Web-Server Seguro:APACHE

Anderson Alves de Albuquerque aaa@cbpf.br
Marita Maestrelli marita@cbpf.br

CAT / CBPF

Prefácio

Em virtude dos diversos problemas de segurança que vem ocorrendo ultimamente, estamos documentando tecnicamente várias dicas sobre configurações seguras de servidores Web (webserver).

Atualmente, podemos acompanhar em várias publicações na mídia *underground* ataques a sites resultando na pixação de suas páginas na Web, denegrindo a imagem dessas instituições. Este documento mostra várias formas de configurar os ***webservers Apache***, e como introduzir suporte a **SSL**.

Todas as idéias aqui apresentadas podem ser adaptadas para qualquer ***webserver***, apenas devemos ter em mente aspectos fundamentais sobre segurança de sistemas.

Para uma idéia inicial sobre segurança na WEB, veja a primeira nota técnica editada pelo CBPF, de outubro de 2000, CBPF-NT-008/00.

Índice:

1. Introdução 2

Apache 2

2. Instalação e configuração do Apache 3

2.1. Sistema operacional e software usado 3

2.2. Requisitos para instalar o servidor WWW 3

2.3. Descompactação do software 3

2.4. Processo de Instalação 4

2.5. Configuração do servidor WWW 6

2.6. Segurança do servidor WWW 11

2.7. Execução do servidor WWW 20

2.8. Execução do apache com chroot 22

2.9. Apache com SSL 24

3. Outras Soluções existentes no mercado 32

4. Bibliografia 33

5. Glossário 36

1 - Introdução

O http surgiu no final da década de 80, proposto por Tim Berners-Lee – CERN, e desde essa época o seu crescimento e os recursos agregados aumentaram exponencialmente.

O serviço HTTP[5.42] desfruta ao lado do SMTP[5.72] como os mais utilizados na internet[5.45] e das intranets[5.45], também possuindo grande utilização nos sistemas de e-commerce(comércio eletrônico). O http graças a sua versatilidade possibilita aos usuários participarem de chats(bate papo interativo), enviar e_mail(mensagens eletrônicas), interagir com Banco de dados, participar de ensino a distância, acesso a *wap http* e Homepages[5.40] interativas. Neste documento abordaremos o Apache que atualmente é o mais utilizado na Internet.

Não enfocaremos os recursos extras que envolvem o servidor http, como: *cgi's*, *Java script*, *Flash*, Banco de Dados, *cookies*, *java applets*, softwares de edições gráficas e outros.

Este procedimento tem como objetivo demonstrar a instalação, configuração e administração do serviço WWW (World Wide Web)[5.84].

Muitas dessas ferramentas merecem cuidados especiais pôr serem muito explorada na execução de um ataque, tanto do lado do *web server* como do cliente (*Browser*[5.10]).

Apache:

Atualmente o servidor apache é o mais utilizado na internet, aproximadamente 55%, entre os tópicos responsáveis por sua popularidade podemos destacar: a sua instalação e configuração é muito simples, suporte a SSL, suporte a cgi's, suporte a banco de dados, grande suporte técnicos em listas de discussão, estabilidade, escalabilidade, segurança quando bem configurado, suportada por diversas plataformas (Linux, *BSD, Solaris, IRIX, Digital UNIX, AIX, IBM OS/2, SCO, HPUX, Windows NT e outros).

O apache é escrito em C ANSI, é um software open source livremente distribuido no site www.apache.org pela "The Apache Software Foundation" onde diversos voluntários contribuem intelectualmente para o "projeto apache". Devemos respeitar a licença do apache que esta em: http://www.apache.org/LICENSE.txt.

No site do apache podemos encontrar diversas informações como: relações e correções de Bugs, informações sobre cgi's, informações sobre banco de dados, informações sobre SSL e outros.

2 - Instalação do *webserver*

2.1 - Sistema operacional e software usado

Sistema operacional UNIX FreeBSD 2.2.8-RELEASE 32 bits. Usaremos o software apache_1_3_12 versão 1.3.12 .

Devemos lembrar que este webserver possui versões para windows 95/98 e windows NT.

2.2 - Requesitos para instalação do Apache

- Compilador gcc (ANSI), versão 2.7.2.1 ou superior e o GNU make;
- Compactadores tar, gzip e gunzip;
- Comandos básicos como: chmod, chown, chgrp e vi;
- Conhecimento básico de UNIX ;
- Espaço em disco de 12MB para temporários e 3Mb para binários;
- Perl 5 (opcional).

2.3 - Descompactação do software

- Inicialmente estar no diretório onde encontra –se o software compactado:

```
$ cd /usr/local/
```

- Usar o compactador “tar” existente no UNIX

```
$ tar -xvzf apache_1_3_12_tar.tar
$ chown –R 0 apache_1.3.12
$ chgrp –R 0 apache_1.3.12
```

- Entrar no diretório onde o software foi descompactado

```
$ cd apache_1.3.12
```

2.4 - Processo de Instalação

- Executar o “script” que preparará o software de acordo com o ambiente do sistema operacional:

```
./configure  \
--prefix=/opt/apache \
--exec-prefix=/opt/apache \
--sbindir=/usr/sbin \
--bindir=/usr/bin \
--libexecdir =/usr/libexec \
--mandir =/usr/man \
--includedir=/usr/include \
```

A instalação do apache será feita no em /opt/apache.
As bibliotecas serão instaladas em /usr/libexec.
Os *includes* serão instalados em /usr/include.
Os manuais serão instalados em /usr/man.
Os binários foram instalados em /usr/bin e /usr/sbin.

- Iniciar o processo de compilação:

$ make

- Iniciar o processo de instalação:

$ make install

- Cuidados especiais com o diretório **cgi-bin**

Remova os arquivos **printenv** e **test-cgi** em /opt/apache/cgi-bin/ , eles normalmente não são necessários e podem oferecer riscos de segurança.

- Esta listagem mostrará a localização e permissões dos arquivos que compõem o software do servidor apache.

Diretório /opt/apache/conf:

O diretorio **conf** deve ter permissão 740 e os arquivos 640, como a seguir:

```
-rw-r-----  1 root    root         348 May  5 03:13 access.conf
```
(contém as diretrizes de controle de acesso do servidor web)

```
-rw-r-----  1 root    root         348 May  5 03:13 access.conf.default
-rw-r-----  1 root    root       31442 May  5 03:32 httpd.conf
```
(contém as diretrizes de operação do servidor Web, atributos de servidor)

```
-rw-r-----  1 root    root       31441 May  5 03:13 httpd.conf.default
-rw-r-----  1 root    root       12441 May  5 03:13 magic
-rw-r-----  1 root    root       12441 May  5 03:13 magic.default
-rw-r-----  1 root    root        9957 May  5 03:13 mime.types
```
(contém as diretrizes sobre os tipos de documentos no servidor Web, tipos MIME reconhecidos)

-rw-r----- 1 root root 9957 May 5 03:13 mime.types.default
-rw-r----- 1 root root 357 May 5 03:13 srm.conf
(contém as diretrizes de controle de documentos do servidor web, recursos do sistema)

-rw-r----- 1 root root 357 May 5 03:13 srm.conf.default
(Recursos do sistema)

Diretório /opt/apache/htdocs/:

O diretorio **htdocs** deve ter permissão 755 e os arquivos 644.
Neste arquivo estará armazenada toda Homepage do Web Server.

Diretório /opt/apache/icons:

O diretorio **icons** deve ter permissão 755 e os arquivos 644.
Neste arquivo estará armazenada os icones usados no alias /icons/ na configuração do Web Server.

Diretório /opt/apache/logs:

O diretorio **logs** deve ter permissão 740.
Neste diretório serão armazenados os logs gerados pelo web server.

-rw-r----- 1 root root 1191 May 5 03:40 access_log
(logs de acesso ao web server)

-rw-r----- 1 root root 878 May 5 03:39 error_log
(logs de erros ocorridos, normalmente devido a acessos indevidos)

-rw-r----- 1 root root 5 May 5 03:39 httpd.pid
(armazena o numero do processo “PID” do daemon httpd)

Diretório /opt/apache/proxy:

O diretorio **proxy** deve ter permissão 740.

Arquivo /usr/sbin/httpd :

Este deve ter a permissão 750.

2.5 - Configuração do servidor WWW

- Sugestão de configuração para o /opt/apache/conf/httpd.conf.

