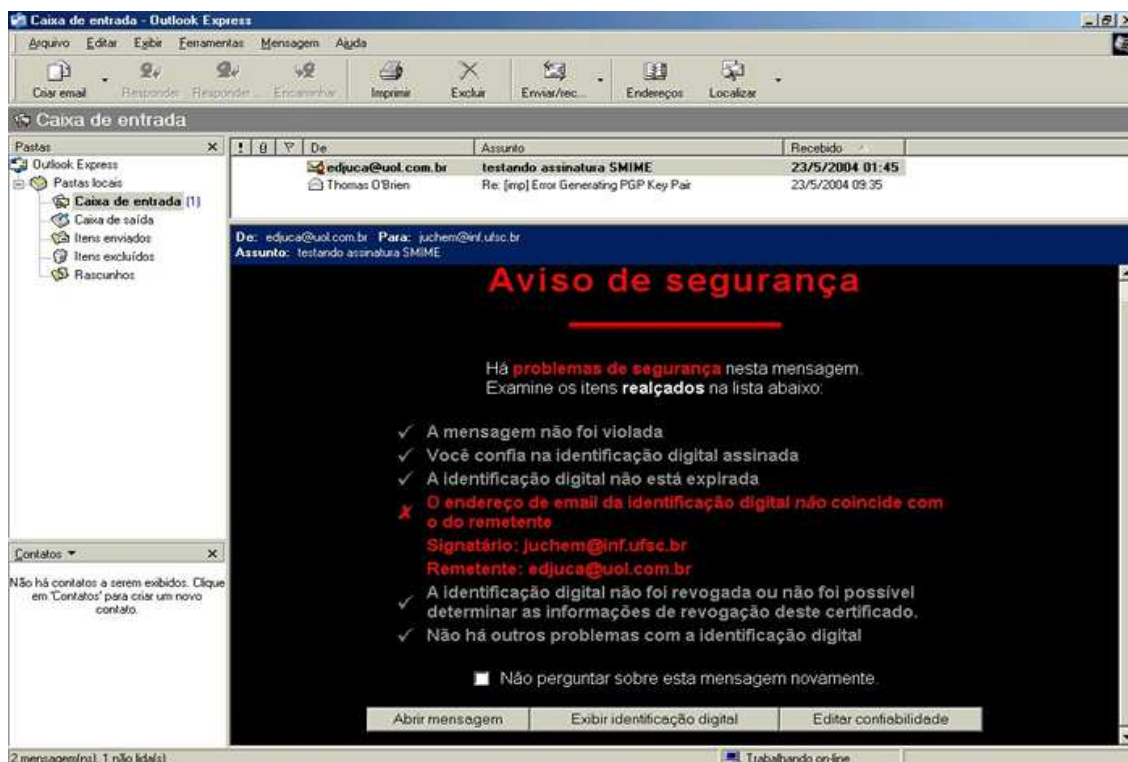


Figura 6.15: Mensagem Assinada - Aviso. Aviso de Segurança.

Ao receber uma mensagem assinada e aparecer a tela acima de Aviso de Segurança, significa que algum dos problemas a seguir podem ter ocorrido:

- a mensagem pode ter sido violada;
- seu certificado pode não estar confiando na identificação digital assinada;
- o endereço de email da identificação digital pode não coincidir com a do remetente;
- a identificação digital pode ter sido revogada.

Ainda neste aviso, você terá as opções de abrir a mensagem, exibir a identificação digital como mostra a próxima figura ou editar a confiabilidade.

