

Table of Contents

.....	1
1	Introdução.....3
1.1	Versão.....3
1.2	Um resumo da História do Apache.....4
1.3	Enviando Correções/Contribuindo com o projeto.....4
1.4	Características do Apache.....5
1.5	Ficha técnica.....5
1.6	Requerimentos.....7
1.7	Arquivos de log criados pelo Apache.....7
2	Instalação.....8
2.1	Iniciando o servidor/reiniciando/recarregando a configuração.....8
2.2	Opções de linha de comando.....10
2.3	Configurando a porta padrão do Apache.....11
2.4	Especificando as interfaces que o Apache atenderá.....11
2.4.1	Especificando endereços/portas adicionais (a diretiva Listen).....13
3	Adicionando uma página no Apache.....14
3.1	Página Raiz.....14
3.2	Sub-páginas.....14
4	Especificando opções/permisões para as páginas.....16
5	Restrições de Acesso.....26
5.1	Autorização.....26
5.2	Autenticação.....34
5.2.1	Criando um arquivo de Senhas.....34

Table of Contents

5	Restrições de Acesso	
5.2.2	Autenticação através de usuários	36
5.2.3	Autenticação usando grupos	39
5.3	Usando autorização e autenticação juntos	41
5.3.1	Acesso diferenciado em uma mesma diretiva	43
5.4	O arquivo .htaccess	45
5.5	Usando a diretiva SetEnvIf com Allow e Deny	49
5.6	A diretiva <Limit>	50
5.7	Diretiva <LimitExcept>	52
6	Definindo documentos de erro personalizados	53
7	Módulos DSO	55
8	Sistema de Log do Apache	60
8.1	A diretiva AgentLog	60
8.2	A diretiva ErrorLog	60
8.3	A diretiva CustomLog	61
8.4	A diretiva RefererLog	62
8.5	A diretiva RewriteLog	62
8.6	A diretiva RewriteLogLevel	63
8.7	A diretiva ScriptLog	63
8.8	A diretiva ScriptLogBuffer	64
8.9	A diretiva ScriptLogLength	64
8.10	A diretiva TransferLog	64
8.11	A diretiva LogFormat	65
8.12	A diretiva LogLevel	67
8.13	A diretiva Anonymous LogEmail	68

Table of Contents

8	Sistema de Log do Apache	
8.14	A diretiva CookieLog	68
8.15	Relatório gráfico de acesso ao sistema	69
<u>Apêndice A. O programa webalizer poderá ser instalado para gerar um relatório gráfico com a estatísticas de visitas por ano/mes/dia/hora usando os dados do access.log. Outra interessante característica são as estatísticas de códigos http (veja “Apêndice B, Licença de Publicação Livre”.....70</u>		
9	Configurando o Apache como servidor proxy	75
9.1	Controlando o acesso ao servidor proxy	79
9.2	Redirecionamento de conexões no Apache	79
10	Virtual Hosts	81
10.1	Virtual hosts baseados em IP	81
10.2	Virtual hosts baseados em nome	84
10.3	Segurança no uso de IP's em Virtual Hosts	86
11	Uso de criptografia SSL	89
11.1	Servidor apache com suporte a ssl	89
11.2	Gerando um certificado digital	90
11.3	Exemplo de configuração do módulo mod-ssl	90
11.4	Autorizando acesso somente a conexões SSL	95
11.5	Iniciando o servidor Web com suporte a SSL	96
12	Exemplo comentado de um arquivo de configuração do Apache	98
12.1	O arquivo httpd.conf	98
12.2	O arquivo srm.conf	123
12.3	O arquivo access.conf	148

Table of Contents

<u>Apêndice B. Licença de Publicação Livre.....</u>	163
<u>Apêndice C. Códigos de retorno HTTP.....</u>	168

Apache

1 Introdução

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

O servidor web é um programa responsável por disponibilizar páginas, fotos, ou qualquer outro tipo de objeto ao navegador do cliente. Ele também pode operar recebendo dados do cliente, processando e enviando o resultado, para que o cliente possa tomar a ação desejada (como em aplicações CGI, banco de dados web, preenchimento de formulários, etc).

O Apache é um servidor Web extremamente configurável, robusto e de alta performance desenvolvido por uma equipe de voluntários (conhecida como Apache Group), buscando criar um servidor web com muitos recursos e com código fonte disponível gratuitamente via Internet. Segundo a Netcraft (<http://www.netcraft.com/survey/>), o Apache é mais usado que todos os outros servidores web do mundo juntos.

Esta apostila documenta a configuração, personalização, introdução aos mecanismos de autenticação e controle de acesso do Apache, sistema proxy, virtual hosting, e exemplos de configuração do servidor httpd. Ela não tem como objetivo ser uma referência completa de configuração, mas sim abordar didaticamente o assunto.

Esta apostila não tenta ser um guia completo ao Apache, mas tentará mostrar como sua estrutura é organizada, as diretivas principais de configuração, diretivas de segurança, virtual hosting, proxy, o uso de utilitários de gerenciamento do servidor, como personalizar algumas partes do servidor e programas úteis de terceiros para análise e diagnóstico do servidor web. Não deixe também de ver o capítulo “12 Exemplo comentado de um arquivo de configuração do Apache”, que contém muitas explicações interessantes e faz parte do aprendizado.

1.1 Versão

É assumido que esteja usando a versão 1.3.22 do Apache. As explicações contidas aqui podem funcionar para versões posteriores, mas é recomendado que se leia a documentação sobre modificações no programa

(changelog) em busca de mudanças que alterem o sentido das explicações fornecidas aqui.

1.2 Um resumo da História do Apache

O Apache tem como base o servidor web NCSA 1.3 (*National Center of Supercomputing Applications*), que foi desenvolvido por Rob McCool. Quando Rob deixou o NCSA, o desenvolvimento foi interrompido, assim muitos desenvolvedores buscaram personalizar sua própria versão do NCSA ou adicionar mais características para atender às suas necessidades. Neste momento começa a história do Apache com *Brian Behlendorf* e *Cliff Skolnick* abrindo uma lista de discussão para interessados no desenvolvimento, conseguindo espaço em um servidor doado pela *HotWired* e trocando patches, corrigindo problemas, adicionando recursos e discutindo idéias com outros desenvolvedores e hackers interessados neste projeto.

A primeira versão oficial do Apache foi a 0.6.2, lançada em abril de 1995 (neste período a NCSA retomava o desenvolvimento de seu servidor web, tendo como desenvolvedores *Brandon Long* e *Beth Frank*, que também se tornaram membros especiais do grupo Apache, compartilhando idéias sobre seus projetos).

Nas versões 2.x do Apache, a escalabilidade do servidor foi ampliada suportando as plataformas Win32 (não obtendo o mesmo desempenho que em plataformas UNIX mas sendo melhorado gradativamente).

1.3 Enviando Correções/Contribuindo com o projeto

Um formulário está disponível na Web para o envio de correções/sugestões em http://www.apache.org/bug_report.html. Uma lista de anúncios sobre o Apache está disponível em apache-announce@apache.org que divulga correções, novas versões e realização de eventos.

Mais detalhes sobre o desenvolvimento do Apache podem ser obtidos na URL <http://dev.apache.org>.

1.4 Características do Apache

Abaixo estão algumas características que fazem desse servidor web o preferido entre os administradores de sistemas:

- Suporte a scripts CGI usando linguagens como *PHP, Perl, ASP, Shell Script, etc*;
- Módulos DSO (Dynamic Shared Objects) permitem adicionar/remover funcionalidades e recursos sem necessidade de recompilação do programa;
- Autenticação, requerendo um nome de usuário e senha válidos para acesso a alguma página/sub-diretório/arquivo (suportando criptografia via Crypto e MD5). Suporte à autorização de acesso, podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/diretório acessado no servidor;
- Suporte a virtual hosting, por nome ou endereço IP: é possível servir 2 ou mais páginas com endereços/portas diferentes através do mesmo processo, ou usar mais de um processo para controlar mais de um endereço;
- Suporte a servidor proxy ftp e http, com limite de acesso, caching (todas flexivelmente configuráveis). Suporte a proxy e redirecionamentos baseados em URLs para endereços internos;
- Suporte a criptografia via SSL, certificados digitais;
- Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador.
- Suporte a tipos mime;
- Personalização de logs.

1.5 Ficha técnica

Utilitários:

- `apache`: Servidor Web Principal;
- `apachectl`: Shell script que faz interface com o apache de forma mais amigável;

Apache

- `apacheconfig`: Script em Perl para configuração interativa básica do Apache;
- `htpasswd`: Cria/Gerencia senhas criptografadas Crypto/MD5;
- `htdigest`: Cria/Gerencia senhas criptografadas Crypto/MD5;
- `dbmmanage`: Cria/Gerencia senhas em formato DBM (Perl);
- `logresolve`: Faz um DNS reverso dos arquivos de log do Apache para obter o endereço de hosts com base nos endereços IP's;
- `ab`: Apache Benchmarking – Ferramenta de medida de desempenho do servidor Web Apache.

Por padrão, os arquivos de configuração do Apache residem no diretório `/etc/apache`:

- `httpd.conf`: Arquivo de configuração principal do Apache, possui diretivas que controlam a operação do daemon servidor. Um arquivo de configuração alternativo pode ser especificado através da opção `-f` da linha de comando;
- `srm.conf`: Contém diretivas que controlam a especificação de documentos que o servidor oferece aos clientes. O nome desse arquivo pode ser substituído através da diretiva `ResourceConfig` no arquivo principal de configuração;
- `access.conf`: Contém diretivas que controlam o acesso aos documentos. O nome desse arquivo pode ser substituído através da diretiva `AccessConfig` no arquivo principal de configuração;

O servidor Web lê os arquivos acima na ordem que estão especificados (`httpd.conf`, `srm.conf` e `access.conf`). As configurações também podem ser especificadas diretamente no arquivo `httpd.conf`. Note que não é obrigatório o uso dos arquivos `srm.conf` e `access.conf`, mas isto proporciona uma melhor organização das diretivas do servidor, principalmente quando se tem um grande conjunto de diretivas. Um exemplo

comentado desses três arquivos de configuração é encontrado no capítulo “12 Exemplo comentado de um arquivo de configuração do Apache”.

1.6 Requerimentos

A máquina mínima para se rodar um servidor Apache para atender a uma rede padrão 10MB/s é um Pentium 90, 24MB de memória RAM, um HD com um bom desempenho e espaço em disco de acordo com o tamanho projetado de seu servidor web (considerando seu crescimento).

Uma configuração mais rápida para redes 100MB/s teria como processador um Cyrix MX, AMD K6 ou Intel Pentium MMX como plataforma mínima (Cyrix é o recomendado pelo alto desempenho no processamento de strings), barramento de HD SCSI com uma boa placa controladora (Adaptec 19160 ou superior) com 64MB de RAM no mínimo.

1.7 Arquivos de log criados pelo Apache

O servidor httpd grava seus arquivos de log geralmente em /var/log/apache, não é possível descrever os arquivos de log usados porque tanto seus nomes como conteúdo podem ser personalizados no arquivo httpd.conf. Mesmo assim, os arquivos de log encontrados na instalação padrão do Apache são os seguintes:

- access.log: Registra detalhes sobre o acesso às páginas do servidor httpd.
- error.log: Registra detalhes dos erros de acesso às páginas ou erros internos do servidor.
- agent.log: Registra o nome do navegador do cliente (campo UserAgent do cabeçalho http).

Mais referências podem ser encontradas no capítulo “8 Sistema de Log do Apache”. Um bom programa para geração de estatísticas de acesso com gráficos é mencionado na seção “0

Relatório gráfico de acesso ao sistema”.

2 Instalação

Adaptado do [Guia Foca GNU/Linux Avançado – Capítulo 11](#).

```
apt-get install apache apache-doc
```

O pacote `apache-doc` contém a documentação de referência do Apache, é recomendável instalá-lo se for desejado entender melhor seu funcionamento ou consultar diretivas.