Para configurar o arquivo **httpd.conf** faremos algumas considerações importantes:

IP do Servidor web	10.10.10.10
Máscara de rede	255.255255.0
Porta de comunicação	80/tcp e 80/udp
Nome do servidor de Web	www.mydomain.com.br
Host Virtual	www.hostvirtual.mydomain.com.br

Abaixo exemplificaremos a configuração do arquivo /opt/apache/conf/httpd.conf :

ServerType standalone|inetd
(modo de execução do servidor standalone ou através do inetd)

ServerRoot "/opt/apache/"
(Raiz do servidor)

PidFile logs/httpd.pid
(Path do log do PID, processo do daemon httpd, apartir da raiz do server)

ScoreBoardFile logs/apache_runtime_status
(usado para armazenar processos internos do web server)

#ResourceConfig conf/srm.conf
(Path do arquivo srm.conf)

#AccessConfig conf/access.conf
(Path do arquivo access.conf)

Timeout 300
(tempo em segundos de time out do web server)

KeepAlive On|off
(permite ou não persistentes tentativas de conexões ao web server)

MaxKeepAliveRequests 100
(máximo de conexões permitidas durante uma persistente tentativa de conexão)

KeepAliveTimeout 15
(tempo em segundos para esperar a próxima requisição)

MinSpareServers 5

MaxSpareServers 10

StartServers 5
(número de servidores startados)

MaxClients 150

(número máximo de conexões simuntaneas)

MaxRequestsPerChild 30
(número máximo de requisições por processo)

Port 80
(porta aberta para conexão)

Listen 10.10.10.10:80
(IP onde será startado o web server, opcional)

<IfDefine SSL>
Listen 80
Porta usada pelo http tradicional
Listen 443
Porta usada pelo SSL (https)
</IfDefine>
(quando utiliza –se o suporte para SSL define –se a porta HTTP e em seguida a porta HTTPS, conforme mostrado)

User nobody
(user que executará o web server)

Group nogroup
(grupo que executará o web server)

ServerAdmin webmaster@mydomain.com.br
(e-mail do responsável pelo web server)

ServerName www.mydomain.com.br
(Host do webserver definido no dns server)

DocumentRoot "/opt/apache/htdocs"
(diretório onde encontraremos as Home Pages do web server, todo o site)

Options Indexes FollowSymLinks
Options SymLinksIfOwnerMatch
(evita uso de links simbólicos para navegação fora da árvore de diretórios de Home Pages)

Options IncludesNoExec
(Evita a execução de comandos arbitrários)

<Directory />
  Options FollowSymLinks
  AllowOverride None
</Directory>
<Directory "/opt/apache/htdocs">
(inicia sessão para o diretório /opt/apache/htdocs)

Options Indexes FollowSymLinks
AllowOverride None
(evita uso de links simbólicos para navegação fora da árvore de diretórios de Home Pages)

Order allow,deny
Allow from all
</Directory>
(fecha a sessão para o diretório /opt/apache/htdocs)

UserDir disabled
(Desabilita a existência de um diretório home de usuário para Hospedagem de Home Pages)

DirectoryIndex index.html index.htm default.htm default.html
(define os arquivos index.html, index.htm, default.htm e default.html como documentos default)

AccessFileName .htaccess
(define o arquivo para controle de acesso)

<Files .htaccess>
Order allow,deny
Deny from all
</Files>
(não permite que visitantes vejam o conteúdo do arquivo .htaccess)

TypesConfig conf/mime.types
(path do arquivo MIME)

DefaultType text/plain
(Indica tipo e subtipo MIME default)

HostnameLookups On|Off
(loga os clientes pelo Hostname ou Ip, respectivamente)

ErrorLog logs/error_log
(define o arquivo de log de erros)

LogLevel warn
(nível warn para os logs)

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
(define o formato dos erros que serão logados)

CustomLog logs/access_log common

```
ServerSignature On|Off| EMail
( mostra ou não informações do web server quando ocorre um erro)

Alias /icons/ "/opt/apache/icons/"
(define o alias para /icons)

<Directory "/opt/apache/icons">
   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
(define as permissões para o alias /icons)

ScriptAlias /cgi-bin/ "/opt/apache/cgi-bin/"
( define o alias para /cgi-bin)

<Directory "/opt/apache/cgi-bin">
   AllowOverride None
   Options None
   Order allow,deny
   Allow from all
</Directory>
(define as permissões para o alias /cgi-bin)

DefaultIcon /icons/unknown.gif
(icone default)

IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t  *
(não listas estes arquivos)

ErrorDocument 404 /missing.html
(acessa o arquivo missing.html caso ocorra o erro 404)

NameVirtualHost  www.mydomain.com.br
(define o host virtual www.mydomain.com.br)

NameVirtualHost www.hostvirtual.mydomain.com.br
(define o host virtual www. Hostvirtual.mydomain.com.br)

<VirtualHost 10.10.10.10>
   ServerAdmin webmaster@ mydomain.com.br
   DocumentRoot /htdocs/
   ServerName www.mydomain.com.br
   ErrorLog logs/ www.mydomain.com.br-error_log
   CustomLog logs/ www.mydomain.com.br-access_log common
</VirtualHost>
(define o ambiente para www.mydomain.com.br)
```

<VirtualHost 10.10.10.10>
ServerAdmin webmaster@ mydomain.com.br
DocumentRoot /htdocs/
ServerName www.hostvirtual.mydomain.com.br
ErrorLog logs/ www.hostvirtual.mydomain.com.br-error_log
CustomLog logs/ www.hostvirtual.mydomain.com.br-access_log common
</VirtualHost>
(define o ambiente para www.hostvirtual.mydomain.com.br)

- Alguns recursos extras que não foram mostrados acima:

IndexIgnore *.exe *.com *
(arquivos que não devem ser mostrados)

HeaderName File
(o conteúdo do arquivo é mostrado no início da listagem)

ReadmeName File
(o conteúdo do arquivo é mostrado no final da listagem)

IndexOptions FancyIndexing
(mostra nas listagens o Ícones e tamanhos do arquivo)

AddDescription "Ärquivo executável" *.exe
(Descrição do arquivo)

DefaultIcon /icon/default.gif
(ícone para arquivo sem descrição)

AddType image/gif .gif
(Indica tipo e subtipo para uma extensão)

DefaultType text/html
(define um tipo e subtipo para arquivos não definidos)

AddEncoding x-giz gz
(codificação MIME para o orquivo)

2.6 - Segurança do servidor WWW

- User e Group:

Verificar se o user *nobody* e o grupo *nogroup* geram os processos **http** do servidor Web. Estes *users* devem ter o campo de senha com "*****" e um *shell nologin*.

Devemos nos certificar que os diretórios e arquivos em /opt/apache/ tenham como donos o **user root** e **group root**, evitando deixa–los como dono o *nobody* e *nogroup*.

- Desabilitar a listagem de arquivos:

IndexIgnore *.exe *.com * *

- Evitar links simbólicos fora da árvore de diretórios:

Options Indexes FollowSymLinks
Options SymLinksIfOwnerMatch

- Ativando senhas

No arquivo de configuração httpd.conf

```
<Directory /opt/apache/htdocs/docs>
# Define o diretório que Terá acesso com senha
AuthName Exemplo
# Label que aparecerá na janela de senha
AuthType Basic
# Tipo de autenticacação
AuthUserFile /opt/apache /conf2/passwd
# Diretório e arquivo onde estará a senha encriptada
<Limit GET POST>
order deny,allow
allow from all x.y.z.*
# Ips que podem acessar o diretório
deny from *.*.*.*
# Ips que não podem acessar o diretório
# require valid-user
# Requer usuários válidos "/opt/apache /conf2/passwd"
</Limit>
</Directory>
<Directory /opt/apache/htdocs/servers/>
# Localização do segundo diretório com acesso através de senhas
AuthName Area_Restrita
AuthType Basic
AuthUserFile /opt/apache/conf2/passwdservers
# Localização do segundo passwd com senhas e usuários
<Limit GET POST>
order deny,allow
allow from all x.z.w.*
# Ips permitidos ao diretório "/opt/apache/conf2/passwdservers"
deny from *.*.*.*
```

```
# Ips com acesso negados
require valid-user
# requer user válido
</Limit>
</Directory>
```

Os arquivos de senha "/opt/apache/conf2/passwdservers" e "/opt/apache /conf2/passwd" devem ter pelo menos permissão 555 e a sintaxe deles são:

```
User:senha_encriptada
João:zb,Kfwdf8o9S
```

A senha encriptada pode sequir o padrão encontrado no *shadow* ou *master.passwd* do sistema. Existem opções onde podemos usar o prórpio arquivo(s) de senha(s) do sistema, em vez de criar outro como mostrado acima.