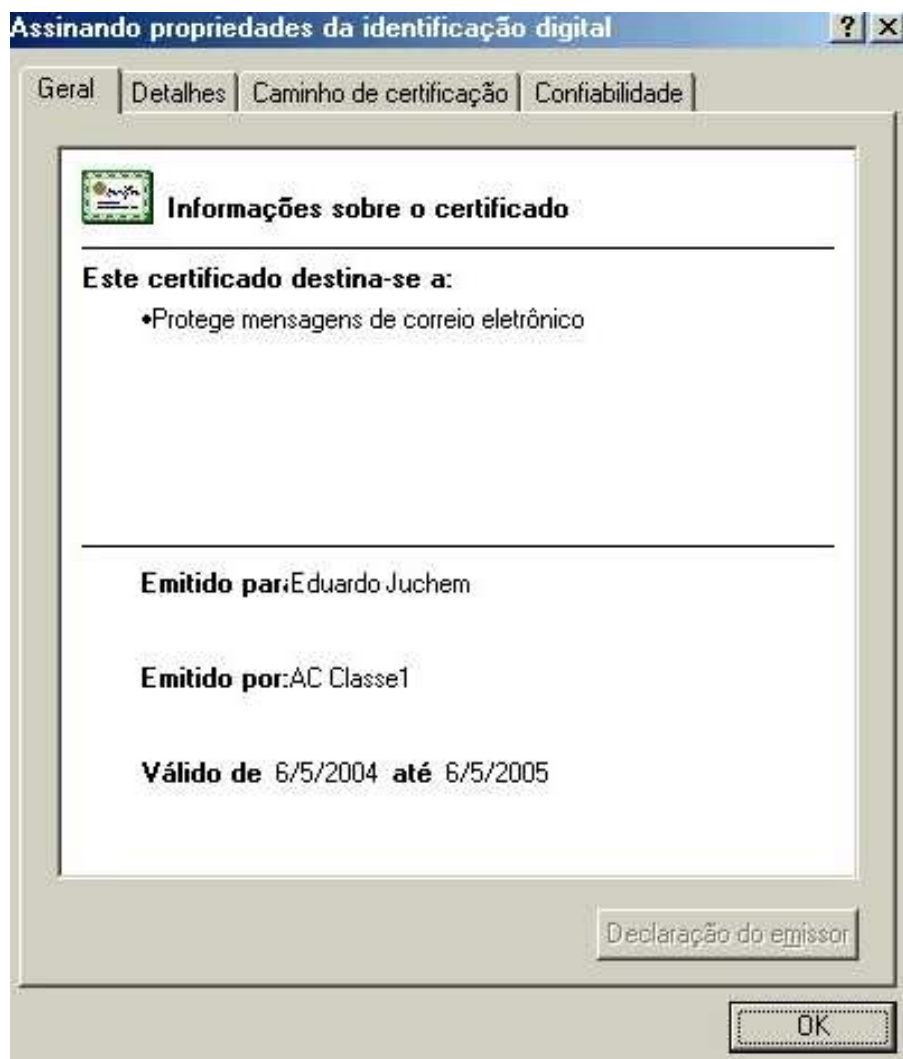


Figura 6.16: Mensagem Assinada - Certificado. Mostrando o certificado.

6.4.10 Atualizando a sua versão CVS do IMP

Ao trabalhar com versões CVS de qualquer software é sempre interessante atualizar os seu pacotes todos os dias, já que são dispostas atualizações e correções diariamente no repositório CVS. Com o IMP não é diferente. Atualizando sua versão CVS do IMP você sempre terá em sua máquina a última versão do software e muitos problemas poderão ser solucionados apenas com esta atualização. Para fazer esta atualização é aconselhável que sejam baixados todos os módulos instalados na máquina e não apenas o IMP. Neste trabalho temos instalados os módulos "horde", "framework" e "imp". Como estes módulos obedecem uma estrutura de diretório, devemos ficar atentos ao diretório em que estamos no momento de fazer esta atualização. Inicialmente, deverão ser feitos os dois passos iniciais do processo de download de pacotes CVS do Horde. Em primeiro lugar digitar o seguinte comando:

```
# export CVSROOT=:pserver:cvsread@anoncvs.horde.org:/repository
```

Para em seguida digitar o comando:

```
# cvs login
```

Após este comando será pedido para entrar com a senha, que como já foi explicado é 'horde'. Isto feito, devemos primeiramente fazer o download dos pacotes "framework" e "imp", para aí então baixarmos o pacote "horde". Desta forma então, devemos ir até o diretório /var/www/html/horde/ e digitar:

```
# cvs co framework
```