2.1 Iniciando o servidor/reiniciando/recarregando a configuração

O Apache pode ser executado tanto como um servidor `inetd` ou como um *daemon*. A inicialização de programas pelo `inetd` é uma boa estratégia quando você precisa de um controle de acesso básico (o fornecido pelo `tcpd`), e o serviço é pouco usado na máquina.

A segurança de um serviço iniciado pelo `inetd` pode ser substituída e melhorada por um firewall bem configurado, garantindo facilidades extras, como por exemplo um relatório de tráfego para a porta do servidor web. Mesmo assim, se o servidor Apache estiver rodando como *daemon* e estiver ocioso, ele será movido para swap liberando a memória RAM para a execução de outros programas.

Neste capítulo, será assumido o funcionamento do Apache como *daemon*, que é o método de funcionamento recomendado para sites de grande tráfego, onde ele é frequentemente requisitado e considerado um serviço crítico.

O método padrão para iniciar programas como *daemons* no Debian é através dos diretórios `/etc/rc?.d`. Cada diretório deste contém os programas que serão executados/interrompidos no nível de execução "?" (`rc1.d/`, `rc2.d/`, etc). O conteúdo destes diretórios são links para os scripts originais em `/etc/init.d/programa`, o nosso programa alvo é `/etc/init.d/apache`. O `/etc/init.d/apache` aceita os seguintes parâmetros:

Apache

- `start`: Inicia o Apache
- `stop`: Finaliza o Apache
- `restart`: Reinicia o Apache, efetuando uma pausa de 5 segundos entre a interrupção do seu funcionamento e reinício.
- `reload`: Recarrega os arquivos de configuração do Apache, as alterações entram em funcionamento imediatamente.
- `reload-modules`: Recarrega os módulos. Basicamente é feito um restart no servidor.
- `force-reload`: Faz a mesma função que o `reload`

Para reiniciar o Apache usando o `/etc/init.d/apache`, digite:

```
./etc/init.d/apache restart
```

ou

```
cd /etc/init.d;./apache restart
```

Na realidade, o que o `/etc/init.d/apache` faz é interagir diretamente com o shell script `apachectl`.

O `apachectl` recebe os parâmetros enviados pelo usuário e converte para sinais que serão enviados para o binário Apache. Da mesma forma, ele verifica os códigos de saída do Apache e os transforma em mensagens de erro legíveis para o usuário comum. Os seguintes comandos são aceitos pelo `apachectl`:

- `httpd-server/start`: Inicia o Apache

- `stop`: Finaliza o Apache (enviando um sinal TERM)
- `restart`: Reinicia o Apache (enviando um sinal HUP)
- `graceful`: Recarrega os arquivos de configuração do Apache (enviando um sinal USR1)
- `fullstatus`: Mostra o status completo do servidor Apache (requer o lynx e o módulo *mod_status* carregado).
- `status`: Mostra o status do processo do servidor Apache (requer o lynx e o módulo *mod_status* carregado).
- `configtest`: Verifica se a sintaxe dos arquivos de configuração está OK (executa um `apache -t`).

2.2 Opções de linha de comando

- `-D nome`: define um nome que será usado na diretiva `<IfDefine nome>`.
- `-d diretório`: especifica o diretório *ServerRoot* (substitui o do arquivo de configuração).
- `-f arquivo` – especifica um arquivo *ServerConfigFile* alternativo.
- `-C "diretiva"`: processa a diretiva antes de ler os arquivos de configuração.
- `-c "diretiva"`: processa a diretiva depois de ler os arquivos de configuração.
- `-v`: mostra a versão do programa.
- `-V`: mostra opções usadas na compilação do Apache.

- -h: Mostra o help on-line do programa
- -l: lista módulos compilados junto com o Apache (embutidos)
- -L: lista diretivas de configurações disponíveis
- -S: Mostra configurações de Virtual Hosting
- -t: executa a checagem de sintaxe nos arquivos de configuração do Apache (incluindo a checagem da diretiva *DocRoot*).
- -T – executa a checagem de sintaxe nos arquivos de configuração do Apache (menos da diretiva *DocRoot*).

2.3 Configurando a porta padrão do Apache

Use a diretiva *Port* para configurar a porta padrão que o Apache receberá requisições por padrão. A diretiva *Listen* também é usada para ajustar o endereço/portas alternativas (usadas também em Virtual Hosts) e substituirá as definições de *Port* (veja o capítulo “2.4.1 Especificando endereços/portas adicionais (a diretiva *Listen*)” para mais detalhes).

OBS:: Somente uma diretiva *Port* e um argumento poderão ser especificados. Para mais controle sobre as portas do sistema use a diretiva *Listen*.

2.4 Especificando as interfaces que o Apache atenderá

A diretiva *BindAddress* é usada para especificar os endereços IP das interfaces ou endereços FQDN em que o Apache responderá às requisições. Mais de um endereço pode ser especificado, separados por espaços. Caso não seja definido, o Apache assumirá o valor "*" (atenderá requisições vindas de qualquer interface).

OBS1: É permitido usar somente uma diretiva `BindAddress`. A diretiva `Listen` deverá ser usada se desejar mais controle sobre as portas do servidor web. Veja a seção seguinte para maiores detalhes.

OBS2: – As interfaces especificadas pela diretiva `Listen` substituirá as especificadas em `BindAddress`.

Exemplos:

`BindAddress 192.168.1.1` – Especifica que os usuários da faixa de rede 192.168.1.* terão acesso ao servidor `httpd`. Isto assume que a máquina possui o endereço 192.168.1.1 em sua interface de rede interna.

`BindAddress *` – Atenderá requisições vindas de qualquer interface de rede.

2.4.1 Especificando endereços/portas adicionais (a diretiva `Listen`)

A diretiva `Listen` é usada para se ter um controle maior sobre a especificação de endereços/portas alternativas que o servidor web esperará por requisições externas. Esta diretiva é muito usada na construção de *Virtual Hosts*. Esta diretiva pode substituir completamente as diretivas `Port` e `BindAddress`. Podem ser usados o número da porta, ou o par endereço:porta:

```
Listen 192.168.1.1:80
```

```
Listen 192.168.7.1:81
```

```
Listen 60000
```

O endereço que deverá ser usado é o da interface de rede (assim como na diretiva `BindAddress`). No exemplo acima, o servidor `httpd` esperará por requisições vindas de 192.168.1.* na porta 80 e também 60000, e requisições vindas de 192.168.7.1 na porta 81 e também 60000.

3 Adicionando uma página no Apache

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

Existem dois tipos de páginas que podem ser adicionadas ao Apache: a página raiz e sub-páginas.

3.1 *Página Raiz*

A página raiz é especificada através da diretiva `DocumentRoot` e será mostrada quando se entrar no domínio principal, como `http://www.focalinux.org`. Na configuração padrão do Apache, `DocumentRoot` aponta para o diretório `/var/www`. Este diretório será assumido como *raiz* caso os diretórios não sejam iniciados por uma `/`:

- `home/focalinux`: Aponta para `/var/www/home/focalinux`
- `/home/focalinux`: Aponta para `/home/focalinux`

Este diretório deve conter um arquivo de índice válido (especificado pela diretiva `DocumentIndex` no `srm.conf`), e permissões de acesso válidas no arquivo `access.conf` para autorizar o acesso às páginas em `/var/www` (veja o capítulo “5 Restrições de Acesso” para maiores detalhes).

3.2 *Sub-páginas*

Sub-páginas são armazenadas abaixo do diretório da *página raiz*, como `http://www.focalinux.org/download`. Elas podem ser um subdiretório da página principal em `/var/www` ou serem criadas através da diretiva `Alias` no arquivo `srm.conf`. Caso seja um sub-diretório, as permissões de acesso de `/var/www` serão herdadas para este sub-diretório, mas também poderão ser modificadas com a especificação de uma nova diretiva de acesso.

Através da diretiva `Alias` a página pode estar localizada em outro diretório do disco (até mesmo outro sistema de arquivos) e as permissões de acesso deverão ser definidas para aquela página. Para criar um endereço

<http://www.focalinux.org/iniciante> que aponta para o diretório `/home/focalinux/download/iniciante` no disco local, basta usar a seguinte diretiva no `srm.conf`:

Alias `/iniciante /home/focalinux/download/iniciante`

Pode ser necessário permitir o acesso à nova página, caso o servidor tenha uma configuração restritiva por padrão (veja o capítulo “5 Restrições de Acesso” para maiores detalhes). Após isto, faça o servidor `httpd` reler os arquivos de configuração ou reiniciá-lo. Após isto, a página `/home/focalinux/download/iniciante` estará acessível via <http://www.focalinux.org/iniciante>.

OBS: Caso inclua uma `/` no diretório que será acessível via URL, o endereço somente estará disponível caso você entre com `/` no final da URL:

Alias `/doc/ /usr/doc/`

O diretório `/doc` somente poderá ser acessado usando a URL <http://www.focalinux.org/doc/>, o uso da URL <http://www.focalinux.org/doc> retornará uma mensagem de URL não encontrada.

4 Especificando opções/permissions para as páginas

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

As opções de restrição podem tanto ser especificadas nas diretivas <Directory>, <Location> ou <Files> quanto nos arquivos .htaccess (ou outro nome de arquivo de controle de acesso especificado pela opção AccessFileName do arquivo de configuração do Apache). Cada diretiva de acesso é especificada entre <tags> e devem ser fechadas com </tag> (como na linguagem HTML). As seguintes diretivas de acesso são válidas no Apache:

Directory: As restrição afetará o diretório no disco especificado, conseqüentemente a página armazenada nele. Por exemplo:

```
<Directory /var/www>
```

```
Order deny,allow
```

```
deny from all
```

```
allow from 10.1.0.1
```

```
</Directory>
```

O acesso ao diretório /var/www será permitido somente ao computador com o endereço IP 10.1.0.1.

DirectoryMatch: Funciona como a diretiva <Directory> mas trabalha com expressões regulares como argumento. Por exemplo:

```
<DirectoryMatch "^/www/.*">
```

```
Order deny,allow
```

```
deny from all
```

```
</DirectoryMatch>
```

Bloqueará o acesso ao diretório /www e sub-diretórios dentro dele.

Files: As restrições afetarão os arquivos do disco que conferem com o especificado. É possível usar os coringas ? e * como no shell. Também podem ser usadas expressões regulares especificando um "~" após Files e antes da expressão. Por exemplo:

```
<Files *.txt>
```

```
Order deny,allow
```

```
deny from all
```

```
</Files>
```

Bloqueia o acesso a todos os arquivos com a extensão .txt

```
<Files ~ "\.(gif|jpe?g|bmp|png)$">
```

```
Order deny,allow
```

```
</Files>
```

Bloqueia o acesso a arquivos gif, jpg, jpeg, bmp, png (note que o "~" ativa o modo de interpretação de expressões regulares).

FilesMatch: Permite usar expressões regulares na especificação de arquivos (equivalente a diretiva <Files ~ "expressão">). Por exemplo:

```
<FilesMatch "\.(gif|jpe?g|bmp|png)$">
```

```
Order deny,allow
```

```
</FilesMatch>
```

Bloqueia o acesso a arquivos gif, jpg, jpeg, bmp, png.

Location: As restrições afetarão o diretório base especificado na URL e seus sub-diretórios. Por exemplo:

```
<Location /security>
```

```
Order allow,deny
```

```
</Location>
```

Bloqueia o acesso de todos os usuários ao diretório /security da URL (a explicação porque o acesso é bloqueado neste caso será explicado na seção “5.1 Autorização”).

LocationMatch: Idêntico a diretiva <Location> mas trabalha com expressões regulares. Por exemplo:

```
<LocationMatch "/(extra|special)/data">
```

```
Order deny,allow
```

```
deny from all
```

```
</LocationMatch>
```

Bloqueará URLs que contém a substring "/extra/data" ou "/special/data".