Para cada diretório declarado acima no httpd.conf é necessário uma configuração extra no diretório protegido dentro de um arquivo chamado **.htaccess**

Em /opt/apache/htdocs/servers/.htaccess:

```
AuthName        Area_Restrita
AuthType        Basic
AuthUserFile    /opt/apache/conf2/passwdservers
# Localização do arquivo de senhas
<Limit GET POST>
require valid-user
# Requer users válidos
</Limit>
```

Em /opt/apache/htdocs/docs/.htaccess:

```
AuthName        Area_Restrita
AuthType        Basic
AuthUserFile    /opt/apache/conf2/passwd
# Localização do arquivo de senhas
<Limit GET POST>
require valid-user
# Requer users válidos
</Limit>
```

- Não podemos esquecer de declarar no Arquivo de configuração httpd.conf a linha:

AccessFileName .htaccess

- Tratamento de erros

No Arquivo de configuração httpd.conf podemos declarar códigos de erros e Home Page que será mostrada na ocorrência deste erro.

```
ErrorDocument  403  http://x.y.z.w/Documentos_err/403.htm
# Erro 403
# Home Page mostrada na ocorrência do erro 403:
# http://x.y.z.w/Documentos_err/403.htm
ErrorDocument  400  http://x.y.z.w/Documentos_err/400.htm
# Erro 400
# Home Page mostrada na ocorrência do erro 400:
# http://x.y.z.w/Documentos_err/400.htm
ErrorDocument  402  http://x.y.z.w/Documentos_err/402.htm
# Erro 400
# Home Page mostrada na ocorrência do erro 400:
# http://x.y.z.w/Documentos_err/400.htm
ErrorDocument  501  http://x.y.z.w/Documentos_err/501.htm
# Erro 501
# Home Page mostrada na ocorrência do erro 501:
# http://x.y.z.w/Documentos_err/501.htm
ErrorDocument  302  http://x.y.z.w/Documentos_err/302.htm
# Erro 302
# Home Page mostrada na ocorrência do erro 302:
# http://x.y.z.w/Documentos_err/302.htm
```

- Esconder a versão do apache

Em **httpd.conf** podemos inibir a exibição da Home Page de erro onde é mostrado o nome do servidor, versão e *e-mail* do responsável:

ServerSignature Off

Ou

Em: src/include/httpd.h
Edite a linha:
#define SERVER_BASEVERSION " version not define "

- Desabilitando o Server Side Includes (SSI):

Evitaremos execução de comandos arbitrários.

Options IncludesNoExec

- Protegendo os arquivos:

```
<Directory />
        Order deny,allow
         Deny from all
</Directory>
```

Evitando assim ataques do tipo:

```
$cd /; ln -s / public_html
$ lynx http://localhost/~root
```

Isso fará com que o usuário navegue normalmente pelo diretório raiz do servidor.

Além disso podemos desabilitar que usuários tenham acesso a Home Pages em seus diretórios, com a seguinte linha:

UserDir disabled

- Evitar que as configs dos users prevaleçam sobre a do server:

```
<Directory />
AllowOverride None
Options None
allow from all
</Directory>
```

- Limitando acesso por ip:

```
<Directory /opt/apache/htdocs/docs>
  <Limit GET POST>
    order mutual-failure
    deny from all
    allow from x.y.z.33
# Somente o ip x.y.z.33 pode acessar o
# diretório /opt/apache/htdocs/docs
  </Limit>
```

```
</Directory>
```

- CGI's:

Nos scripts CGI's devemos evitar entradas ilegais na URL, execução de comandos arbitrários, ocorrências de caracteres e palavras sutis, como:

",” ; “../../” ; passwd; master.passwd; shadow; “/etc/”; “/var/mail”; “/var/log”; “>”; “<”; “|”; “/home/”; “/bin/sh” ou qualquer shell, “/bin/ls” ou comandos arbitrários e etc....

Também devemos usar caminhos absolutos para o script, permissões 755 (wrx-rx-rx), e desabilitar ou deletar scripts instalados por default no webserver em “/cgi-bin”. Devemos também Ter uma política de controle burocrático dos cgi's, pois em sites que utilizam muito os cgi's é comum webmasters esquecerem ativos cgi's que não estão mais sendo usados pelo webserver.

Devemos também colocar os softwares cgi's em diretórios pré-determinados, como: “~/cgi-bin/”.

Devemos tomar cuidado especial:

- Arquivos .htaccess (com permissões de acesso);
- arquivos de senhas de acesso do webserver declarados em AuthUserFile, estes devem estar em locais onde o cliente tenha acesso, tirando –o dos diretórios subordinados ao “Documento root”
- Verificar permissões constantemente, evitando erros grosseiros como: 777, 757, 775 e etc....
- Verificar constantemente a existências de setuid, setgid e sticky:

```
$ find  /opt/apache/  -perm -04000 -print
(setuids)
$ find  /opt/apache/  -perm –02000 -print
(setgids)
$ find /opt/apache/  -perm -01000 –print
(stickys)
$ find /opt/apache/  -name  “core” –print
(core)
```

- Outro cuidado especial é criar um checksum com dos arquivos em:

Podemos com isso detectar através de alarmes instalações de backdoors e alterações não permitidas.

Arquivo /usr/sbin/httpd

Arquivo /usr/sbin/apachectl

Diretório /opt/apache/conf e os seus sub-diretórios e arquivos :
Arquivo /opt/apache/conf /access.conf
Arquivo /opt/apache/conf /access.conf.default
Arquivo /opt/apache/conf /httpd.conf
Arquivo /opt/apache/conf /httpd.conf.default
Arquivo /opt/apache/conf /magic
Arquivo /opt/apache/conf /magic.default
Arquivo /opt/apache/conf /mime.types
Arquivo /opt/apache/conf /mime.types.default
Arquivo /opt/apache/conf /srm.conf
Arquivo /opt/apache/conf /srm.conf.default

Diretório /opt/apache/cgi-bin e os seus sub-diretórios e arquivos:
Arquivos /opt/apache/cgi-bin/*

Para isso existem ferramentas e programas como: **md5**, **tripwire**, **mtree**, **aid**e, **sha1** e **rmd160**. Neste *checksum* não podemos esquecer de modificar a base de dados com assinatura digital, após as mudanças realizadas, evitando receber alarmes falsos.

- Podemos em /etc/fstab colocar propriedades nosuid, nodev, noexec e ro nas partições, exemplo:

Uma partição /opt/apache/ com nosuid e nodev;
(evitar a execução de suids)

Uma partição /opt/apache/conf/ com nosuid, nodev, noexec, ro;
(evitar a execução de suids, execução de arquivos e manter a partição como read only)

Uma partição /opt/apache/logs/ com nosuid,nodev;
(evitar a execução de suids)

Uma partição /opt/apache/cgi-bin/ com nosuid, nodev, ro e caso não seja usado cgi's podemos acrescentar noexec;
(evitar a execução de suids e manter a partição como read only)

Uma partição /opt/apache/htdocs/ com nosuid,nodev.
(evitar a execução de suids)

Nos **BSD**'s ou **Linux** , podemos ainda habilitar os *flags* de imutabilidade para ~/conf/ e ~/cgi-bin/. Assim podemos evitar a instalações de *backdoors* e alterações não permitidas. No diretório ~/logs/ podemos ativar o flag append only (sappnd) nos BSD's, mas na podemos esquecer de adaptar o scripts /etc/daily, /etc/weekly e /etc/monthly que realizam rotação dos logs.

- Podemos filtrar ocorrências nos logs:

Algumas dessas ocorrências podem caracterizar tentativas de ataques. Devemos analisar pois essas ocorrências podem ser acessos normais em alguns sistemas, dependendo apenas dos recursos usados e configurações do administrador.

```
$ cat /opt/apache/logs/* |grep "cgi-bin"
$ cat /opt/apache/logs/* |grep "passwd"
$ cat /opt/apache/logs/* |grep "shadow"
$ cat /opt/apache/logs/* |grep "master.passwd"
$ cat /opt/apache/logs/* |grep "etc"
$ cat /opt/apache/logs/* |grep "/home/"
$ cat /opt/apache/logs/* |grep "bin/"
$ cat /opt/apache/logs/* |grep "var/"
$ cat /opt/apache/logs/* |grep "usr/"
$ cat /opt/apache/logs/* |grep "dev/"
$ cat /opt/apache/logs/* |grep "passwd"
$ cat /opt/apache/logs/* |grep "root"
$ cat /opt/apache/logs/* |grep ".exe"
$ cat /opt/apache/logs/* |grep ".dll"
$ cat /opt/apache/logs/* |grep "c:\"
$ cat /opt/apache/logs/* |grep "tmp/"
$ cat /opt/apache/logs/* |grep ":\"
$ cat /opt/apache/logs/* |grep ">"
$ cat /opt/apache/logs/* |grep "<"
$ cat /opt/apache/logs/* |grep "../.."
$ cat /opt/apache/logs/* |grep "?"
$ cat /opt/apache/logs/* |grep "phf"
$ cat /opt/apache/logs/* |grep "/bin/cat"
$ cat /opt/apache/logs/* |grep "172.16."
$ cat /opt/apache/logs/* |grep "192.168."
$ cat /opt/apache/logs/* |grep "10."
```

- Desabilitar o uso "~/cgi-bin/":

Caso não seja necessário usar cgi's podemos desabilitar "~/cgi-bin/" comentando a linha:

ScriptAlias /cgi-bin/ "/opt/apache/cgi-bin/"

- Iniciar o script **perl**:

O *Taint Mode* ativado verifica a entrada dos arquivos antes de usa –los.