Depois digitar:

```
# cvs co imp
```

Executado estes dois comandos voltamos para o diretório anterior /var/www/html/ com o comando:

```
# cd ..
```

E aqui digitamos:

```
#cvs co horde
```

Com o download dos três pacotes feito podemos ir até o diretório `/horde/framework/` e digitar o comando:

```
# php install-packages.php
```

Isto irá instalar todos os arquivos com data mais recente baixados. Após cada atualização dos módulos no CVS, você deverá logar no horde e ir no menu Administration->Configuration->Horde e gerar novamente o arquivo `conf.php`, copiar o conteúdo gerado e substituir pelo conteúdo do `conf.php` anterior em `/horde/config/`

6.5 Problemas e Soluções

Talvez a maior lição que possa ser tirada deste trabalho seja a que desmistifica o uso de softwares livre e o suporte aos problemas encontrados. No caso do presente trabalho, estamos trabalhando com uma ferramenta ainda em desenvolvimento, mas que mesmo assim apresenta uma série de avanços e disponibilidades que ainda não se vê presentes em outras soluções proprietárias. A questão da configuração, que certamente em versões CVS, trazem alguns problemas que ainda não estão documentados, pode ser facilmente solucionada através da lista oficial do IMP (imp@lists.horde.org), e o mais importante, com rapidez. Fazemos questão de ilustrar aqui dois exemplos de problemas encontrados durante a implantação deste trabalho e que foram rapidamente solucionados ou através da lista, ou através do "Bug Tracking", um dispositivo para serem enviados bugs(problemas) encontrados nas aplicações do Projeto Horde.

O primeiro problema encontrado, foi um bug que causava um erro ao tentar encaminhar ou responder uma mensagem no IMP:

```
Fatal error: Call to undefined function: cloneobject() in  
/var/www/html/horde/imp/compose.php on line 184
```

Para solucionar este erro foi gerado um chamado (ticket) no "Bug Tracking" na página do Horde (<http://www.horde.org>). O email enviado com a descrição do bug segue abaixo:

DO NOT REPLY TO THIS MESSAGE. THIS EMAIL ADDRESS IS NOT MONITORED.

-Ticket 59 -Queue: IMP -Created By:

-Summary: failed in reply and forward messages

Comment by juchem@inf.ufsc.br on Sun Apr 4 14:31:48 2004

(today): error message:

Fatal error: Call to undefined function: cloneobject() in /var/www/html/horde/imp/compose.php on line 184

imp HEAD 2004-04-03 downloaded from cvs

Eduardo

A resposta trazendo a solução veio no email abaixo:

DO NOT REPLY TO THIS MESSAGE. THIS EMAIL ADDRESS IS NOT MONITORED.

-Ticket 59 -Queue: IMP -New State: Bogus

-Summary: failed in reply and forward messages

Comment by Chuck Hagenbuch <chuck@horde.org>\$ on Sun Apr 4

18:43:52 2004 (today): Your Horde_Util package is out of date.

--

Gostaria de chamar a atenção para o fato de que o email foi mandado num domingo às 14:30, e a resposta veio no mesmo dia praticamente 4 horas mais tarde, às 18:40. Bastou seguir a instrução atualizando o pacote Horde_Util para resolver o problema.

Outro problema encontrado durante o desenvolvimento deste trabalho, foi com relação ao envio de mensagens assinadas digitalmente. Foi mandado um email para a lista oficial do IMP afim de procurar uma solução:

Hi,

I'm having the following problem with S/MIME signed messages in the CVS version of IMP:

I have a certificate installed in a computer at my College. I exported my key in the pkcs7 format, supported by IMP, and brought it to my computer at home. Inside Horde, in the menu Options->Mail->S\MIME Options, I could successfully import my key by importing the pkcs7 file. But when I try to sign a message with the option S/MIME Sign Message in the Encryption Options box, and try to send this message, I'm having the following error:

S/MIME Error: Need passphrase for personal private key.