O uso das diretivas <Directory> e <Files> é apropriada quando você deseja trabalhar com permissões a nível de diretórios/arquivos no disco local (o controle do proxy também é feito via <Directory>), o uso da diretiva <Location> é adequado para trabalhar com permissões a nível de URL. A ordem de processamento das diretivas de acesso são processadas é a seguinte:

1. A diretiva <Directory> (com exceção de <DirectoryMatch>) e os arquivos .htaccess são processados simultaneamente. As definições dos arquivos .htaccess substituem as de <Directory>

2. Expressões regulares de <DirectoryMatch>, <Directory>.
3. <Files> e <FilesMatch> são processados simultaneamente.
4. <Location> e <LocationMatch> são processados simultaneamente.

Normalmente é encontrado a opção *Options* dentro de uma das diretivas acima, a função desta diretiva é controlar os seguintes aspectos da listagem de diretórios:

- **All:** Todas as opções são usadas exceto a *MultiViews*. É a padrão caso a opção *Options* não seja especificada.
- **ExecCGI:** Permite a execução de scripts CGI.
- **FollowSymLinks:** O servidor seguirá links simbólicos neste diretório (o caminho não é modificado). Esta opção é ignorada caso apareça dentro das diretivas <Location>, <LocationMatch> e <DirectoryMatch>.
- **Includes:** É permitido o uso de includes no lado do servidor.
- **IncludesNOEXEC:** É permitido o uso de includes do lado do servidor, mas o comando *#exec* e *#include* de um script CGI são desativados.
- **Indexes:** Se não existir um arquivo especificado pela diretiva <DirectoryIndex> no diretório especificado, o servidor formatará automaticamente a listagem ao invés de gerar uma resposta de acesso negado.
- **MultiViews:** Permite o uso da Negociação de conteúdo naquele diretório. A negociação de conteúdo permite o envio de um documento no idioma requisitado pelo navegador do cliente.
- **SymLinksIfOwnerMatch:** O servidor somente seguirá links simbólicos se o arquivo ou diretório alvo tiver como dono o mesmo user ID do link. Esta opção é ignorada caso apareça dentro das diretivas

<Location>, <LocationMatch> e <DirectoryMatch>.

Múltiplos parâmetros para *Options* podem ser especificados através de espaços.

OBS1: A opção *Options* não tem efeito dentro da diretiva *FILES*.

OBS2: Tanto faz usar maiúsculas quanto minúsculas nas diretivas de configuração, opções e parâmetros de configuração do Apache, a capitalização apenas ajuda a leitura e interpretação: *SymLinksIfOwnerMatch* (*LinksSimbólicosSeDonoConferir*).

As opções especificadas para o diretório afetam também seus sub-diretórios, a não ser que sejam especificadas opções separadas para o sub-diretório:

```
<Directory /var/www>
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

Ao acessar o diretório */var/www/focalinux*, as permissões usadas serão de */var/www*, ao menos que uma diretiva *<Directory>* ou *<Location>* seja especificada:

```
<Directory /var/www>
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

```
<Directory /var/www/focalinux>
```

```
Options Includes
```

```
</Directory>
```

As opções e restrições de acesso de /var/www/focalinux serão EXATAMENTE as especificadas no bloco da diretiva <Directory /var/www/focalinux> e somente os *includes* serão permitidos. Para adicionar ou remover uma opção individual definidas por diretivas anteriores, podem ser usado os sinais "+" ou "-", por exemplo:

```
<Directory /var/www>
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

```
<Directory /var/www/focalinux>
```

```
Options +Includes -Indexes
```

```
</Directory>
```

As opções Indexes e FollowSymLinks são definidas para o diretório /var/www, então as permissões do diretório /var/www/focalinux serão FollowSymLinks (do diretório /web/docs) e Includes (adicionada) e o parâmetro Indexes não terá efeito neste diretório.

É permitido fazer um aninhamento das diretivas <Directory> e <Files>:

```
<Directory /var/www>
```

```
Order allow,deny
```

```
allow from all
```

```
<Files LEIAME-DONO.txt>
```

```
Order deny,allow
```

```
deny from all
```

```
</Files>
```

```
</Directory>
```

Neste caso, somente os arquivos LEIAME-DONO.txt existentes no diretório /var/www e seus sub-diretórios serão bloqueados.

Se a diretiva <Files> for usada fora de uma estrutura <Directory>, ela terá efeito em todos os arquivos disponibilizados pelo servidor. Este é excelente método para proteger os arquivos de acesso, senhas e grupos, conforme será explicado mais adiante.

Qualquer outro tipo de aninhamento de diretivas resultará em um erro de configuração ao se tentar carregar/recarregar o Apache. Um exemplo de diretiva incorreta:

```
<Directory /var/www>
```

Options Indexes FollowSymLinks

```
<Directory /var/www/focalinux>
```

Options +Includes -Indexes

```
</Directory>
```

```
</Directory>
```

O correto é:

```
<Directory /var/www>
```

Options Indexes FollowSymLinks

```
</Directory>
```

```
<Directory /var/www/focalinux>
```

Options +Includes -Indexes

```
</Directory>
```

Espero que tenha observado o erro no exemplo acima.

OBS1: Você pode verificar se a configuração do apache está correta digitando “apache -t” como usuário root, se tudo estiver correto com suas configurações ele retornará a mensagem: "Syntax OK".

OBS2: Se Options não for especificado, o padrão será permitir tudo exceto *MultiViews*.

OBS3: Qualquer restrição afetará o diretório atual e todos os seus sub-diretórios! Defina permissões de sub-diretórios específicos separadamente caso precise de um nível de acesso diferente. Veja também a seção sobre arquivos OverRide (.htaccess) para detalhes sobre este tipo de arquivo.

OBS4: A diretiva de acesso "<Directory />" não afetará outros sistemas de arquivos montados dentro de seus subdiretórios. Caso uma diretiva de acesso padrão não seja especificada para outros sistemas de arquivos, o acesso será automaticamente negado.

5 Restrições de Acesso

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

A restrição de acesso do Apache é feita através de *Autorização* e *Autenticação*. Através da *autorização*, é checado se o endereço/rede especificada tem ou não permissão para acessar a página. A *autenticação* requer que seja passado nome e senha para garantir acesso a página. Os métodos de *Autorização* e *Autenticação* podem ser combinados como veremos mais adiante.

5.1 Autorização

A restrição de acesso por autorização (controlado pelo módulo `mod_access`), permite ou não o acesso ao cliente de acordo com o endereço/rede especificada. As restrições afetam também os sub-diretórios do diretório alvo. Abaixo um exemplo de restrição de acesso que bloqueia o acesso de qualquer host que faz parte do domínio `.spammers.com.br` a URL `http://servidor/teste`:

```
<Location /teste>
```

```
Option Indexes
```

```
Order allow,deny
```

```
allow from all
```

```
deny from .spammers.com.br
```

```
</Location>
```

A opção `Option` foi explicada acima, seguem as explicações das outras diretivas:

· **Order:** Especifica em que ordem as opções de acesso allow/deny serão pesquisadas. Caso não seja especificada, o padrão será deny/allow. Note que a ordem de pesquisa de allow e deny é a inversa da especificada. A diretiva Order aceita os seguintes valores:

o deny,allow Esta é a padrão, significa um servidor mais restritivo; a diretiva allow é processada primeiro e somente depois a diretiva deny. Caso nenhuma diretiva allow e deny seja especificadas, ou não conferirem, **PERMITE TUDO** como padrão.

o allow,deny Significa um servidor mais permissivo, a opção deny é processada primeiro e somente depois a opção allow. Caso nenhuma diretiva allow e deny seja especificada, ou não conferirem, **BLOQUEIA TUDO** como padrão.

o mutual-failure Somente permite o acesso se o usuário receber autorização através da opção allow e **NÃO** ser bloqueado pela opção deny, caso uma das checagens falhe, o acesso é imediatamente negado. É uma opção interessante quando você quer somente pessoas de um determinado endereço/rede acessando o seu sistema, desde que não estejam em sua lista negra :-)

ATENÇÃO: É importante saber se a página será permissiva ou restritiva para escolher a ordem mais adequada ao seu caso, também leve em consideração a possibilidade do processamento cair na diretiva de acesso padrão, caso nem a diretiva allow e deny conferiram e estiver usando a ordem de acesso "allow,deny" ou "deny,allow". Um sistema mal configurado neste aspecto poderá trazer sérias conseqüências.

É comum em páginas permissivas se definir a seguinte configuração:

Order allow,deny

allow from all

O motivo é que em um grande site, se forem adicionadas mais restrições nesta página (devido a alguns domínios que têm usuários mal comportados, bloqueio de acesso à rede do concorrente, potenciais atacantes, etc...), estas deverão ser lidas antes da diretiva "allow from all", e podem passar despercebidas ao

administrador, e podem simplesmente não funcionar caso a opção Order não esteja ajustada corretamente (lembre-se, você é o administrador e a integridade do site depende de sua atenção na escolha da ordem correta das diretivas de acesso).

· **allow from:** Especifica o endereço que terá acesso ao recurso especificado. A diretiva allow from aceita os seguintes valores:

- o all O acesso é permitido a todos.
- o um endereço de domínio completo (FQDN). Por exemplo www.debian.org.br.
- o um endereço de domínio parcial. Qualquer computador que confira com o início ou fim terá o acesso permitido. Por exemplo, .spammers.com.br, .debian.org.
- o um endereço IP completo, como 192.168.1.1
- o um endereço IP parcial como 192.168.1.
- o um par rede/máscara como 10.1.0.0/255.255.0.0 ou 10.1.0.0/16, uma faixa de acesso a máquinas de uma mesma rede pode ser definida facilmente através deste método.

OBS1: É necessário reiniciar o Apache depois de qualquer modificação em seu arquivo de configuração (executando `apachectl restart`), ou recarregar os arquivos de configuração (`apachectl graceful`).

OBS2: Mais de um host pode ser especificado, separando com um espaço:

```
allow from 192.168. .debian.org.br
```

Permitirá o acesso de qualquer máquina que o endereço IP confira com 192.168.*.* e qualquer computador do domínio debian.org.br

OBS3: Regras baseadas em nomes simples de hosts (como www) não conferirão! Deverá ser usado o FQDN ou IP: www.dominio.com.br

OBS4: Caso Order não seja especificado, deny,allow será usado como padrão (ou seja, permitirá tudo como padrão).

· **deny from:** Especifica os endereços que NÃO terão acesso ao recurso especificado. As explicações referentes a esta diretiva de acesso são idêntica as de allow from.

É recomendável o uso de endereços IP ao invés de endereços DNS e um mecanismo anti-spoofing no firewall ou código de roteamento, pois ficará mais difícil um ataque baseado em DNS spoofing, aumentando consideravelmente a segurança de seu servidor web.

ATENÇÃO: Caso receba erros 403 (acesso negado) sem bloquear a URL nas diretivas de acesso, significa que o servidor Web não tem permissões para acessar/abrir o diretório da página. Certifique-se que o *dono* e *grupo* do processo Apache (especificado pela diretiva *User* e *Group*) possuem permissões de acesso àquele diretório.

Abaixo alguns exemplos de permissões de acesso:

```
<Directory /var/www>
```

```
Options SymLinksIfOwnerMatch Indexes MultiViews
```

```
Order allow,deny
```

```
allow from all
```

```
</Directory>
```

Permite o acesso de qualquer usuário de qualquer lugar (allow from all), permite também a visualização da listagem formatada de arquivos caso nenhum arquivo especificado na diretiva DirectoryIndex seja encontrado (Indexes), permite negociação de conteúdo (MultiViews) e seguir links caso o dono do arquivo confira com o nome do link (SymLinksIfOwnerMatch).

```
<Directory /var/www>
```

```
Options SymLinksIfOwnerMatch Indexes MultiViews
```

```
</Directory>
```

Tem o mesmo significado da diretiva acima por métodos diferentes; quando nenhuma opção Order é especificada, deny,allow é definido como padrão, e como nenhuma opção de acesso allow/deny foi especificada, o padrão "Order deny,allow" é usado e permite TUDO como padrão.