#!/usr/local/bin/perl -T

- Exemplos CGI's explorados:

CGI	**Versões Afetadas**
Count.cgi	1.0-2.3
webdist.cgi	1.0-1.2
php.cgi	Até 2.0
nph-test.cgi	ALL
AnyForm	1.0
FormMail	1.0
phf	ALL
Big Brother	N.D.
nph-publish	1.0-1.1

N.D.: Não disponível

Exemplo de ocorrências de explorações em webservers, estas ocorrências podem ser filtradas nos logs ou em *sniffers* (IDS):

CGI's	***URL (exploração)***
PHF CGI access	/cgi-bin/phf
PHP CGI access	/cgi-bin/php.cgi
Webgais CGI access	/cgi-bin/webgais
Websendmail CGI access	/cgi-bin/websendmail
Webdist CGI access	/cgi-bin/webdist.cgi
Htmlscript CGI access	/cgi-bin/htmlscript
CGI pfdisplay access	/cgi-bin/pfdisplay.cgi
Cgichk Pfdispaly (sic) access	/cgi-bin/pfdispaly.cgi
CGI Perl access	/cgi-bin/perl.exe
Wwwboard CGI access	/cgi-bin/wwwboard.cgi
WWW-SQL CGI access	/cgi-bin/www-sql
Guestbook CGI access	/cgi-bin/guestbook.cgi
CGI Man access	/cgi-bin/man.sh
CGI view-source access	/cgi-bin/view-source?../../../../../../../etc/passwd
Finger CGI access	/cgi-bin/finger
Campas CGI access	/cgi-bin/campas
NPH CGI access	/cgi-bin/nph-test-cgi

rwwwshell CGI access	/cgi-bin/rwwwshell.pl
NPH-publish CGI access	/cgi-bin/nph-publish
Aglimpse CGI access	/cgi-bin/aglimpse
Glimpse CGI access	/cgi-bin/glimpse
AT-admin CGI access	/cgi-bin/AT-admin.cgi
Filemail CGI access	/cgi-bin/filemail.pl
JJ CGI access	/cgi-bin/jj
Maillist CGI access	/cgi-bin/maillist.pl
Info2www CGI access	/cgi-bin/info2www
Files CGI access	/cgi-bin/files.pl
Bnbform CGI access	/cgi-bin/bnbform.cgi
Survey CGI access	/cgi-bin/survey.cgi
AnyForm CGI access	/cgi-bin/AnForm2
Textcounter CGI access	/cgi-bin/textcounter.pl
Classifieds CGI access	/cgi-bin/classifieds.cgi
Environ CGI access	/cgi-bin/environ.cgi
Wrap CGI access	/cgi-bin/wrap
Cgiwrap CGI access	/cgi-bin/cgiwrap
Edit CGI access	/cgi-bin/edit.pl
Perlshop CGI access	/cgi-bin/perlshop.cgi
Args CGI access	/cgi-dos/args.bat
Upload CGI access	/cgi-win/uploader.exe
Rguest CGI access	/cgi-bin/rguest.exe
Wguest CGI access	/cgi-bin/wguest.exe
IIS vti_inf access	/_vti_inf.html
IIS service.pwd access	/_vti_pvt/service.pwd
IIS users.pwd access attempt	/_vti_pvt/users.pwd
IIS authors.pwd access	/_vti_pvt/authors.pwd
IIS administrators.pwd access	/_vti_pvt/administrators.pwd
IIS shtml.dll access	/_vti_pvt/shtml.dll
IIS shtml.exe access	/_vti_pvt/shtml.exe
IIS Codebrowser access	/iissamples/exair/howitworks/codebrws.asp
IIS Openfile access	/cfdocs/expelval/openfile.cfm
IIS Exprcalc access	/cfdocs/expelval/exprcalc.cfm
IIS Displayopenedfile access	/cfdocs/expelval/displayopenedfile.cfm
IIS Sendmail access	/cfdocs/expelval/sendmail.cfm
IIS Search97 access	/search97.vts
IIS Carbo.dll access	/carbo.dll
IIS Codebrowser access	/iissamples/sdk/asp/docs/codebrws.asp
IIS Showcode access	/msads/Samples/SELECTOR/showcode.asp
IIS Bdir access attempt	/scripts/iisadmin/bdir.htr
IIS CGImail access	/scripts/CGImail.exe
IIS NewDSN access	/scripts/tools/newdsn.exe
IIS Fpcount access	/scripts/fpcount.exe

Extraído do sniffer snort http://www.clark.net/~roesch/security.html

- Acessos ao servidor web para manutenção de Home Pages configurações:

Os acessos ao servidor web devem ser realizado através de um canal encriptado usando ferramentas como o ssh e túneis ssl. Evitando usar acessos através dos serviço: ftp, telnet, rlogin, nfs e outos.

2.7 - Execução do servidor WWW

- Como inicializar o Apache:

Existem duas formas de carregar o web server: **standalone** ou **inetd**, definidas em ServerType no arquivo httpd.conf.

Forma stand alone:

```
# ------------  inicio do script -----------------
#!/bin/sh
#   start apache
if [ ! -d /usr/sbin/apachectl ]
then
/usr/sbin/apachectl  start &
fi
# ------------  fim  do script -----------------
```

Ou poderiamos fazer assim:

```
# ------------  inicio do script -----------------
#!/bin/sh
#   start apache
if [ ! -d /usr/sbin/httpd ]
then
/usr/sbin/httpd –f  /opt/apache/conf/httpd.conf &
fi
# ------------  fim  do script -----------------
```

- no freebsd podemos adicionar este script em /etc/rc.local

No /etc/services devemos colocar a linha:

```
www          80/tcp      # http
www          80/udp      # http
```

Iniciar o apache no /etc/inetd.conf:

Basta adicionar esta linha:

```
www   stream  tcp  nowait nobody /usr/sbin/httpd –f  /opt/apache/conf/httpd.conf
```

Para reiniciar o apache quando iniciado pelo inetd:

$ ps xa | grep inetd
4993 ?? Ss   0:00.03 inetd
(neste exemplo o número 4993 apresenta o valor do processo ,PID ,do daemon inetd)

$ kill –HUP 4993

Para reiniciar o apache quando iniciamos como standalone:

$ apachectl restart

Para terminar a execução do apache quando iniciado pelo inetd:

$ ps xa | grep inetd
4993 ?? Ss   0:00.03 inetd
(neste exemplo o número 4993 apresenta o valor do processo ,PID ,do daemon inetd)

$ kill –9 4993

Para terminar a execução do apache quando iniciamos como standalone:

$ apachectl stop

Comandos importantes do *daemon httpd*:

-f <conf>
(lê o arquivo de configuração designado)

-v
(mostra a versäo do httpd)

-l
(lista todos os modulos copilados no httpd)

-T
(testa as configurações do web server)

-h
(imprime o help do web server)

Comandos importantes do *apachectl*:

start	Startar o httpd
Startssl	Startar o httpd com SSL

Stop	Para a execução do httpd
Restart	Reiniciar o httpd
Configtest	Faz um teste nos arquivos de configuração
Help	Mostra um pequeno help

2.8 - Execução do Apache com chroot[5.17]

O *chroot* é um binário que evita a subida na árvore de diretórios, enjaulando o *daemon* criando assim uma "maquina virtual". Isso é parecido com o que acontece no ftp anonymous.

$ chroot /nova/raiz_do_server executável

Isso minimiza o impacto de *bugs* intrínsecos no *daemon httpd* , e falhas de programação CGI. A segurança do servidor web aumenta muito com a utilização do *chroot*, com isso em tese um invasor so terá acesso a nova raiz da "máquina virtual" caso invada o servidor web.

O servidor Apache deve ser inicializado:

$ chroot /opt/apache/ /usr/sbin/httpd –f /conf/httpd.conf &

Neste exemplo, escolhemos a nova raiz como /opt/apache/ e daemon em /usr/sbin/httpd, já o arquivo de configuração esta apontando para /conf/httpd.conf porque o httpd passará a enchegar a raiz em / em /opt/apache/

Não podemos esquecer de alterar o httpd.conf, modificando todos os paths substituindo /opt/apache/ por / .
Alguns Exemplos:

De: ServerRoot "/opt/apache/"
Para: ServerRoot "/"

De: DocumentRoot "/opt/apache/htdocs"
Para: DocumentRoot "/htdocs"

De: Alias /icons/ "/opt/apache/icons/"
Para: Alias /icons/ "/icons/"

Verificar todas as diretrizes, como: <Directory> e logs
O CGI's Também tem que Ter seus paths absolutos modificados.

Outros parâmetros com diretórios, devem ser modificados ,para poder carregar o novo *path*.

Quando iniciarmos o servidor httpd com o chroot , apareceram alguns erros por causa das dependências de libs, /etc/passwd, /etc/group. Neste caso, basta criar o novo *path* delas a partir de: /opt/apache/ , e copiar os arquivos que estão faltando.