S/MIME support is activated in prefs.php. Am I doing something wrong, or just missing something before importing the pkcs7 file?

My machine is: - Fedora Core 1 system; - PHP 4.3.4; - Httpd 2.0.48

Any help will be welcome Thanks in advance Edu

A resposta veio em seguida:

Zitat von Eduardo Juchem <juchem@inf.ufsc.br>:

> Hi,

>I'm having the following problem with S/MIME signed

>messages in the CVS

>version of IMP:

> I have a certificate installed in a computer at my

>College. I exported my key in the pkcs7 format, supported

>by IMP, and brought it to my computer at

>home. Inside Horde, in the menu

>Options->Mail->S\MIME Options, I could

>successfully import my key by importing the pkcs7
>file. But when i try to
>sign a message with the option S/MIME Sign
Message”>in the Encryption
>Options box, and try to send this message, i’m having
>the following error:
>S/MIME Error: Need passphrase for personal private >key.
>S/MIME suport is activated in prefs.php. Am i doing
>something wrong, or
>just missing something before importing the pkcs7 >file?
>Don’t you get a popup prompting for a passphrase? If not,
>make sure you allow unrequested popups in your browser for
>the site running IMP.
>Jan.

Novamente o tempo entre o envio do problema e o retorno com alguma solução ou sugestão não demorou muito. O email com a dúvida foi enviado à 1:45 de uma quarta-feira e a resposta chegou às 10:10 do mesmo dia. Esta dúvida enviada para a lista, aparentemente ocasionou uma modificação no código do IMP, de forma alertar o usuário que a janela de popup não pôde ser aberta, questionando se o usuário não está com o seu browser configurado para bloquear janelas popup. Esta mudança pode ser vista acessando o site <http://cvs.horde.org/cvs.php/imp/templates/javascript/> . Você visualizará a página abaixo:

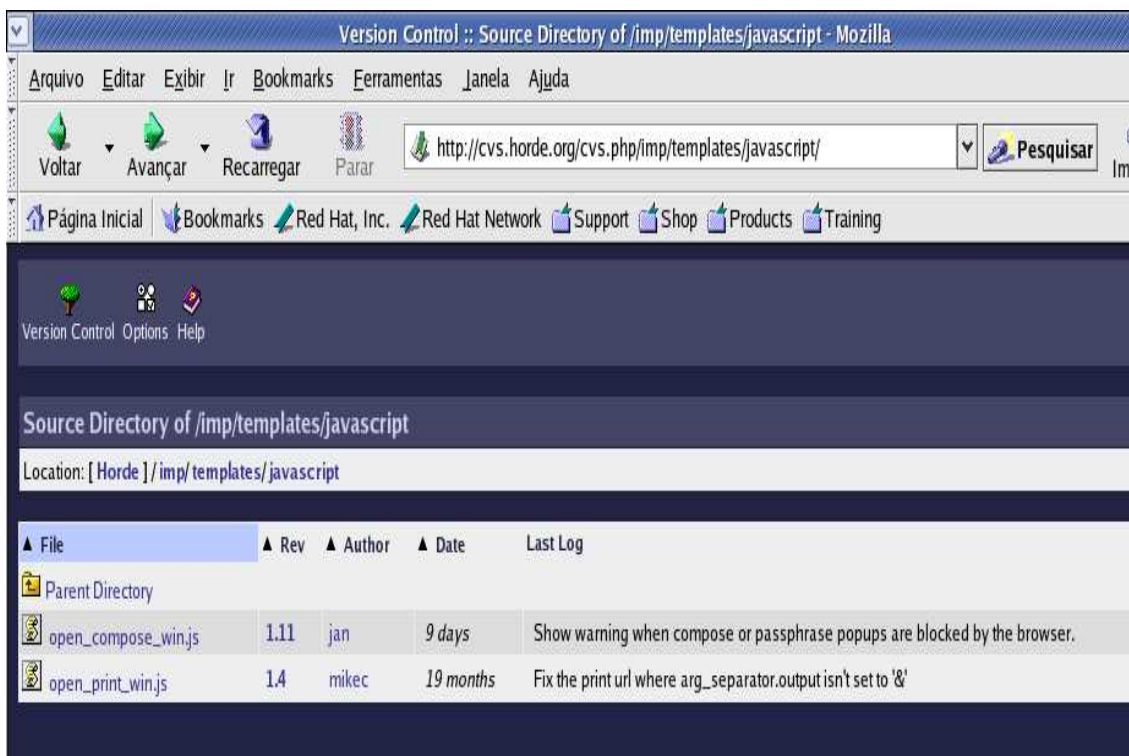


Figura 6.17: Resposta - Popup. Resposta do problema de configuração de popup.

Nesta página, clicando em `open_compose_win.js`, irá entrar em outra página com o histórico das modificações efetuadas neste arquivo:

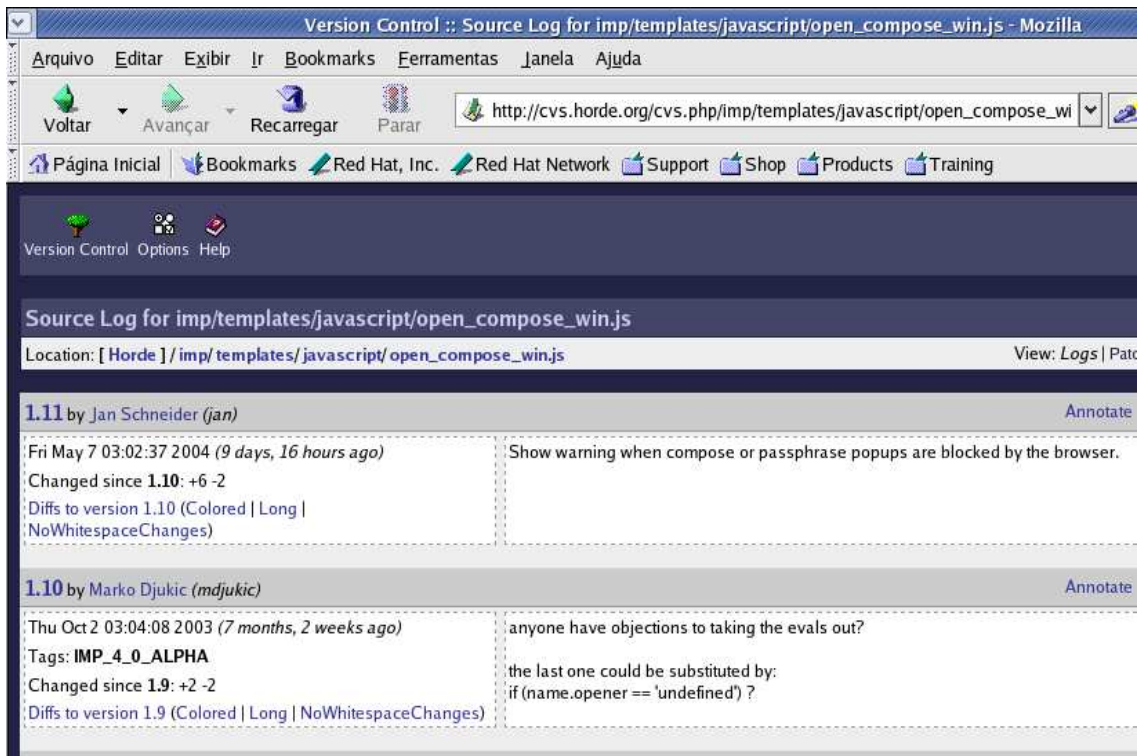


Figura 6.18: Histórico. Mostra o histórico de problemas resolvidos.