```
<Directory /var/www>
```

```
Options Indexes
```

```
Order deny,allow
```

```
deny from all
```

```
</Directory>
```

Esta regra acima não tem muita lógica pois restringe o acesso de todos os usuários ao diretório /var/www, ao menos se esta for sua intenção...

```
<Location /focalinux>
```

Options All

Order allow,deny

allow from all

</Location>

A regra acima permite o acesso a URL <http://www.servidor.org/focalinux> de qualquer host na Internet

<Files .htaccess>

Order deny,allow

deny from all

</Files>

Bloqueia o acesso a qualquer arquivo .htaccess do sistema

<Files ~ "leiam-(arm|alpha|m68k|sparc|powerpc)\.txt">

Order deny,allow

deny from all

</Files>

Bloqueia o acesso a qualquer arquivo `leiam-arm.txt`, `leiam-alpha.txt`, `leiam-m68k.txt`, `leiam-sparc.txt` e `leiam-powerpc.txt` fazendo uso de expressões regulares.

```
<Directory /var/www>
```

```
Options Indexes
```

```
Order mutual-failure
```

```
allow from .dominio.com.br
```

```
deny from lammer.dominio.com.br
```

```
</Directory>
```

A diretiva acima somente permite acesso ao diretório /var/www de máquinas pertencentes ao domínio .dominio.com.br desde que não seja lammer.dominio.com.br.

```
<Directory /var/www>
```

```
Options Indexes MultiViews
```

```
Order allow,deny
```

```
deny from .com .com.br
```

```
allow from all
```

```
</Directory>
```

Bloqueia o acesso ao diretório /var/www de computadores pertencentes aos domínios .com e .com.br.

```
<Directory /var/www>
```

```
Options None
```

```
Order deny,allow
```

```
allow from 192.168.1. .metainfo.org .debian.org
```

```
deny from 200.200.123.
```

```
</Directory>
```

A regra acima permite o acesso de máquinas da rede 192.168.1.*, do domínio *.metainfo.org e *.debian.org, o acesso de máquinas da rede 200.200.123.* é bloqueado (nada contra, peguei nesse número ao acaso :-).

Note que a máquina 192.168.4.10 terá acesso LIVRE a regra acima, pois não conferirá nem com allow nem com deny, então o processamento cairá na diretiva padrão de deny,allow, que neste caso permite o acesso caso nem allow e deny conferiram com o padrão.

```
<Directory /var/www>
```

```
Options None
```

```
Order allow,deny
```

```
allow from 192.168.1. .metainfo.org .debian.org
```

```
deny from 200.200.123.
```

```
</Directory>
```

A regra acima é idêntica a anterior somente com a mudança da opção Order. Bloqueia o acesso de máquinas da rede 200.200.123.* e permite o acesso de máquinas da rede 192.168.1.*, do domínio *.metainfo.org e *.debian.org.

Note que a máquina 192.168.4.10 terá acesso BLOQUEADO a regra acima, pois não conferirá nem com allow nem com deny, então o processamento cairá na diretiva padrão de allow,deny que neste caso bloqueia o acesso.

5.2 Autenticação

Através da *autenticação* (controlado pelo módulo mod_auth) é possível especificar um *nome* e *senha* para acesso ao recurso solicitado. As senhas são gravadas em formato criptografado usando *Crypto* ou *MD5* (conforme desejado). O arquivo de senhas pode ser centralizado ou especificado individualmente por usuário, diretório ou até mesmo por arquivo acessado.

5.2.1 Criando um arquivo de Senhas

O arquivo de senhas pode ser criado e mantido através do uso de 3 utilitários: htpasswd, htdigest e dbmmanage:

5.2.1.1 O htpasswd

Este é usado para criar o arquivo de senhas. Para criar um banco de dados com o nome senhas para o usuário convidado, é usada a seguinte sintaxe:

```
htpasswd -c -m senhas convidado
```

Você será perguntado por uma senha para o usuário convidado e para redigitá-la. A opção "-c" indica que deverá ser criado um arquivo, a opção "-m" indica a utilização de senhas criptografadas usando o algoritmo

MD5, que garante maior segurança que o método *Crypto*. A senha pode ser especificada diretamente na linha de comando através da opção "-b" (isto é um ótimo recurso para utilização em shell scripts ou programas CGI de integração com o navegador).

```
htpasswd -b -d senhas chefe abcdef
```

No exemplo acima, uma senha de alta segurança será introduzida no banco de dados senhas tornando impossível o acesso a página do usuário :-)

Note que esta senha foi cadastrada usando o algoritmo de criptografia *Crypto* (opção -d). O algoritmo *SHA* também pode ser usado como alternativa, através da opção "-s". Para modificar a senha do usuário convidado, basta usar a mesma sintaxe (sem a opção "-c" que é usada para criar um novo arquivo):

```
htpasswd -m senhas convidado
```

ou

```
htpasswd -b -m senhas convidado nova_senha
```

Opcionalmente você pode especificar a opção "-d" para atualizar também o formato da senha para *Crypto*. Podem existir senhas de criptografias mistas (*SHA*, *Crypto*, *MD5*) no mesmo arquivo sem nenhum problema.

A mudança do formato de senhas é útil quando se deseja aumentar o nível de segurança oferecido por um melhor sistema ou para manter a compatibilidade com alguns scripts/programas que compartilhem o arquivo de senhas.

5.2.1.2 O htdigest e dbmmanage

Estes são idênticos ao htpasswd, a diferença é que o htdigest permite criar/manter um arquivo de senhas usando a autenticação Digest, enquanto o dbmmanage permite manter o banco de dados de senhas em um

arquivo DB, DBM, GDBM e NDBM, formatos conhecidos pelo Perl.

5.2.2 Autenticação através de usuários

Através deste método é possível especificar que usuários terão acesso ao recurso definido, usando senhas de acesso individuais criptografadas usando um dos utilitários da seção anterior. Para restringir o acesso ao endereço `http://servidor.org/teste`:

```
<Location /teste>

AuthName "Acesso à página do Foca Linux"

AuthType basic

AuthUserFile /home/gleydson/SenhaUsuario

# AuthGroupFile /home/users/SenhaGrupo

Require valid-user

</Location>
```

Ao tentar acessar o endereço `http://servidor/teste`, será aberta uma janela no navegador com o título *Enter username for Acesso à página do Foca Linux at servidor.org*, a diretiva `Require valid-user` definem que o usuário e senha digitados devem existir no arquivo especificado por `AuthUserFile` para que o acesso seja garantido. Uma explicação de cada opção de acesso usado na autenticação:

- **AuthName** Será o nome que aparecerá na janela de autenticação do seu navegador indicando qual área restrita está solicitando senha (podem existir várias no servidor, bastando especificar várias diretivas de restrições).

- **AuthType** Especifica o método de que o nome e senha serão passados ao servidor. Este método de autenticação pode ser *Basic* ou *Digest*
 - o **Basic** Utiliza a codificação *base64* para encodificação de nome e senha, enviando o resultado ao servidor. Este é um método muito usado e pouco seguro, pois qualquer sniffer instalado em um roteador pode capturar e descobrir facilmente seu nome e senha.
 - o **Digest** Transmite os dados de uma maneira que não pode ser facilmente decodificada, incluindo a codificação da área protegida (especificada pela diretiva *AuthName*) que possui a sequência de login/senha válida. A diferença deste método é que você precisará de arquivos de senhas diferentes para cada área protegida especificada por *AuthName* (também chamada de *Realm*).
- **AuthUserFile** É o arquivo gerado pelo utilitário *htpasswd* que contém a senha correspondente ao usuário
- **AuthGroupFile** É um arquivo texto que contém o nome do grupo, dois pontos (":") e o nome dos usuários que podem ter acesso ao recurso, separados por vírgulas. No exemplo acima ele se encontra comentado, mas a seguir encontrará exemplos que explicam em detalhes o funcionamento desta diretiva.
- **Require** Especifica que usuários podem ter acesso ao diretório. Podem ser usadas uma das 3 sintaxes:
 - o **Require user** usuário1 usuário2 usuário3 Somente os usuários especificados são considerados válidos para ter acesso ao diretório.
 - o **Require group** grupo1 grupo2 grupo3 Somente os usuários dos grupos especificados são considerados válidos para terem acesso ao diretório. Esta diretiva é útil quando deseja que somente alguns usuários de determinado grupo tenham acesso ao recurso (por exemplo, usuários do grupo *admins*).
 - o **Require valid-user** Qualquer usuário válido no banco de dados de senhas pode acessar o diretório. É bem útil quando as opções de acesso especificadas por **Require user** são muito longas.

A opção `Require` deve ser acompanhado das diretivas `AuthName`, `AuthType` e as diretivas `AuthUserFile` e `AuthGroupFile` para funcionar adequadamente.

OBS: É necessário reiniciar o Apache depois de qualquer modificação em seu arquivo de configuração (`apachectl restart`), ou recarregar os arquivos de configuração (`apachectl graceful`). Note que o `apachectl` é somente um shell script para interação mais amigável com o servidor web apache, retornando mensagens indicando o sucesso/falha no comando ao invés de códigos de saída.

Alguns exemplos para melhor assimilação:

```
<Location /teste>
```

```
AuthName "Acesso a página do Foca Linux"
```

```
AuthType basic
```

```
AuthUserFile /home/gleydson/SenhaUsuario
```

```
Require user gleydson
```

```
</Location>
```

As explicações são idênticas a anterior, mas somente permite o acesso do usuário `gleydson` a URL `http://servidor.org/teste`, bloqueando o acesso de outros usuários contidos no arquivo `AuthUserFile`.

```
<Location /teste>
```

```
AuthName "Acesso a página do Foca Linux"
```

```
AuthType basic
```

```
AuthUserFile /home/gleydson/SenhaUsuario
Require user gleydson usuario1 usuario2
</Location>
<Location /teste>
AuthName "Acesso a página do Foca Linux"
AuthType basic
AuthUserFile /home/gleydson/SenhaUsuario
Require user gleydson
Require user usuario1
Require user usuario2
</Location>
```

As 2 especificações acima são equivalentes e permite o acesso aos usuários gleydson, usuario1 e usuario2 a página <http://servidor.org/teste>.

5.2.3 Autenticação usando grupos

Há casos onde existem usuários de um arquivo de senhas que devem ter acesso a um diretório e outros não, neste caso a diretiva `valid-user` não pode ser especificada (porque permitiria o acesso de todos os usuários do arquivo de senha ao diretório) e uma grande lista de usuários ficaria bastante complicada de ser gerenciada

com vários usuários na diretiva `Require user`.

Quando existe esta situação, é recomendado o uso de grupos de usuários. Para fazer uso desse recurso, primeiro deverá ser criado um arquivo que armazenará o nome do *grupo* e dos usuários pertencente àquele grupo usando a seguinte sintaxe (vamos chamar este arquivo de `SenhaGrupo`):

```
admins: gleydson usuario2
```

```
usuarios: usuario1 usuario2 usuario3 gleydson
```

Agora adaptamos o exemplo anterior para que somente os usuários especificados no grupo `admins` do arquivo criado acima:

```
<Location /teste>
```

```
AuthName "Acesso a página do Foca Linux"
```

```
AuthType basic
```

```
AuthUserFile /home/gleydson/SenhaUsuario
```

```
AuthGroupFile /home/gleydson/SenhaGrupo
```

```
Require group admins
```

```
</Location>
```

Agora somente os usuários pertencentes ao grupo `admins` (`gleydson` e `usuario2`) poderão ter acesso ao diretório `/teste`.

OBS1: Verifique se o servidor Web possui acesso a leitura no arquivo de senhas de usuários e grupos, caso contrário será retornado um código "500 – Internal Server Error". Este tipo de erro é caracterizado por tudo estar OK na sintaxe dos arquivos de configuração após checagem com "apache -t" e todas as diretivas de controle de acesso apontam para os diretórios e arquivos corretos.

OBS2:: Sempre use espaços para separar os nomes de usuários pertencentes a um grupo.

OBS3: NUNCA coloque os arquivos que contém senhas e grupos em diretórios de acesso público onde usuários podem ter acesso via o servidor Web. Tais localizações são /var/www, /home/"usuario"/public_html e qualquer outro diretório de acesso público que defina em seu sistema.

É recomendável também ocultar estes arquivos através da diretiva <Files> evitando possíveis riscos de segurança com usuários acessando os arquivos de senha e grupo.

Na distribuição Debian, qualquer arquivo iniciando com .ht* será automaticamente ocultado pelo sistema, pois já existe uma diretiva <Files ~ "\.ht">. Tal diretiva pode também ser especificada no arquivo de acesso .htaccess. Assim um arquivo .htsenha e .htgroup são bons nomes se se estiver desejando ocultar dados de olhos curiosos...

5.3 Usando autorização e autenticação juntos

Os métodos de *autorização* e *autenticação* podem ser usados ao mesmo tempo dentro de qualquer uma das diretivas de controle de acesso. As diretivas de *autorização* são processadas primeiro (mod_access) e depois as diretivas de *autenticação* (mod_auth). Segue um exemplo:

<Directory /var/www>

Options Indexes

Order deny,allow

```
allow from .dominiolocal.com.br  
  
deny from all  
  
AuthName "Acesso ao diretório do servidor Web"  
  
AuthType basic  
  
AuthUserFile /var/cache/apache/senhas  
  
Require valid-user  
  
</Directory>
```

Para ter acesso ao diretório /var/www, primeiro o computador deve fazer parte do domínio .dominiolocal.com.br, assim ela passa pelo teste de autorização, depois disso será necessário fornecer o login e senha para acesso a página, digitando o login e senha corretos, o teste de autenticação será completado com sucesso e o acesso ao diretório /var/www autorizado.

```
<Directory /var/www>  
  
Options Indexes  
  
Order mutual-failure  
  
allow from .dominiolocal.com.br  
  
deny from lammer.dominiolocal.com.br  
  
AuthName "Acesso ao diretório do servidor Web"
```

```
AuthType basic
```

```
AuthUserFile /var/cache/apache/senhas
```

```
AuthGroupFile /var/cache/apache/grupos
```

```
Require group admins
```

```
</Directory>
```

No exemplo acima, é usado o método de autorização com a opção `Order mutual-failure` e o método de autenticação através de *grupos*. Primeiro é verificado se o usuário pertence ao domínio `.dominiolocal.com.br` e se ele não está acessando da máquina `lammer.dominiolocal.com.br`, neste caso ele passa pelo teste de autorização. Depois disso ele precisará fornecer o nome e senha válidos, com o login pertencente ao `AuthGroupFile`, passando pelo processo de autenticação e obtendo acesso ao diretório `/var/www`.

5.3.1 Acesso diferenciado em uma mesma diretiva

É interessante permitir usuários fazendo conexões de locais confiáveis terem acesso direto sem precisar fornecer nome e senha e de locais inseguros acessarem somente após comprovarem *quem* realmente são. Como é o caso de permitir usuários de uma rede privada terem acesso completo aos recursos e permitir o acesso externo ao mesmo recurso somente através de senha. Isto pode ser feito com o uso da diretiva `Satisfy` junto ao bloco de *autorização/autenticação*. Vamos tomar como base o exemplo anterior:

```
<Directory /var/www>
```

```
Options Indexes
```

```
Order mutual-failure
```

```
allow from .dominiolocal.com.br  
  
deny from lammer.dominiolocal.com.br  
  
AuthName "Acesso ao diretório do servidor Web"  
  
AuthType basic  
  
AuthUserFile /var/cache/apache/senhas  
  
AuthGroupFile /var/cache/apache/grupos  
  
Require group admins  
  
Satisfy any  
  
</Directory>
```

Note que o exemplo é o mesmo com a adição da diretiva `Satisfy any` no final do bloco do arquivo. Quando a opção `Satisfy` não é especificada, ela assumirá "all" como padrão, ou seja, o usuário deverá passar no teste de autorização e autenticação para ter acesso.

A diferença do exemplo acima em relação ao da seção anterior é se a máquina passar no teste de autorização ela já terá acesso garantido. Caso falhe no teste de autorização, ainda terá a chance de ter acesso a página passando na checagem de autenticação.

Isto garante acesso livre aos usuários do domínio `.dominiolocal.com.br`. Já os outros usuários, incluindo acessos vindos de `lammer.dominiolocal.com.br` que pode ser uma máquina com muito uso, poderá ter acesso ao recurso caso tenha fornecido um nome e senha válidos para passar pelo processo de autenticação. Tenha isto em mente... este tipo de problema é comum e depende mais de uma política de segurança e conduta interna, o sistema de segurança não pode fazer nada a não ser permitir acesso a um nome e senha válidos.

Tenha cuidado com o uso da opção Satisfy em diretivas que especificam somente o método de autenticação:

```
<Directory /var/www>
```

```
Options Indexes
```

```
AuthName "Acesso ao diretório do servidor Web"
```

```
AuthType basic
```

```
AuthUserFile /var/cache/apache/senhas
```

```
AuthGroupFile /var/cache/apache/grupos
```

```
Require group admins
```

```
Satisfy any
```

```
</Directory>
```

ATENÇÃO PARA O DESCUIDO ACIMA!: Como o método de autorização **NÃO** é especificado, é assumido deny,allow como padrão, que permite o acesso a **TODOS** os usuários. O bloco acima **NUNCA** executará o método de autenticação por este motivo. A melhor coisa é **NÃO** usar a opção Satisfy em casos que só requerem autenticação ou usar Satisfy all (que terá o mesmo efeito de não usá-la, hehehe).

A falta de atenção nisto pode comprometer silenciosamente a segurança de seu sistema.

5.4 O arquivo .htaccess

O arquivo `.htaccess` deve ser colocado no diretório da página que deverá ter suas permissões de acesso/listagem controladas. A vantagem em relação a inclusão direta de diretivas de acesso dentro do arquivo de configuração do Apache, é que o controle de acesso poderá ser definido pelo próprio webmaster da página, sem precisar ter acesso direto a configuração do Apache, que requerem privilégios de root.

Outro ponto fundamental é que não há necessidade de reiniciar o servidor Web, pois este arquivo é lido no momento de cada acesso ao diretório que controla. O nome do arquivo `OverRide` pode ser definido através da diretiva `AccessFileName` no arquivo de configuração do Apache, `.htaccess` é usado como padrão.

O controle de que opções estarão disponíveis no `.htaccess` são definidas na diretiva `AllowOverride` que pode conter o seguintes parâmetros:

- **None** O servidor não buscará o arquivo `.htaccess` nos diretórios
- **All** O servidor utilizará todas as opções abaixo no arquivo `.htaccess`
- **AuthConfig** Permite o uso de diretivas de autenticação (`AuthDBMGroupFile`, `AuthDBMUserFile`, `AuthGroupFile`, `AuthName`, `AuthType`, `AuthUserFile`, `Require`, etc.).
- **FileInfo** Permite o uso de diretivas controlando o tipo de documento (`AddEncoding`, `AddLanguage`, `AddType`, `DefaultType`, `ErrorDocument`, `LanguagePriority`, etc.).
- **Indexes** Permite o uso de diretivas controlando a indexação de diretório (`AddDescription`, `AddIcon`, `AddIconByEncoding`, `AddIconByType`, `DefaultIcon`, `DirectoryIndex`, `FancyIndexing`, `HeaderName`, `IndexIgnore`, `IndexOptions`, `ReadmeName`, etc.).
- **Limit** Permite o uso de diretivas controlando o acesso ao computador (`allow`, `deny` e `order`).
- **Options** Permite o uso de diretivas controlando características específicas do diretório (*Options* e *XBitHack*).

OBS: Não tem sentido usar a opção AllowOverride dentro da diretiva <Location>, ela será simplesmente ignorada.

Para acesso ao arquivo .htaccess do diretório /var/www/focalinux, o Apache buscará os arquivos .htaccess na seqüência: /.htaccess, /var/.htaccess, /var/www/.htaccess, /var/www/focalinux/.htaccess, qualquer diretiva que não exista no .htaccess do diretório /var/www/focalinux terá seu valor definido pela diretiva dos arquivos .htaccess dos diretórios anteriores. Somente após esta seqüência de checagens o acesso ao documento é permitido (ou negado).

Por este motivo, muitos administradores decidem desativar completamente o uso de arquivos .htaccess no diretório raiz e habilitar somente nos diretórios especificados pela diretiva <Directory> no arquivo de configuração do Apache, evitando brechas de segurança na manipulação destes arquivos (esta é uma boa idéia a não ser que se dedique 24 horas somente na administração do seu servidor Web e conheça toda sua estrutura hierárquica de segurança:

```
<Directory />
```

```
AllowOverride none
```

```
</Directory>
```

```
<Directory /var/www>
```

```
AllowOverride limit authconfig indexes
```

```
</Directory>
```

Na especificação acima, o arquivo .htaccess será procurado no diretório /var/www e seus sub-diretórios, usando somente opções que controlam a autorização de acesso (limit), autenticação e opções (authconfig) e de

indexação de documentos (indexes).

Alguns exemplos do uso do arquivo .htaccess:

Para permitir o acesso direto de usuários da rede 192.168.1.* diretamente, e requerer senha de acesso para outros usuários, o seguinte arquivo .htaccess deve ser criado no diretório /var/www:

```
Order deny,allow
```

```
allow from 192.168.1.0/24
```

```
deny from all
```

```
AuthName "Acesso a página Web principal da Empresa"
```

```
AuthType basic
```

```
AuthUserFile /var/cache/apache/senhas
```

```
Require valid-user
```

```
Satisfy any
```

Note que a sintaxe é exatamente a mesma das usadas na diretivas de acesso, por este motivo vou dispensar explicações detalhadas a respeito.

ATENÇÃO: A diretiva Options Indexes deverá ser especificada no AllowOverRide e não no arquivo .htaccess. Agora você já sabe o que fazer se estiver recebendo erros 500 ao tentar acessar a página (Erro interno no servidor)...

5.5 Usando a diretiva SetEnvIf com Allow e Deny

É possível especificar o acesso baseado em variáveis de ambiente usando a diretiva SetEnvIf, isto lhe permite controlar o acesso de acordo com o conteúdo de cabeçalhos HTTP. A sintaxe é a seguinte:

```
SetEnvIf [atributo] [expressão] [variável]
```

Isto poder ser facilmente interpretado como: Se o "atributo" especificado conter a "expressão", a "variável" será criada e armazenará o valor verdadeiro. Veja abaixo:

```
SetEnvIf User-Agent ".*MSIE*" EXPLoder
```

```
<Directory /var/www>
```

```
Order deny,allow
```

```
allow from all
```

```
deny from env=EXPLoder
```

```
</Directory>
```

Se o Navegador (campo User-Agent do cabeçalho http) usado para acessar a página for o Internet Explorer, a variável EXPLoder será criada e terá o valor verdadeiro (porque a expressão de SetEnvIf conferiu com a expressão).

Note o uso de "deny from env=VARIÁVEL". Neste caso se o navegador for o Internet Explorer, o acesso será bloqueado (pois o navegador conferiu, assim a variável EXPLoder recebeu o valor verdadeiro).

É permitido especificar as diretivas de acesso normais junto com especificação de variáveis de ambiente, basta separá-los com espaços. Uma descrição completa dos cabeçalhos HTTP, conteúdo e parâmetros aceitos por cada um são descritos na RFC 2068.

5.6 A diretiva <Limit>

Esta diretiva é semelhante a <Directory> mas trabalha com métodos HTTP (como GET, PUT, POST, etc) ao invés de diretórios. A diretiva <Limit> pode ser usada dentro da diretiva de acesso <Directory>, <Location>, mas nenhuma diretiva de controle de acesso pode ser colocada dentro de <Limit>.

Os métodos HTTP válidos são: GET, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK e UNLOCK. Note que os métodos são case-sensitive. Por exemplo:

```
<Directory /var/www>
```

```
Option Indexes
```

```
<Limit POST PUT DELETE>
```

```
Order deny,allow
```

```
allow from 192.168.1.0/24
```

```
deny from all
```

```
</Limit>
```

```
</Directory>
```

Somente permitem o uso dos métodos POST, PUT, DELETE de máquinas da rede interna.

OBS1: Se o método GET é bloqueado, o cabeçalho HTTP também será bloqueado.

OBS2: A diretiva de acesso <Limit> somente terá efeito na diretiva <Location> se for especificada no arquivo de configuração do servidor web. A diretiva <Location> simplesmente é ignorada nos arquivos .htaccess...

Este abaixo é usado por padrão na distribuição Debian para restringir para somente leitura o acesso aos diretórios de usuários acessados via módulo mod_userdir:

```
<Directory /home/*/public_html>

    AllowOverride FileInfo AuthConfig Limit

    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

    <Limit GET POST OPTIONS PROPFIND>

        Order allow,deny

        Allow from all

    </Limit>

    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>

        Order deny,allow

        Deny from all

    </Limit>
```

</Directory>

5.7 Diretiva <LimitExcept>

Esta diretiva é semelhante a <Limit>, mas atinge todos os métodos HTTP, menos os especificados.

6 Definindo documentos de erro personalizados

Adaptado do [Guia Foca GNU/Linux Avançado – Capítulo 11](#).

Documentos de erro personalizados são definidos através da diretiva `ErrorDocument`. É possível especificar códigos de erros que serão atendidos por certos documentos ou colocar esta diretiva dentro de blocos de controle de acesso `<Directory>`, `<Location>` ou `<VirtualHost>` para que tenham mensagens de erro personalizadas, ao invés da padrão usada pelo servidor `httpd`.

`ErrorDocument` [*código de erro*] [*documento*]

Onde:

- **código de erro** Código de erro da mensagem (veja [Códigos HTTP, Section 11.15](#) como referência). O código de erro `401` deve referir-se a um arquivo local.
- **documento** Documento, mensagem de erro ou redirecionamento que será usado no servidor caso aquele código de erro seja encontrado:

Para definir uma mensagem de erro padrão para todo servidor web, basta colocar a diretiva `ErrorDocument` fora das diretivas que controlam o acesso a diretórios e virtual hosts (o início do arquivo `httpd.conf` é ideal).

Exemplos:

- `ErrorDocument 404 /cgi-bin/erros404.pl` Direciona para um script em Perl que manda um e-mail ao administrador falando sobre o link quebrado e envia o usuário a uma página de erro padrão.
- `ErrorDocument 404 /naoencontrada.html` Direciona o usuário para o arquivo `naoencontrada.html` (dentro de `DocumentRoot`) quando ocorrer o erro 404. Note que o diretório / levado em consideração é o especificado pela diretiva `DocumentRoot`.

- `ErrorDocument 500 "Erro Interno no servidor"` Mostra na tela a mensagem "Erro Interno no servidor" quando ocorrer o erro 500.
- `ErrorDocument 401 /obtendoacesso.html` Direciona o usuário ao arquivo explicando como obter acesso ao sistema.
- `ErrorDocument 503 http://www.debian.org/error.html` Redireciona o usuário a URL especificada.
- `ErrorDocument 403 "Acesso negado"` Mostra na tela a mensagem "Acesso negado" no caso de erros 403.

7 Módulos DSO

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

Os módulos *DSO* permitem adicionar/remover características do Apache sem necessidade de recompilar todo o servidor web, assim interrompendo o serviço para a atualização dos arquivos. Módulos de programas terceiros também podem ser compilados e adicionado sem problemas através deste recurso.

Os módulos são carregados através da diretiva `LoadModule` no arquivo de configuração. O formato é o seguinte:

```
LoadModule [nome_do_modulo] [caminho_do_arquivo_so]
```

- **nome_do_modulo** Especifica o nome do módulo, não deve conter espaços.
- **caminho_do_arquivo_so** Define a localização do arquivo que contém o módulo especificado. Por padrão os módulos estão localizados em `/usr/lib/apache/[versão]`

A posição em que os módulos aparecem podem ter influência em seu funcionamento, alguns requerem que sejam especificados antes de outros módulos para funcionarem corretamente (como o módulo `php3_module`, que deve ser carregado antes de qualquer módulo de controle de CGI's). Leia a documentação específica sobre o módulo em caso de dúvidas, os módulos que acompanham o Apache são documentados em detalhes no manual do Apache.

Para usar uma característica/diretiva/opção do Apache que dependa de um certo módulo, obviamente você deverá carregar o módulo correspondente (em caso de dúvidas, leia a documentação sobre o módulo). Veja a seção “12.1 O arquivo `httpd.conf`” para exemplos do uso da diretiva `LoadModule`.

Por exemplo, se você quiser utilizar as diretivas de autorização (`allow`, `deny`, `order`) deverá ter o módulo `mod_access` carregado, para usar as diretivas de autorização (`authname`, `authuserfile`, `authtype`, *etc*) deverá ter

o módulo `mod_auth` carregado. Mais detalhes podem ser encontrados na seção “5.1 Autorização”. **OBS:** O suporte a *DSO* atualmente só está disponível para plataforma UNIX e seus derivados, como o GNU/Linux.

Também é possível ativar certas diretivas verificando se o módulo correspondente estiver ou não carregado através da diretiva `IfModule`:

```
<IfModule mod_userdir.c>
```

```
UserDir disabled root
```

```
UserDir public_html
```

```
</IfModule>
```

Nas linhas acima, as diretivas `UserDir` somente serão executadas se o módulo `mod_userdir.c` estiver carregado através da diretiva `LoadModule`.

Segue abaixo uma lista de módulos padrões que acompanham do Apache, os módulos marcados com "*" são ativados por padrão:

- Criação de Ambiente
 - mod_env* Ajusta variáveis de ambiente para scripts CGI/SSI
 - mod_setenvif* Ajusta variáveis de ambiente de acordo com cabeçalhos http
 - mod_unique_id Gera identificadores únicos para requisições

- Decisão de tipo de conteúdo de arquivos
 - mod_mime* Determinação de tipo/encodificação do conteúdo (configurado)
 - mod_mime_magic Determinação de tipo/encodificação do conteúdo (automático)
 - mod_negotiation* Seleção de conteúdo baseado nos cabeçalhos "HTTP Accept*"

- Mapeamento de URL
 - mod_alias* Tradução e redirecionamento de URL simples
 - mod_rewrite Tradução e redirecionamento de URL avançado
 - mod_userdir* Seleção de diretórios de recursos por nome de usuário
 - mod_speling Correção de URLs digitadas incorretamente
 - mod_vhost_alias Suporte para virtual hosts dinamicamente configurados em massa.

- Manipulação de Diretórios
 - mod_dir* Manipulação de Diretório e arquivo padrão de diretório
 - mod_autoindex* Geração de índice automático de diretório

- Controle de Acesso
 - mod_access* Controle de acesso por autorização (usuário, endereço, rede)
 - mod_auth* Autenticação HTTP básica (usuário, senha)
 - mod_auth_dbm Autenticação HTTP básica (através de arquivos NDBM do Unix)
 - mod_auth_db Autenticação HTTP básica (através de arquivos Berkeley-DB)

- mod_auth_anon Autenticação HTTP básica para usuários no estilo anônimo
 - mod_auth_digest Autenticação MD5
 - mod_digest Autenticação HTTP Digest
- Respostas HTTP
- mod_headers Cabeçalhos de respostas HTTP (configurado)
 - mod_cern_meta Cabeçalhos de respostas HTTP (arquivos no estilo CERN)
 - mod_expires Respostas de expiração HTTP
 - mod_asis* Respostas HTTP em formato simples (raw)
- Scripts
- mod_include* Suporte a Includes no lado do servidor (SSI – Server Sides Includes)
 - mod_cgi* Suporte a CGI (Common Gateway Interface)
 - mod_actions* Mapeia scripts CGI para funcionarem como 'handlers' internos.
- Manipuladores de conteúdo Interno
- mod_status* Visualiza status do servidor em tempo de execução.
 - mod_info Visualiza sumário de configuração do servidor.
- Registros de Requisições
- mod_log_config* Registro de requisições personalizáveis
 - mod_log_agent Registro especializado do User-Agent HTTP (depreciado)
 - mod_log_refer Registro especializado do Referrer HTTP (depreciado)
 - mod_usertrack Registro de cliques de usuários através de Cookies HTTP
- Outros

Apache

- `mod_imap*` Suporte a Mapeamento de Imagem no lado do servidor.
 - `mod_proxy` Módulo de Cache do Proxy (HTTP, HTTPS, FTP).
 - `mod_so` Inicialização do Dynamic Shared Object (DSO)
- Experimental
 - `mod_mmap_static` Cache de páginas freqüentemente servidas via `mmap()`
 - Desenvolvimento
 - `mod_example` Demonstração da API do Apache (somente desenvolvedores)

8 Sistema de Log do Apache

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

O Apache é bem flexível na especificação do que será registrado em seus arquivos de log, possibilitando utilizar um arquivo de log único, diversos arquivos de logs registrando cada evento ocorrido no sistema (conexão, navegador, bloqueio de acesso, erros, etc) incluindo os campos que deseja em cada arquivo e a ordem dos campos em cada um deles.

Enfim qualquer coisa pode ser especificada de forma que atenda as suas necessidades particulares de logging.

8.1 A diretiva AgentLog

AgentLog arquivo/pipe indica o nome do arquivo que registrará o nome do navegador que está acessando a página (conteúdo do cabeçalho User-Agent). É possível usar o pipe "|" para direcionar os erros para um programa de formatação ou processamento. **ATENÇÃO:** Se um programa for usado como pipe, ele será executado sob o usuário que iniciou o apache. Revise o código fonte do programa para ter certeza que não contém falhas que possam comprometer a segurança de seu sistema.

Exemplo:

```
AgentLog /var/log/apache/agent.log
```

8.2 A diretiva ErrorLog

ErrorLog arquivo/pipe especifica o arquivo que registrará as mensagens de erro do servidor Apache. É possível usar o pipe "|" para direcionar os erros para um programa de formatação ou processamento.

Exemplo:

```
ErrorLog /var/log/apache/errors.log
```

8.3 A diretiva CustomLog

Permite especificar onde os logs serão gravados para os arquivos de logs personalizados. Esta diretiva também aceita apelidos definidos pela diretiva *LogFormat*.