Exemplos:

```
$ chroot /opt/apache/   /usr/sbin/httpd –f   /conf/httpd.conf &
Couldn't open /usr/libexec/ld.so.

        Para isso basta criar o diretório e copiar o ld.so:

$ pwd
/opt/apache
$ mkdir usr
$ mkdir usr/libexec
$ cp /usr/libexec/ld.so /usr/libexec

$ chroot /opt/apache/   /usr/sbin/httpd –f   /conf/httpd.conf &
ld.so failed: Can't find shared library "libcrypt.so.2.0"
$ pwd
/opt/apache
$ mkdir ./usr/lib
$ cp /usr/lib/libcrypt.so.2.0 usr/lib

$ chroot /opt/apache/   /usr/sbin/httpd –f   /conf/httpd.conf &
ld.so failed: Can't find shared library "libc.so.3.0"
$ cp /usr/lib/libc.so.3.0 usr/lib

$ chroot /opt/apache/   /usr/sbin/httpd –f   /conf/httpd.conf &
httpd: bad user name nobody
$ pwd
/opt/apache
$ mkdir etc
$ echo 'nobody:*:65534:65534::0:0:Ninguem:/nada:/nada' > etc/master.passwd
$ pwd_mkdb -p -d  etc/ etc/master.passwd

$ chroot /opt/apache/   /usr/sbin/httpd –f   /conf/httpd.conf &
httpd: bad group name nogroup
$ pwd
/opt/apache
$ echo 'nogroup:*:65533' > etc/group
$ chroot /opt/apache/   /usr/sbin/httpd –f   /conf/httpd.conf &
httpd-chrooted
```

O *perl* deve ser adicionado a partir da nova raiz.

```
$ pwd
```

```
/opt/apache
$ mkdir usr ; mkdir usr/bin
$ cp /usr/bin/perl usr/bin/perl
```

Quando executarmos algum script perl , podemos obter mensagens de erros em virtude das dependências.

```
ld.so failed: Can't find shared library "libm.so.2.0"
  [Sun  Nov 1 1:11:11 2000] access to /cgi-bin/my.cgi
failed for 127.0.0.1, reason: Premature end of script headers

$ pwd
  /opt/apache
$ cp /usr/lib/libm.so.2.0 usr/lib
```

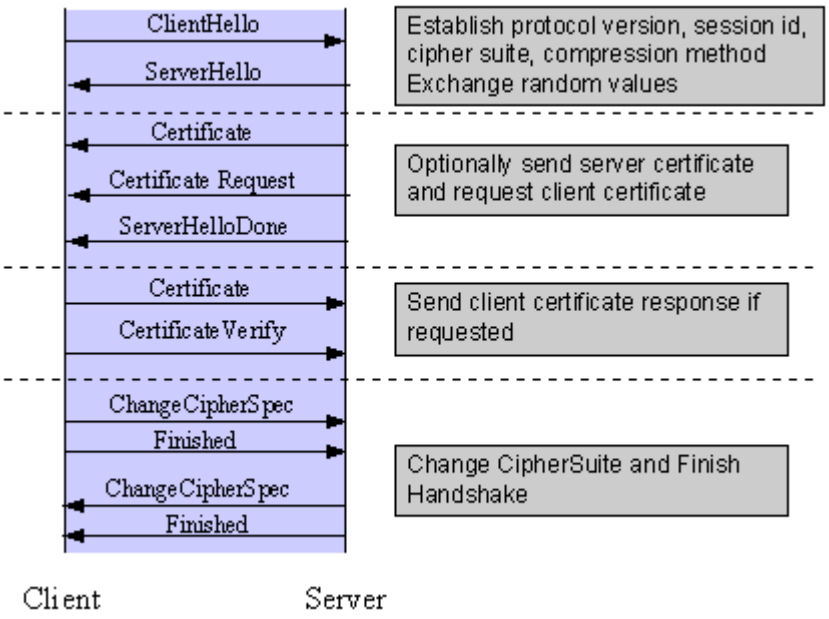
Caso utilize outras linguagem para cgi's o procedimento acima deve ser repetido para acharmos as dependências.

2.9 - Apache com SSL

Em alguns servidores existe a necessidade de utilizar encriptação em virtude da sua natureza, neste exemplo usaremos o SSL. Também usaremos chaves que não são registradas em uma C.A. (Autoridade Certificadora), em virtude do valor destas chaves. Neste caso, as chaves não poderão ser de 128 bits (2^{128} combinações possíveis). Caso o *site* utilize algum *firewall*[5.33] ou proteção a porta 443 tcp/udp deverá ser liberada par que o apache + SSL funcione adequadamente.

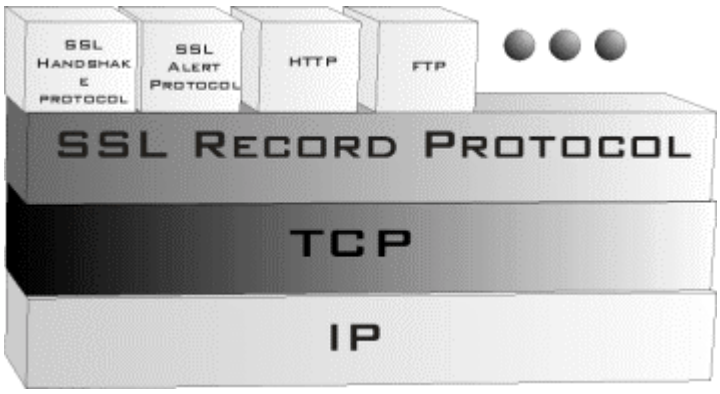
O SSL atua no nível de aplicação, com a utilização do SSL procuramos atender requisitos básicos de privacidade, integridade e autenticação.

Estabelecimento da conexão cliente–servidor (*handshake sequence*) se dá pelo exemplo:



O conjunto usado pelo SSL possui três algoritmos:

- Algoritmo para troca de chaves;
- Algoritmo para cifragem de dados;
- Algoritmo para inserção de redundância nas mensagens.



Relação de autoridades certificadoras:

Autoridades Certificadoras	**Site**
128i	http://www.128i.com
BelSign NV/SA	http://www.belsign.be
CertiSign	http://www.certisign.com.br
Deutsches Forschungsnetz	http://www.pca.dfn.de/dfnpca/certify/ssl/
Equifax Inc.	http://www.equifaxsecure.com/ebusinessid/
Excon	http://www.excon.com.br
Global Sign	http://www.globalsign.net

IKS GmbH	http://www.iks-jena.de/produkte/ca/
NLsign B.V.	http://www.nlsign.nt
TC TrustCenter	http://www.trustcebter.de/
Thawte Consulting	http://www.thawte.com/
Uptime Commerce Ltd.	http://www.uptimecommerce.com
Verisign	http://www.verisign.com

Neste documento utilizaremos os softwares:

Softwares	**Site**	**Autor**
Apache 1.3.9	http://www.apache.org	-
Ssleay 0.9.0	ftp://ftp.apache-ssl.org/SSLeay/ http://www.apache-ssl.org/	- -
Mod ssl 2.4.6	ftp://ftp.modssl.org/source/ http://www.modssl.org/source/	Criada em Abril de 1998 por Ralf S. Engelschall
Openssl 0.9.4	http://www.openssl.org	Biblioteca craida por Eric A. Young e Tim Hudson

O sistema operacional usado será o Linux Red Hat.

Instalação do Ssleay:

```
$ tar -xvf SSLeay-0.9.0b.tar
$ cd SSLeay-0.9.0b
$ ./Configure  linux-elf
$ make
$ find . -name  ssleay -print
  ./apps/ssleay
$ cp apps/ssleay /usr/local/bin/
# Copiar o binário ssleay para
# /usr/local/bin/
```

Instalação do Openssl:

```
$ tar -xvf openssl-0.9.4.tar
$ cd openssl-0.9.4
$  make
$  make test
$  make install
```

Instalação do Apache:

```
$ tar -xvf apache_1.3.9.tar
```

Instalação do Modssl e finalização da instalação do apache:

```
$ tar -xvf mod_ssl-2.4.6-1.3.9.tar
$ cd mod_ssl-2.4.6-1.3.9
$ ./configure --with-apache=../apache_1.3.9 --with-ssl=../openssl-0.9.4 --
prefix=/usr/local/www
# -- with-apache = Path do apache
# --with-ssl        =  Path do openssl
# --prefix          =   Path onde será instalado o apache

$ cd ../apache_1.3.9
$ make
$ make certificate
# vide anotação 2.9.1, abaixo
$ make install
$ chmod 511 usr/local/www/bin usr/local/www/bin/*
$ /usr/local/www/bin/apachectl startssl
# Startar o apche com suporte ssl
$ ps uax|grep http
# (deverão aparecer várias listagem com: /usr/local/www/bin/httpd -DSSL)
```

Para carregar o servidor com suporte *ssl* basta adicionar em um script rc a linha de comando: *"**$ /usr/local/www/bin/apachectl startssl".***