A alteração do dia 07 de maio de 2004, informa o que foi modificado, que neste caso, foi a inserção de um aviso ao usuário informando que a janela de popup não pôde ser aberta, solicitando que o mesmo verifique se seu browser não está bloqueando janelas popups. Clicando em "Colored" podemos ver as modificações feitas no código:

```

version 1.10                                     version 1.11
Line 27 function open_compose_win(args)          Line 27 function open_compose_win(args)
}
param =
"toolbar=no,location=no,status=yes,scrollbars=yes,resizable=yes,width="
+ Width + ",height=" + Height + ",left=0,top=0";
name = window.open(url, name, param);
if (!eval("name.opener")) {
    name.opener = self;
}
}
}
<?php if (!strstr($_SERVER['PHP_SELF'], 'javascript.php')); ?> -->

Legend:
Removed in v.1.10
changed lines
Added in v.1.11

Concluído

```

Figura 6.19: Código. Mostra as modificações feita no código.

6.6 Conclusão

O Webmail IMP está realmente um passo a frente em relação a seus concorrentes de código aberto. Particularmente a possibilidade de utilizar assinatura digital já faz do IMP um Webmail diferenciado, pois até o momento da conclusão deste trabalho não foi verificada a existência de outro Webmail que fosse de código aberto e que apresentasse esta funcionalidade. Isto levando em conta apenas a assinatura digital de emails.

Mas apesar de possuir este diferencial, a solução adotada no IMP para possibilitar a assinatura digital não é a mais desejável. No IMP, a chave privada do usuário precisa ser armazenada em sua área do banco de dados que fica no servidor. Desta maneira, o usuário está correndo o risco de ter sua chave privada corrompida no caso de uma invasão no sistema. O ideal seria que o usuário não se separasse de sua chave privada armazenando-a, por exemplo, em um smartcard. Desta forma o usuário poderia levar sua chave privada consigo para onde fosse, e a assinatura ou o deciframento seriam feitos na

máquina cliente, apenas com a presença do dispositivo que contivesse a chave privada, e não no servidor, que contém as chaves privadas de todos os usuários do sistema de webmail.

Apesar da solução adotada pelo IMP para o armazenamento das chaves pública e privada não ser a ideal, a segurança da chave privada e, por consequência, do sistema, ainda está resguardada. Isto porque na primeira mensagem de email que o usuário for usar sua assinatura digital, o IMP sempre irá solicitar que ele forneça a senha utilizada para a geração de sua chave privada.

Capítulo 7

Considerações Finais

Com a popularização e já tão necessária utilização de correio eletrônico para troca de informações entre pessoas, os sistemas de Webmail tornam-se ainda mais necessários que os clientes de email, por sua facilidade de acesso a qualquer computador de qualquer lugar do mundo. Existindo a necessidade de utilização surge os problemas em relação a segurança.

Desta forma, este projeto propõe uma garantia de autenticidade e integridade às mensagens de emails enviadas através de um sistema de Webmail seguro. Mostrando as vantagens e importância de se implantar um sistema de Webmail Seguro nas empresas.

Para implementar esta segurança o Webmail IMP utiliza o protocolo S/MIME. O S/MIME utiliza os certificados no padrão X.509 que dependem de uma autoridade certificadora, como a AC do LabSEC utilizada neste trabalho. No Brasil grande parte das autoridades certificadoras são subordinadas, controladas pela ICP Brasil (Infra-estrutura de Chave Pública do Brasil).

Quanto aos objetivos iniciais deste trabalho, eles foram alcançados, tendo como único problema remanescente a questão do armazenamento das chaves privadas pelo IMP. A solução adotada por este Webmail, de armazenar as chaves privadas no lado do servidor, e não na máquina cliente, não garante totalmente a segurança das chaves. Isto porque, por estarem armazenadas em um banco de dados, as chaves estão sujeitas a outras formas de ataque as quais não dependam da segurança oferecida por um protocolo seguro, como o

https. Citando apenas uma destas formas de ataque, a própria segurança física da máquina onde as chaves estão guardadas é uma questão a ser considerada, já que a única forma de se garantir a segurança de uma chave privada é que esta não saia de perto de seu dono. Sendo assim, a segurança oferecida pelo túnel seguro, garante sigilo apenas durante a conexão com o Webmail, mas a partir do momento que esta conexão é desfeita, as chaves permanecem guardadas no servidor.