```
CustomLog [arquivo/pipe] [formato/nome]
```

Onde:

- `arquivo/pipe` Arquivo de log personalizado ou pipe.

`formato/nome` Especifica o formato do arquivo de log (da mesma forma que o especificado na opção *LogFormat*). Deverá ser especificado entre "aspas" caso tiver espaços. Veja a seção “0

- A diretiva *LogFormat*” para maiores detalhes.

Ao invés de especificar o formato, também é possível usar um apelido definido pela opção *LogFormat* ([veja “0](#)

[A diretiva *LogFormat*”](#)), neste caso os parâmetros definidos pelo *LogFormat* para "nome" serão atribuídos a diretiva *CustomLog*.

Exemplos:

```
CustomLog /var/log/apache/common.log "%h %l %u %t \"%r\" %>s %b"
```

```
CustomLog /var/log/apache/common.log common
```

8.4 A diretiva RefererLog

RefererLog [arquivo/pipe] indica que arquivo/pipe registrará os campos Referer do cabeçalho HTTP. Esta diretiva é mantida por compatibilidade com o servidor web NCSA 1.4.

A configuração padrão do Apache usa uma diretiva alternativa para a especificação do *referer* que é a seguinte:

```
LogFormat "%{Referer}i -> %U" referer
```

```
CustomLog /var/log/apache/referer.log referer
```

Exemplo:

```
RefererLog /var/log/apache/referer.log
```

8.5 A diretiva RewriteLog

RewriteLog: [arquivo/pipe] indica o arquivo/pipe que registrará qualquer regravação de URL feita pelo Apache.

OBS: Não é recomendável direcionar o nome de arquivo para /dev/null como forma de desativar este log, porque o módulo de regravação não cria a saída para um arquivo de log, ele cria a saída de log internamente. Isto somente deixará o servidor lento. Para desativar este registro, simplesmente remova/comente a diretiva RewriteLog ou use a opção RewriteLogLevel 0.

Exemplo:

```
RewriteLog "/usr/local/var/apache/logs/rewrite.log"
```

8.6 A diretiva RewriteLogLevel

RewriteLogLevel [num] especifica os detalhes que serão incluídos no registro da opção RewriteLog, os valores permitidos estão entre 0 e 9. Se for usado 0, o registro do RewriteLog é totalmente desativado (esta é a padrão). **OBS:** Qualquer valor acima de 2 deixa o servidor Web cada vez mais lento devido ao processamento e a quantidade de detalhes registrados no arquivo especificado por RewriteLog.

8.7 A diretiva ScriptLog

ScriptLog [arquivo] especifica o nome do arquivo de log que receberá as mensagens de erros gerados por scripts CGI executados no servidor. Esta opção é controlada pelo módulos mod_cgi.

Os arquivos de log serão abertos por um sub-processo rodando com as permissões do usuário especificado na diretiva "user".

OBS: Esta opção somente é recomendada como depuradora de scripts CGI, não para uso contínuo em servidores ativos.

Exemplo:

```
ScriptLog /var/log/apache/cgiscrpts.log
```

8.8 A diretiva ScriptLogBuffer

ScriptLogBuffer especifica o tamanho do cabeçalho PUT ou POST gravado no arquivo especificado por ScriptLog. O valor padrão é 1024 bytes. Esta opção é controlada pelo módulos mod_cgi

Exemplo:

```
ScriptLogBuffer 512
```

8.9 A diretiva ScriptLogLength

ScriptLogLength: [tamanho] especifica o tamanho máximo do arquivo de log gerado pela opção ScriptLog. O valor padrão é 10385760 bytes (10.3MB). Esta opção é controlada pelo módulos mod_cgi

Exemplo:

```
ScriptLogLength 1024480
```

8.10 A diretiva TransferLog

TransferLog [arquivo/pipe] indica o arquivo que armazenará as transferências entre o servidor http e o cliente. Ela cria o arquivo de log com o formato definido pela opção LogFormat mais recente ou o formato padrão do arquivo de log do Apache.

Se omitido, o arquivo não será gerado

Exemplo:

TransferLog /var/log/apache/transferências.log

8.11 A diretiva LogFormat

LogFormat define os campos padrões do arquivo gerado pela opção TransferLog. O seu formato é o seguinte:

LogFormat [formato] [nome]

Quando o formato não é especificado, assume o valor padrão %h %l %u %t \"%r\" %s %b. A especificação do [nome] permite que você utilize o formato especificado em uma opção CustomLog ou outra diretiva LogFormat, facilitando a especificação do formato do log.

Os seguintes formatos são válidos:

- %b Bytes enviados, excluindo cabeçalhos HTTP.
- %f Nome do arquivo.
- %{FOOBAR}e O conteúdo da variável de ambiente FOOBAR.
- %h Máquina cliente.
- %a Endereço IP da máquina cliente.
- %A Endereço IP local. Muito útil em virtual hostings.
- %{Foobar}i O conteúdo de Foobar: linhas de cabeçalho na requisição enviada ao servidor.

- %l O nome de login remoto enviado pelo identd (se fornecido).
- %{Foobar}n O conteúdo de "FooBar" de outro módulo.
- %{Foobar}o O conteúdo de Foobar: linhas de cabeçalho na resposta.
- %p A porta do servidor servindo a requisição.
- %P A identificação do processo filho que serviu a requisição.
- %r A primeira linha da requisição.
- %s Status. Para requisições que foram redirecionadas internamente. Este é o status de uma requisição *original*. Use %s para a última.
- %t Hora, no formato do arquivo de log (formato inglês padrão).
- %{format}t Hora, no formato definido por strftime.
- %T O tempo necessário para servir a requisição, em segundos.
- %u Usuário remoto (através do auth, pode ser falso se o status de retorno (%s) for 401).
- %U O caminho da URL requisitada.
- %v O nome canônico definido por *ServerName* que serviu a requisição.
- %V O nome do servidor de acordo com a configuração de *UseCanonicalName*.

Exemplos:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v" full
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %P %T" debug
```

```
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
```

8.12 A diretiva LogLevel

Define o nível de alerta das mensagens que serão gravadas no arquivo especificado pela diretiva ErrorLog. Quando não é especificado, assume o nível "error" como padrão. Abaixo os parâmetros aceitos em sua respectiva ordem de importância:

- emerg O sistema está inutilizável.
- alert A ação deve ser tomada imediatamente.
- crit Condições críticas.
- error Condições de erro.
- warn Condições de alerta.
- notice Condição normal mas significativa.

- info Mensagens informativas.
- debug Mensagens do nível de depuração.

Note que os níveis são os mesmos usados pelo syslog. Quando um nível particular é especificado, as mensagens de todos os níveis de maior importância também serão registradas. Por exemplo, se o nível "info" for especificado, as mensagens com os níveis de "notice" e "warn" também serão registradas. É recomendado o uso de um nível de no mínimo crit.

8.13 A diretiva `Anonymous_LogEmail`

Se estiver como "on" a senha digitada será registrada no arquivo especificado por `ErrorLog`. Esta diretiva é *ativada* por padrão.

Exemplo:

```
Anonymous_LogEmail off
```

8.14 A diretiva `CookieLog`

Especifica o arquivo que será usado para registrar os cookies

OBS1: Caso o caminho do arquivo não for especificado nas diretivas, será assumido `DocumentRoot` como diretório padrão.

OBS2: Caso esteja usando o pipe, o dono do processo será o mesmo que iniciou o servidor WEB Apache. Tenha certeza do funcionamento do programa para não comprometer o seu sistema, e cuide para que ele não possa ser modificado indevidamente por outros usuários.

Exemplo:

CookieLog /var/log/apache/cookies.log

8.15 Relatório gráfico de acesso ao sistema

Apêndice A. O programa webalizer poderá ser instalado para gerar um relatório gráfico com a estatísticas de visitas por ano/mes/dia/hora usando os dados do access.log. Outra interessante característica são as estatísticas de códigos http (veja “Apêndice B, Licença de Publicação Livre

Esta é uma tradução não-oficial da Open Publication License versão 1.0, de 8 de junho de 1999, e não é substituto legal para a Licença original, disponível em <http://www.opencontent.org/openpub>. Entretanto, esta tradução poderá auxiliar pessoas que falem Português a entender melhor a licença. É permitido a qualquer pessoa copiar e distribuir cópias desse documento de licença, desde que sem a implementação de qualquer mudança.

OPEN PUBLIC LICENSE

Draft v1.0, 8 June 1999

I. Requisitos comuns às versões modificadas e não modificadas

Os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) podem ser reproduzidos e distribuídos no todo ou em parte, em qualquer meio físico ou eletrônico, desde que os termos desta licença estejam incluídos, e que esta licença ou uma incorporação dela por referência (com quaisquer das opções escolhidas pelo autor ou editor) estejam presentes na reprodução.

A forma apropriada para uma incorporação por referência deste livro é:

Copyright© 2002 Alfamídia Ltda. Este material somente poderá ser distribuído se sujeito aos termos e condições firmados na Licença de Livre Publicação (Open Publication License), versão 1.0 ou superior (a versão mais atual encontra-se disponível em <http://www.opencontent.org/openpub/>).

Esta referência, devidamente preenchida com os dados da publicação, deve ser seguida imediatamente com quaisquer opções escolhidas pelos autores ou editor do documento (consultar a seção *Termos opcionais*).

É permitida a redistribuição comercial de material licenciado pela Licença de Livre Publicação (Open Publication License).

Qualquer publicação no formato livro padrão (papel) requer obrigatoriamente a citação dos autores e editor originais. Os nomes dos autores e do editor devem aparecer em todas as superfícies externas do livro. Em todas as faces externas do livro, o nome do editor original deve estar impresso em tamanho tão grande quanto o título do trabalho, e citado como proprietário em relação àquele título.

II. Copyright

O *copyright* de todo trabalho protegido pela Licença de Livre Publicação (Open Publication License) pertence aos autores ou proprietários.

III. Escopo da licença

Os termos de licença a seguir aplicam-se a todos os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License), a não ser que explicitamente indicado no trabalho.

A mera adição de trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) ou partes de trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) em uma mesma mídia que contenha outros trabalhos ou programas não protegidos por essa licença não decorre em aplicação da Licença de Livre Publicação (Open Publication License) para esses outros trabalhos. O trabalho resultante deve explicitamente conter uma nota especificando a inclusão do material protegido pela Licença de Livre Publicação (Open Publication License) e o aviso de copyright apropriado.

APLICABILIDADE. Se alguma parte desta licença não puder ser aplicada em alguma jurisdição, as partes restantes deste documento continuam sendo aplicadas.

AUSÊNCIA DE GARANTIA. Os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) são fornecidos "como estão", sem garantias de qualquer tipo, explícita ou implícita, incluindo, mas não limitado a, as garantias implícitas de comercialização e conveniência para um propósito particular, ou garantia de não-infração.

IV. Requisitos para trabalhos modificados

Todas as versões modificadas de documentos cobertos por esta licença, incluindo traduções, antologias, compilações e documentação parcial, deve seguir os requisitos abaixo:

A versão modificada deve ser indicada como tal.

As pessoas que fizerem as modificações e as datas de modificação devem ser identificadas.

O reconhecimento dos autores e editor originais (se aplicável) deve ser mantido de acordo com as práticas acadêmicas usuais de citação.

O local da versão não-modificada do documento deve ser indicado.

Os nomes originais dos autores não devem ser utilizados para indicar ou garantir seu endosso ao documento resultante sem a autorização expressa dos autores.

V. Práticas recomendadas

Em adição aos requisitos desta licença, é solicitado e extremamente recomendado aos redistribuidores que:

Se os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) estiverem sendo distribuídos em impressos ou CD-ROM, os autores sejam informados por email, ao menos trinta dias antes, para que os autores tenham tempo de providenciar documentação atualizada. Esta notificação deve descrever as modificações introduzidas no documento, se existirem.

Todas as modificações substanciais (incluindo exclusões) devem ser marcadas claramente no documento, ou então descritas em um anexo ao documento.

Finalmente, mesmo não sendo obrigatório sob esta licença, é considerado de bom tom oferecer uma cópia sem ônus de todo o material modificado (impresso e CD-ROM) para os autores originais.

VI. Termos opcionais

Os autores e editores de documentos protegidos pela Licença de Livre Publicação (Open Publication License) podem escolher certas opções de licença simplesmente incluindo alguns parágrafos após a cópia da licença ou sua referência. Estas opções são consideradas parte da licença e devem ser incluídas com ela (ou com a referência a ela) nos trabalhos derivados.

As opções que se aplicam a este trabalho são:

A:É vedada a distribuição de versões com modificações substanciais deste documento sem a expressa permissão dos proprietários do direito autoral.

B:É vedada a distribuição deste trabalho ou qualquer derivado seu em qualquer formato de livro padrão (papel) sem a prévia autorização dos proprietários do direito autoral.

Políticas de Publicação Livre

(O texto a seguir não é considerado parte da licença.)

Os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) estão disponíveis e podem ser acessados na home page da Open Publication <http://works.opencontent.org/>.