Após executar a linha "***$ /usr/local/www/bin/apachectl startssl"*** será solicitada senha digitada na hora da criação dos certificados, isso pode ser um incomodo porque quando o servidor for reiniciado nem sempre os administradores estão presentes, e o excesso de intervenção humana no servidor é um retrocesso no que esperamos de um sistema informatizado. Essa digitação desnecessária acontece porque a proteção das chaves neste momento é realizado pela digitação da senha sem a preocupação das permissões, isso é uma postura protetora, evitando que o administrador deixe as permissões das chaves como 777, 755 e outras. O procedimento abaixo mostrará como evitar essa digitação excessiva de senha ao inicializar o servidor **Apache com ssl**. Lembrando que as permissões dos certificados (senhas) devem ser 400.

```
$ cd /usr/local/www/conf/ssl.key
$ cp server.key server.key.sav
$ openssl rsa –in server.key.sav –out server.key
$ chmod 400 server.key server.key.sav
$ /usr/local/www/bin/apachectl startssl
# Agora a senha não será solicitda, a proteção do certificado está na
# permissão 400 do /usr/local/www/conf/ssl.key/server.key
```

Tabela 2.9.1

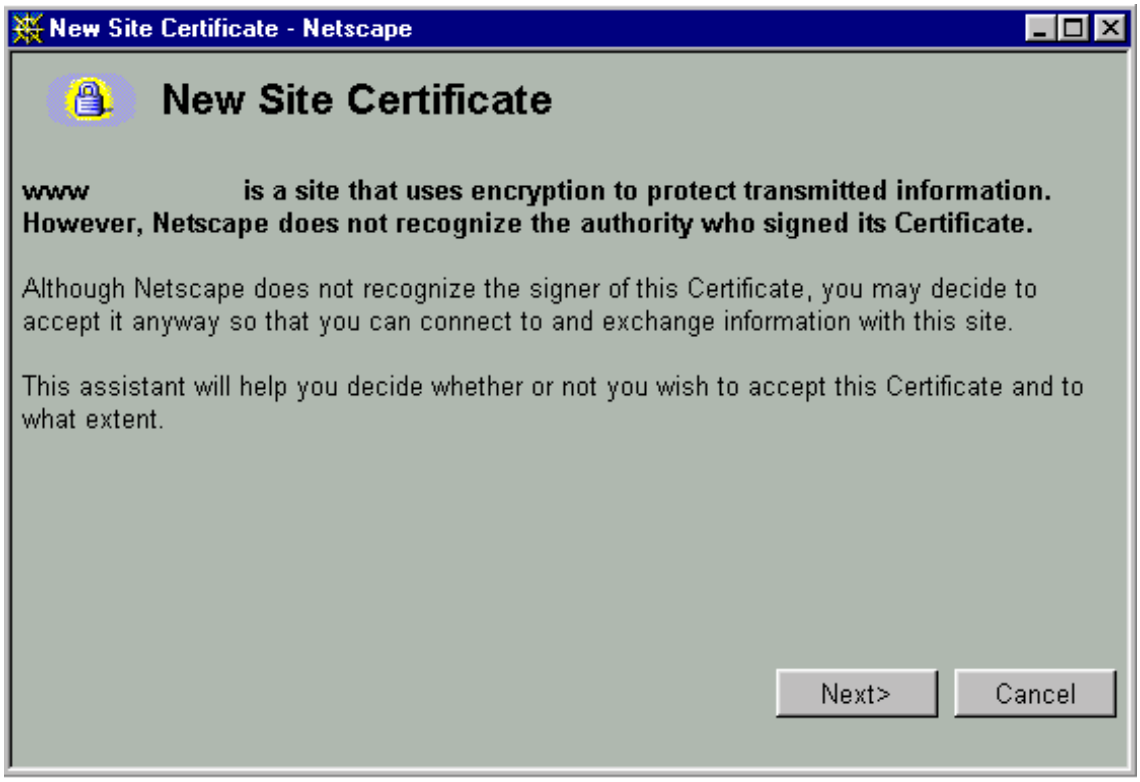
Caso seja necessário criar novos certificados, basta:

```
$ cp /usr/local/teste/openssl-0.9.4/apps/openssl /usr/local/bin/
```

```
# copiar o binário openssl para um diretório do path do root
$ cp /usr/local/teste/mod_ssl-2.4.6-1.3.9/pkg.contrib/sign.sh .
# script para assinatura
$ cd /usr/local/www/conf/ssl.key/
$ openssl genrsa -des3 -out /usr/local/www/conf/ssl.key/server.key 1024
$ openssl req -new -key /usr/local/www/conf/ssl.key/server.key -out
/usr/local/www/conf/ssl.key/server.csr
# Gerando a CSR – Certificate Signing Request
# Este certificado é enviado para a C.A .
$ openssl genrsa -des3 -out /usr/local/www/conf/ssl.key/ca.key 1024
$ openssl req -new -x509 -days 365 -key /usr/local/www/conf/ssl.key/ca.key -out
/usr/local/www/conf/ssl.key/ca.crt
$ ./sign.sh server.csr
$ cp server.csr /usr/local/www/conf/ssl.csr/
$ cp server.crt  /usr/local/www/conf/ssl.crt/
$ chmod 400 /usr/local/www/conf/ssl.key/server* /usr/local/www/conf/ssl.key/ca*
$ chmod 400 /usr/local/www/conf/ssl.crt/server* /usr/local/www/conf/ssl.key/ca*
$ chmod 400 /usr/local/www/conf/ssl.csr/server* /usr/local/www/conf/ssl.key/ca*
#  Agora basta executar o procedimento "Tabela 2.9.1"para evitar a digitação da senha
# do certificado toda vez que o servidor for reiniciado.
$ killall –9 httpd
$ /usr/local/www/bin/apachectl startssl
```

Tabela 2.9.2

Agora basta usar um browser como o *netscape* e digitar a URL https://my_ip/ , caso não ocorra erros teremos uma tela informando que o site possui suporte a SSL e não é certificado por uma C.A..:



Após esta tela terão várias telas com informações sobre o site (servidor SSL) e avisando que o certificado não é autorizado por uma C.A. , basta clicar em <next> e em <finish> na ultima tela de informação.

O arquivo de configuração httpd.conf terá algumas configurações extras no final, como por exemplo:

```
# os parâmetros relevantes serão comentados
<IfDefine SSL>
Listen 80
Listen 443
</IfDefine>
# define as portas do server, e 443 como
# a usada pelo server ssl
SSLPassPhraseDialog  builtin
SSLSessionCache        dbm:/usr/local/www/logs/ssl_scache
# diretório usado para cache
SSLSessionCacheTimeout  300
# Time out da sessão ssl
SSLMutex  file:/usr/local/www/logs/ssl_mutex
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
SSLLog      /usr/local/www/logs/ssl_engine_log
# Diretório de log do protocolo SSL
SSLLogLevel info
# Log diretrize info para o syslogd
```

```
</IfModule>
<IfDefine SSL>
<VirtualHost _default_:443>
#define o virtual host ssl na porta 443
DocumentRoot "/usr/local/www/htdocs"
# Diretório onde estará as Home Pges
ServerName localhost.localdomain
# FDQN do server
ServerAdmin webmaster@localhost.localdomain
# responsável pelo server
ErrorLog /usr/local/www/logs/error_log
TransferLog /usr/local/www/logs/access_log
# Logs do servidor SSL
SSLEngine on
# habilita o server ssl
ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile /usr/local/www/conf/ssl.crt/server.crt
# Certificado RSA ou DSA do servidor
SSLCertificateKeyFile /usr/local/www/conf/ssl.key/server.key
# Certificado, Key privada
#SSLCertificateChainFile /usr/local/www/conf/ssl.crt/ca.crt
#SSLCACertificatePath /usr/local/www/conf/ssl.crt
# Diretório onde ficam os certificados C.A .
#SSLCACertificateFile /usr/local/www/conf/ssl.crt/ca-bundle.crt
# Certificado  da C.A.
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
CustomLog /usr/local/www/logs/ssl_request_log \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
</IfDefine>
```

- Anotação 2.9.1.:

No momento da criação dos certificados ocorrerão 4 etapas:

1° etapa: Geração da chave privada server.key, realizada automaticamente. Usaremos o algorítmo RSA, também existe a opção do DSA.

2° etapa: Preenchimento de dados para o certificado, as informações devem estar corretas porque o certificado será validado apenas no servidor que ele foi gerado.

```
Enter PEM pass phrase:Nomonomonomo nomonomo
Verifying password - Enter PEM pass phrase:Nomonomonomo nomonomo
Country Name (2 letter code) [US]:BR
State or Province Name (full name) [MA]: Rio de Janeiro
Locality Name (eg, city) [Cambridge]:RJ
Organization Name (eg, company) [The Open Group]: CBPF
```

```
Organizational Unit Name (eg, section) [Research Institute]:CAT/CBPF
Common Name (eg, YOUR name) [example.osf.org]: www.cat.cbpf.br
Email Address []:ssl@cat.cbpf.br
```

3° etapa: Gera o certificado x.509, para isso deveremos escolher a opção de 1 a 3. No nosso caso escolheremos a 3.