7.1 Trabalhos Futuros

Um trabalho que pode ser desenvolvido futuramente é o desenvolvimento de uma tecnologia diferente da adotada pelo IMP na questão do armazenamento das chaves privadas. Juntamente com o desenvolvimento desta tecnologia, deverá se ter em vista a sua integração com o sistema de Webmail IMP, ou seja, a forma como o IMP irá interagir com o dispositivo que armazena as chaves privadas.

Outro trabalho sugerido é o estudo de uma solução para o processo de cifragem das mensagens eletrônicas no IMP. Tendo em vista que esta é uma versão ainda em desenvolvimento, algumas de suas funcionalidades ainda não estão implementadas por completo. Desta forma, juntamente com a equipe de desenvolvimento do IMP, pode-se propor uma solução para este problema.

Referências Bibliográficas

- [BRA] BRASIL, F. **O projeto Fedora Brasil**. Disponível em <<http://www.fedorabrasil.com.br/about.html>>. Acesso em: Maio, 2004.
- [BUR] BURNETT, S.; PAINE, S. **Criptografia e Segurança - Guia Oficial RSA**. 1. ed. ed. Rio de Janeiro - RJ: RSA Press.
- [CER] CERTISIGN. **CertiSign**. Disponível em <<http://www.certisign.com.br>>. Acesso em: Junho, 2004.
- [CN] CBPF-NT. **Webmail Seguro**. Disponível em <<http://www.projetederedes.kit.net/>>. Acesso em: Janeiro, 2003.
- [CUS] CUSTÓDIO, R. F. **Segurança em Computação**. Disponível em <<http://www.inf.ufsc.br/custodio/cursos/>>. Acesso em: Janeiro, 2003.
- [ICI] ICICEMAIL. **E-mail**. Disponível em <<http://limbo.ime.usp.br/mac339/index.php/IcicEMail>>. Acesso em: Agosto, 2003.
- [IET] IETF. **S/MIME Mail Security (smime)**. Disponível em <<http://www.ietf.org/html.charters/smime-charter.html>>. Acesso em: Dezembro, 2002.
- [IMP] IMP. **Internet Messaging Program**. Disponível em <<http://www.horde.org/imp>>. Acesso em: Dezembro, 2002.
- [LAB a] LABORATORIES, R. **Frequently Asked Questions About Today's Cryptography**. CD-ROM.
- [LAB b] LABSEC. **Laboratório de Segurança em Computação - Manual para Utilização do Certificado Digital**. Disponível em <<http://ac.labsec.ufsc.br/manual/index.htm>>. Acesso em: Maio, 2004.
- [OPE] OPENSSL. **The Open Source toolkit for SSL/TLS**. Disponível em <<http://www.openssl.org>>. Acesso em: Dezembro, 2002.

- [RED] REDHAT. **RedHat Project**. Disponível em <<http://www.redhat.com/>>. Acesso em: Maio, 2004.
- [RSA] RSA. **RSA Laboratories**. Disponível em <<http://www.rsasecurity.com/rsalabs/node.asp?id=2124>>. Acesso em: Maio, 2004.
- [RSO] RSOUTLOOK. **Como funciona o e-mail?** Disponível em <<http://pwp.netcabo.pt/rsoutlook/PT/email.htm>>. Acesso em: Agosto, 2003.
- [STA] STALLINGS, W. **S/MIME.In:Cryptography and Network Security-Principles and Practice**. 2. ed. ed. New Jersey: Prentice Hall.
- [WIR] WIRELESS, O. **O que é email?** Disponível em <<http://www.overnet.com.br/suporte/email.php>>. Acesso em: Agosto, 2003.