4° etapa: Será solicitada uma senha para a chave privada, esta senha será solicitada toda a vez que o servidor com suporte ssl for startado. Após a criação dos certificados ele foram colocados em :

```
$ find /usr/local/ -name "ssl*" -print
/usr/local/www/conf/ssl.key
-rw-r--r--   1 root     root         1207 Oct  1 22:44 README.KEY
-r--------   1 root     root          963 Oct  1 22:44 server.key
-r--------   1 root     root          668 Oct  1 22:44 snakeoil-ca-dsa.key
-r--------   1 root     root          887 Oct  1 22:44 snakeoil-ca-rsa.key
-r--------   1 root     root          668 Oct  1 22:44 snakeoil-dsa.key
-r--------   1 root     root          891 Oct  1 22:44 snakeoil-rsa.key

/usr/local/www/conf/ssl.csr
-rw-r--r--   1 root     root         926 Oct  1 22:44 README.CSR
-r--------   1 root     root         733 Oct  1 22:44 server.csr

/usr/local/www/conf/ssl.crt

lrwxrwxrwx   1 root     root          19 Oct  1 22:44 0cf14d7d.0 -> snakeoil-ca-dsa.crt
lrwxrwxrwx   1 root     root          16 Oct  1 22:44 5d8360e1.0 -> snakeoil-dsa.crt
lrwxrwxrwx   1 root     root          16 Oct  1 22:44 82ab5372.0 -> snakeoil-rsa.crt
-rw-r--r--   1 root     root        1522 Oct  1 22:44 Makefile
-rw-r--r--   1 root     root        1386 Oct  1 22:44 README.CRT
-r--------   1 root     root      102580 Oct  1 22:44 ca-bundle.crt
lrwxrwxrwx   1 root     root          19 Oct  1 22:44 e52d41d0.0 -> snakeoil-ca-rsa.crt
lrwxrwxrwx   1 root     root          10 Oct  1 22:44 fa1ffcec.0 -> server.crt
-r--------   1 root     root        1180 Oct  1 22:44 server.crt
-r--------   1 root     root        1472 Oct  1 22:44 snakeoil-ca-dsa.crt
-r--------   1 root     root        1192 Oct  1 22:44 snakeoil-ca-rsa.crt
-r--------   1 root     root        1452 Oct  1 22:44 snakeoil-dsa.crt
-r--------   1 root     root        1176 Oct  1 22:44 snakeoil-rsa.crt
```

3 - Outras Soluções existentes no mercado

- Apache+ssl-1.15 Apache-SSL

ftp://ftp.ox.ac.uk/pub/crypto/SSL/apache_1.2.6+ssl_1.15.tar.gz

ftp://ftp.win.ne.jp/pub/network/security/apache-ssl/apache_1.2.6+ssl_1.15.tar.gz

- CERN httpd

ftp://ftp.w3.org/pub/httpd/
ftp://ftp.win.ne.jp/pub/network/www/w3c/httpd/

- NCSA HTTPd

ftp://ftp.ncsa.uiuc.edu/Web/httpd/Unix/ncsa_httpd/
ftp://ftp.win.ne.jp/pub/network/www/ncsa/httpd/Unix/ncsa_httpd/

- WN - server for the HTTP

http://hopf.math.nwu.edu/
ftp://ftp.win.ne.jp/pub/network/www/

- SQUID (Proxy Cache)

ftp://squid.nlanr.net/pub/
http://cache.is.co.za/squid
http://squid.nlanr.net
ftp://ftp.squid-cache.org/pub/squid-2/STABLE/

- Wwwoffle (Proxy Cache)

http://www.gedanken.demon.co.uk/wwwoffle/

- Squid Guard (Proxy Cache)

http://ftp.ost.eltele.no/pub/www/proxy/squidGuard

- Webalizer (IDS e estatística do web server)

ftp://webalizer.dexa.org/pub/webalizer/
http://www.mrunix.net/webalizer/download.html

4 - Bibliografia:

Sites:

- Absoluta - “http://www.absoluta.org/seguranca/”;
- Aker firewall - “http://www.aker.com.br”;
- Apache – “http://www.apache.org”;
- Apostila - “http://apostilas.virtualave.net”;
- Atrition - “http://www.attrition.org”;
- BOS-BR - “http://www.sekure.org”;
- Brasirc - “http://www.brasirc.net/seguranca/”;
- BSD - “http://www.bsd.org”;
- CERT-RS – “http://www.cert-rs.tche.br”;
- CERT - “http://www.cert.org/”;
- CIAC - “http://ciac.llnl.gov/ciac/”;
- Conectiva - “http://www.conectiva.com.br”;
- CEFET-RJ - “http://www.cefet-rj.br/Ensino/Engenharia/redeslocais/”;
- CEFET-RJ - “http://www.cefet-rj.br/Ensino/Engenharia/redeslocais/default.htm”;
- CEFET-RJ - “http://www.cefet-rj.br/Nucleo/Seminario/seminarios.htm”;
- Defecon - “http://www.defcon.org”;
- FreeBSD - “http://www.freebsd.org”;
- Freshmeat - “http://www.freshmeat.net”;
- Forbidden.net-security - “http://forbidden.net-security.org” ;
- ftp unicamp - “ftp://ftp.unicamp.br”;
- GNU Project - “http://www.gnu.org”;
- GTA UFRJ - “http://www.gta.ufrj.br”;

- Hacker - “http://www.hacker.com.br”;
- Hackers - “http://www.hackers.com.br”;
- HackerNews - “http://www.hackernews.com.br”;
- HackersNews - “http://www.hackersnews.com.br”;
- ISS - “http://www.iss.net”;
- LDP - “http://ldp.conectiva.com.br”;
- Linux.com – “http://www.linux.com”;
- Linux.org - “http://www.linux.org”;
- Linux Security - “http://www.linuxsecurity.com.br”;
- Linux Unicamp - “http://www.linux.unicamp.br”;
- Linux Usp - “http://www.linux.usp.br”;
- Modulo - “http://www.modulo.com.br”;
- NetBSD - “http://www.netbsd.org”;
- Net-Security - “http://www.net-security.org”;
- NIC - “http://www.nic.br”;
- Olinux - “http://www.olinux.com.br”;
- OpenBSD – “http://www.openbsd.org”;
- OpenSSL - “http://www.openssl.org”;
- OpenWap - “http://www.openwap.org”;
- Packetstorm - “http://packetstorm.securify.com/”;
- Penta UFRGS - “http://www.penta.ufrgs.br”;
- RedHat - “http://www.redhat.com”;
- RNP – “http://www.rnp.br”;
- RootSheel - “http://www.rootshell.com”;
- Search Lycos - “http://ftpsearch.lycos.com/”;

- SecureNet - “http://www.securenet.com.br”;
- Security focus - “http://www.securityfocus.com”;
- security.linuxtoday - “http://security.linuxtoday.com”;
- Security Portal - “http://www.securityportal.com”;
- Segurança NT - “http://www.segurancant.com/”;
- SQUID CACHE – “http://squid.nlanr.net”;
- SSL - “http://www.ssl.org”;
- Technotronic – “http://www.technotronic.com”;
- Unicamp - “http://www.security.unicamp.br”;
- Wap master - “http://www.wapmaster.com.br”;
- Wapresources - “http://www.wap-resources.net”;
- Wwsecurity - “http://www.wwsecurity.net” .

Livros:

- Computer Networks; Andrew S. Tanenbaum, Prentice Hall PTR, 1996;
- Segurança na Internet; Marcus Gonçalves; Axcel Books, 1997;
- Comércio & segurança na Web; Simson Garfinkel; Market Books Brasil, 1999;
- Segurança na Internet; Terry Bernstein; Editora Campus, 1997.

Revistas:

- PC Master; Ano3 número 7; Editora Europa, 1999;
- Security Magazine - “http://www.securitymagazine.com.br”;
- Revista do Linux - “http:// www.revistadolinux.com.br”.

5 - Glossário:

[5.1] 802.3
Padrão IEEE para ethernet.

[5.2] ACK
Abreviatura para acknowledgement, confirmação.

[5.3] Active X
Ambiente de Programação baseado em programação por objetos, arquitetura compatível com o Microsoft Windows, criado pela microsoft; objetivando criar mais agilidades e opções de novas aplicações mais pesadas.

[5.4] ANSI (American National Standards Institute)
Grupo que define padrões nos EUA.

[5.5] ARP (Address Resolutions Protocol)
Vículo entre o endereço IP e o endereço de nível físico.

[5.6] Apache
Servidor de HTTP desenvolvido pela "The Apache Software Foundation". Atualmente é o servidor de web mas usado na internet. Suportado por plataformas Linux, Unix, OS/2, Windows e etc....

[5.7] Applet
Uma página da www pode fazer referência a um programa escrito em *Java*. Desta forma, o programa é trascrito junto com a página. Chama-se este programa de *"Applet"*.

[5.8] Binário
Sistema numérico que utiliza a base 2.

[5.9] Bit
A menor unidade dentro de uma posição de memória.

[5.10] Browsers
São programas que possibilitam a visualizam de Home Pages.

[5.11] BSD
Linha (sabor) de unix desenvolvido na universidade de Berkley da California EUA. Possui basicamente quatro distribuições: BSDI, FreeBSD, NetBSD e OpenBSD.

[5.12] BUG
Expressão que caracteriza falha de programação em software ou em um sistema operacional.

[5.13] C
Linguagem de programação muito utilizada no meio científico, usada para escrever 99% do código do UNIX, Linux e outros sistemas operacionais. Pode ser usada em computadores de pequeno porte ou em super computadores.

[5.14] CGI (*Common Gaytway Interface)*
Programas que possibilitam real possibilidade de interagir com o usuário.

[5.15] Chave privada
Chave mantida em sigilo no sistema de criptografia assimétrico.

[5.16] Chave publica
Chave distribuida para o mundo.

[5.17] chroot
Possibilita a alteração da raiz do sistema tendo como referencial um o software que foi startado pelo chroot.

[5.18] Conectiva Linux
Empresa brasileira que distribui uma versão do linux baseada no linux Red Hat.

[5.19] Compilador
Programa usado para traduzir uma informação em linguagem de máquina

[5.20] Criptografia
Arte de codificar e decodificar mensagens.

[5.21] Criptografia chave pública
Técnica de criptografia onde uma chave deve ser mantida em sigilo e a outra distribuida.

[5.22] Criptografia assimétrica
Algorítmo onde chaves diferentes são utilizadas no processo de criptografia de descriptografia.

[5.23] Criptografia simétrica
Algorítmo onde a mesma chave é utilizada no processo de criptografia de descriptografia.

[5.24] Decimal
Sistema numérico que utiliza a base 10.

[5.25] DNS (Domain name system)
Base de dados que converte endereço ips em nomes ou endereços ips em nomes. Porta 53/udp. Também pode fornecer outras informações como: localização do servidor de e-mail, função dos servidores, sistema operacional dos servidores e outras informações.

[5.26] EAD
Vide ensino a distância.

[5.27] Endereço IP
Endereço com 32 bits (IPV4) , este caracteriza um host na rede. Também é dividido em redes classe A, B, C, D e E.

[5.28] Ensina a Distância
Utilização de recursos da internet para prover ensino, abramgendo desde o ensino fundamental até a pós-graduação.

[5.29] FDQN
Nome completo do host, composto de host mais dmain.

[5.30] FreeBSD
Sistema operacional baseado nos BSD de Berkley, voltado para as plataforma I386.

[5.31] Filtro de pacotes
Equipamento que encaminham ou descartam pacotes em função de regras pré-definidas.

[5.32] Filtro de pacotes Inteligente
Filtro de pacotes aprimorado, onde alguns problemas do filtro de pacotes são sanados.

[5.33] Firewall
Equipamento usado para proteger uma rede, pode ser dividido basicamente em firewall de filtro de pacotes, NAT, proxy (gateway de aplicação) e gateway de circuito.

[5.34] FTP (File Transfer Protocol)
Protocolo para transferência de arquivos, porta 21/tcp.

[5.35] Gateway
Equipamento responsável pelo roteamento de pacotes entre duas ou mais redes.

[5.36] Gzip
Compactador muito usado no unix.

[5.37] Gopher
Serviço precursor do WWW.

[5.38] Gunzip
Descompactador muito usado no unix.

[5.39] Handshake Sequence
Sequência de comunicação do protocolo(s) que caracteriza(m) estabelecimento da conexão entre cliente e servidor.

[5.40] HOME PAGE
Um local na web que representa a presença de uma organização.

[5.41] HOST
Qualquer computador, equipamento, de usuário final que conecta à rede.

[5.42] HTTP (Hyper Text Transfer Protocol)
Protocolo da World Wide Web.

[5.43] ICMP (Internet Control Message Protocol)
Integrante do protocolo IP e lida com mensagens de controle e erro.

[5.44] IEEE 802.3
Vide 802.3 .

[5.45] Internet
Conjunto de hosts, redes e roteadores que utiliza o protocolo TCP/IP unindo mais de 61 paises. Formando uma única rede. **Intranet** – uma Internet local

[5.46] INTERNIC
Organização que fornece informações sobre serviços e protocolos. Também é responsável pelo registro de IP e nomes na internet.

[5.47] Inetd
Software que gerencia o processo inicializado por um servidor. Chamado de super servidor.

[5.48] IP (Internet Protocol)
Protocolo usado para identificar os computadores com endereços ips de origem e destino.

[5.49] IPV4
Versão atual do ip.

[5.50] IPV6
Versão da nova versão do IP.

[5.51] ISP
Internet Service Provider, Provedor de Serviço Internet.

[5.52] JAVA
Linguagem de script que independe da plataforma. Esta é uma poderosa linguagem orientada a objeto criada pela SUN.

[5.53] JAVA SCRIPT
É uma adaptação da *Java*, de forma a ser interpretada pelo Browser, sem necessidade de ser compilada pelo autor.

[5.54] Linux
Sistema operacional baseado no minix desenvolvido por Linus Trovald no início da década de 90. Hoje este sistema operacional possui diversas distribuições pelo mundo.

[5.55] LOG
Armazenamento de ocorrências para realização de auditorias.

[5.56] Mapa Sensitivo
São figurar onde são definidas zonas através de coordenadas de forma a selecionar áreas que ao serem acionadas funcionam como links.

[5.57] NetBSD
Sistema operacional baseado nos BSD de Berkley, voltado aspectos de segurança e multiplataforma.

[5.58] NEWS
É composto por informações agrupadas por categorias

[5.59] NFS (Network file system)
Possibilita a compartilhamento de *file system* entre duas ou mais máquinas UNIX/Linux. Porta 2049/tcp e udp.

[5.60] OpenBSD
Sistema operacional baseado nos BSD de Berkley, voltado para aspectos de segurança. Possui grande suporte a criptografia, VPN e firewall. Considerado um dos sitemas mais seguros entre os sistemas free e pagos em ambiente standard.

[5.61] OS/2
Sistema operacional desenvolvido pela IBM voltado para o mercado coorporativo e doméstico.

[5.62] Pacote
Um conjunto de informações, dados e cabeçlhos, que são usados na transmissão de dados.

[5.63] Perl
Linguagem interpletada voltada para o processamento de texto, muito utilizada em CGI's na WEB. Também facilita a depuração de logs do sistema.

[5.64] Plugins
São módulos que podem ser conectados (instalados) ao browser, de forma que este passe a reconhecer os mais diversos formatos de informação. Diversas empresas criaram *Plugins*, permitindo assim ao browser visualizar gráficos vetoriais, vídeos, som, planilhas, *VRML*, *Flash* e outros.

[5.65] Protocolo
Descrição formal das regras que possibilitam a troca de informações entre os dispositivos de rede.

[5.66] Provedores de Backbone Internet
São instituições que disponibilizam acesso a internet para redes locais.

[5.67] Raiz
Em UNIX é a conta que detem o controle do sistema.

[5.68] Red Hat
Distribuição do Linux.

[5.69] rlogin
Protocolo de acesso remoto.

[5.70] Root
Raiz do sistema. Conta que possui o controle sobre o sistema.

[5.71] Sendmail
Implementação mais comum do SMTP.

[5.72] SMTP (Simple Mail Transport Protocolo)
Protocolo de transmissão de e-mails. Porta 25/tcp.

[5.73] SSL (Secure Socks Layer)
Protocolo desenvolvido pela netscape que oferece serviço de segurança acima do protocolo tcp.

[5.74] Standalone
Um servidor é startado, ou processo, sem auxílio de algum software "gerenciador" como o inetd.

[5.75] SSH (Secure Shell)
Protocolo de acesso remoto que possui um canal encriptado. Possui as versões ssh1 e ssh2 estas versões de ssh utilizam algorítimo de encriptação RSA e DSA respectivamente. Porta 22/tcp.

[5.76] SQUID
Software proxy cache para acessos a WEB.

[5.77] Tar
Aglomerador de arquivos muito usado em UNIX/Linux.

[5.78] TCP (Transmission Control Protocol)
Protocolo orientado a comunicação e atua na camada de transporte .

[5.79] TELNET
Protocolo que possibilita a emulação de terminal entre duas maquinas. Porta 23/tcp.

[5.80] UDP (User Datagram Protocol)
Protocolo não orientado a comunicação e atua na camada de transporte .

[5.81] VPN (Virtual Private Netwaork)
Canal de comunicação encriptado gerado através de uma rede de ip pública não confiável.

[5.82] Wais
Sistema de banco de dados de acesso remoto usado na internet.

[5.83] Windows
Sistema operacional desenvolvido pela microsoft, possuidor de vários problemas de estabilidade e segurança. Mas, possui recursos que tornam o sistema amigável.

[5.84] WWW (Word Wide Web)
Recurso de hipermídia para acessar a internet. Desenvolvido pelo CERN.