Os autores de trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) podem incluir suas próprias licenças nesses trabalhos, desde que os termos dessa licença não sejam mais restritivos que os da Licença de Livre Publicação (Open Publication License).

Em caso de dúvidas sobre a Licença de Livre Publicação (Open Publication License), contactar David Wiley <dww2@opencontent.org> ou a lista de autores de publicações <opal@opencontent.org> via email.

Para se inscrever na lista de autores de publicações livres (Open Publication Author's List), mande um email para <opal-request@opencontent.org> com a palavra **subscribe** no corpo da mensagem.

Para enviar mensagens para a lista de autores de publicações livres (Open Publication Author's List), mande um email para **opal@opencontent.org** ou simplesmente responda a uma mensagem postada.

Para se desinscrever na lista de autores de publicações livres (Open Publication Author's List), mande um email para **opal-request@opencontent.org** com a palavra **unsubscribe** no corpo da mensagem.

Códigos de retorno HTTP”), onde é possível saber a quantidade de links quebrados existentes em nosso servidor (estes poderão ser detectados usando o pacote de análise de sites linbot). O webalizer também é compatível com os formatos de log do squid e proftpd. Na distribuição Debian ele pode ser instalado a partir do pacote webalizer e gera um relatório geral quando é executado sem opções.

9 Configurando o Apache como servidor proxy

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

O Apache pode ser configurado para funcionar como servidor proxy transparente para sua rede interna, possibilitando inclusive o uso de cache de disco. É possível se fazer conexões HTTP (incluindo SSL) e FTP. Através desta característica também é possível usar uma das características mais interessante desse servidor web: o redirecionamento de conexões para uma determinada URL para uma outra máquina, que pode ser um outro host remoto ou uma máquina da rede interna (não acessível diretamente via Internet).

O primeiro passo é ativar o módulo de proxy no arquivo httpd.conf, basta descomentar a linha:

```
# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
```

O seguinte bloco pode ser colocado no final do arquivo httpd.conf para configurar um servidor proxy para realizar conexões diretas (sem o uso de cache) e permitir o uso de servidores proxy em sua rede:

```
# Suporte a Proxy

#

<IfModule mod_proxy.c>

ProxyRequests off

ProxyRemote * http://debian:3128

ProxyBlock microsoft.com microsoft.com.br

NoProxy 192.168.1.0/24
```

```
ProxyDomain .gms.com.br
```

```
# Ativa/Desativa a manipulação de cabeçalhos HTTP/1.1 "Via:".
```

```
#
```

```
# ("Full" adiciona a versão do servidor Apache; "Block" remove todos os cabeçalhos
```

```
# de saída "Via:")
```

```
# Escolha uma das opções: Off | On | Full | Block
```

```
#
```

```
#ProxyVia On
```

```
#</IfModule>
```

Segue a explicação de cada uma das diretivas acima:

- `ProxyRequests [on/off]` Ativa (on) ou Desativa (off) o serviço de proxy do servidor Apache. Note que o módulo `libproxy.so` deve estar carregado para que o bloco `<IfModule libproxy.c>` seja processado. A desativação desta diretiva não afeta a diretiva `ProxyPass`.
- `ProxyRemote [origem] [URL]` Esta opção é útil para fazer o Apache redirecionar suas requisições para outro servidor proxy (como o squid ou o gateway da rede, caso o Apache esteja sendo executado em uma máquina interna). A origem pode ser uma URL completa (como `http://www.debian.org`), uma URL parcial (como `ftp`, `http`) ou `"*"` para que o redirecionamento seja sempre usado.

- `ProxyBlock` [padrão] Permite bloquear o acesso a endereços que contenham o padrão especificado. Podem ser especificadas palavras, máquinas, domínios, URLs separados por espaços. O Apache fará a resolução DNS no caso de endereços IP e fará o cache para requisições futuras.
- `NoProxy` [endereços] Permite especificar endereços Internos que não serão redirecionados para o servidor proxy especificado por `ProxyRemote`. Podem ser usados nomes de máquinas, endereços IP, subredes ou domínios separados por espaços.
- `ProxyDomain` [endereço] Especifica o endereço que será adicionado a URL caso seja recebida uma requisição que contenha somente um nome de máquina. É útil em redes Internas.

Note que quando o suporte a proxy não está ativado no Apache, qualquer endereço de URL externa levará à página definida pela diretiva `DocumentRoot`. Isto deixará de funcionar após configurar o serviço de proxy.

O uso do cache é interessante para acelerar as requisições HTTP da rede interna para a rede externa, desta forma, se uma requisição foi feita anteriormente, será descarregado o arquivo do disco rígido e assim evitar uma nova conexão externa (isto libera a rede para outras coisas). Para configurar um cache no serviço proxy, adicione as seguintes linhas no final do bloco anterior de proxy:

```
# As linhas abaixo ativam o cache do apache, o cache não funcionará ao menos que  
  
# CacheRoot seja especificado  
  
CacheRoot /var/spool/apache  
  
CacheForceCompletion 70  
  
CacheSize 5  
  
CacheGcInterval 3
```

CacheDefaultExpire 5

CacheMaxExpire 300

NoCache 192.168.1.0/24 a_domain.com outrodomínio.com.br outro.dominio.net

Cada diretiva acima possui o seguinte significado:

- **CacheRoot** Diretório base onde serão criados os outros diretórios de cache. O cache só será ativado se esta diretiva for definida.
- **CacheForceCompletion [num]** Se uma transferência for cancelada e passar de num%, o Apache continuará a transferência e armazenará o arquivo no cache. O valor padrão é 90.
- **CacheSize [num]** Define o tamanho máximo do diretório de cache do Apache, em KB. Não especifique um valor que tome mais de 70% do espaço em disco. O valor padrão é 5.
- **CacheGcInterval [num]** Define o tempo que o cache será checado em busca de arquivos maiores que o total do cache. Arquivos que ultrapassem o tamanho do cache são automaticamente eliminados.
- **CacheDefaultExpire [num]** Define o tempo que os documentos ficarão no cache, se foram transferidos através de protocolos que não suportam horas de expiração. O valor padrão é 1 hora.
- **CacheMaxExpire [num]** Define o tempo que os documentos permanecerão armazenados no cache (em horas). Esta opção ignora a hora de expiração do documento (caso fornecida). O valor padrão é 24 horas.
- **NoCache [endereços]** Permite especificar lista de palavras, máquinas, domínios, IP's que não serão armazenados no cache do Apache. Caso seja usado NoCache * o cache será desativado completamente. Note que o cache também pode ser desativado comentando a diretiva CacheRoot.

Se você desejar um servidor cache mais flexível, rápido, dinâmico, configurável (com possibilidade de uso de restrições baseadas em URL, tempo de acesso, autenticação), instale o squid e configure o apache para fazer forward de conexões para ele (veja “9.2 Redirecionamento de conexões no Apache”).

9.1 Controlando o acesso ao servidor proxy

Incluir o bloco abaixo no arquivo access.conf para definir o acesso dos serviços de proxy nas redes desejadas (se a sua configuração for aberta como padrão isto pode ser opcional):

```
# Acesso aos serviços proxy do apache
```

```
<Directory proxy:*>
```

```
    Order deny,allow
```

```
    Deny from all
```

```
    Allow from .seudominio.com.br
```

```
</Directory>
```

Para explicações sobre o processo de bloqueio acima, veja a seção “5.1 Autorização”.

9.2 Redirecionamento de conexões no Apache

Este recurso do Apache é interessante para criar clusters de servidores em sua rede interna. O que ele faz é pegar uma requisição a um determinado endereço e redireciona-lo a outra máquina e as respostas são repassadas ao servidor web (para o cliente a mesma máquina esta atendendo a requisição, para você o processamento das requisições esta sendo distribuído internamente na rede).

As seguintes diretivas são usadas para realizar o redirecionamento de conexões: ProxyPass e ProxyPassReverse

- ProxyPass [*diretório_da_url* [*outro_servidor:/diretório*] permite que a URL seja redirecionada para o servidor local e diretório especificado. Por exemplo, assumindo que o endereço principal de nosso servidor é <http://www.focalinux.org> e desejamos que a URL <http://www.focalinux.org/download> seja atendida por uma máquina localizada na nossa rede privada com o endereço <http://192.168.1.54>. Basta incluir a linha:

```
ProxyPass /download http://192.168.1.54
```

Qualquer requisição externa a <http://www.focalinux.org/download/iniciante> será atendida por <http://192.168.1.54/iniciante>.

- ProxyPassRemote [*diretório_da_url* [*outro_servidor:/diretório*] esta diretiva permite modificar o cabeçalho Location nas mensagens de respostas de redirecionamento enviadas pelo Apache. Isto permite que o endereço retornado seja o do servidor (que faz a interface externa com o cliente) e não da máquina do redirecionamento.

```
ProxyPass /download http://192.168.1.54
```

```
ProxyPassReverse /download http://192.168.1.54
```

Se a máquina 192.168.1.54 redirecionar a URL para <http://192.168.1.54/download/iniciante>, a resposta será modificada para <http://www.focalinux.org/download/iniciante> antes de ser retornada ao cliente.

10 Virtual Hosts

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

Virtual Hosts (sites virtuais) é um recurso que permite servir mais de um site no mesmo servidor. Podem ser usadas diretivas específicas para o controle do site virtual, como nome do administrador, erros de acesso a página, controle de acesso e outros dados úteis para personalizar e gerenciar o site. Existem 2 métodos de virtual hosts:

- **Virtual Hosts baseados em IP** Requer um endereço IP diferente para cada site. Este poderá ser um IP real (da interface de rede) ou um apelido, o que interessa é que deve haver um endereço IP diferente para cada site. O número de sites servidos estará limitado a quantidade de endereços IP disponíveis em sua classe de rede. O apache foi um dos primeiros servidores a incluir suporte a virtual hosts baseados em IP.
- **Virtual Hosts baseados em nome** Este utiliza nomes para identificar os sites servidos e requerem somente um endereço IP. Desta maneira é possível servir um número ilimitado de sites virtuais. O navegador do cliente deve suportar os cabeçalhos necessários para garantir o funcionamento deste recurso (praticamente todos os navegadores atuais possuem este suporte).

As explicações desta seção são baseadas na documentação do Apache.

10.1 Virtual hosts baseados em IP

Existem duas maneiras de rodar este tipo de host virtual: Através de daemons httpd separados ou em um único daemon httpd usando a diretiva `<VirtualHost>`.

As vantagens do uso de *daemons separados* para servir requisições é a proteção sob *UID* e *GID* diferente dos outros servidores, assim o administrador do *site1* não terá acesso ao `httpd.conf`, página do *site2* (porque ele estará rodando sob uma *UID* e *GID* diferentes e o acesso é restrito). Para usar este método, especifique a opção `-f [arquivo_cfg]` para utilizar um arquivo de configuração personalizado e a diretiva `Listen`

endereço:porta para dizer onde o servidor aguardará as requisições.

As vantagens do uso de um *mesmo daemon* para servir as requisições são: quando não há problema se os administradores de outros sites tenham acesso ao mesmo arquivo de configuração ou quando há a necessidade de servir muitas requisições de uma só vez (quanto menos servidores web estiverem em execução, melhor o desempenho do sistema). Abaixo um exemplo de configuração de virtual hosts servindo os sites `www.site1.com.br` e `www.site2.com.br`:

ServerAdmin webmaster@site.com.br

<VirtualHost www.site1.com.br>

ServerName www.site1.com.br

ServerAdmin site1@site1.com.br

DocumentRoot /var/www/www_site1_com_br

TransferLog /var/log/apache/site1/access.log

ErrorLog /var/log/apache/site1/error.log

User www-data

Group www-data

</VirtualHost>

```
<VirtualHost www.site2.com.br>  
  
ServerName www.site2.com.br  
  
DocumentRoot /var/www/www_site2_com_br  
  
TransferLog /var/log/apache/site2/access.log  
  
ErrorLog /var/log/apache/site2/error.log  
  
</VirtualHost>
```

Qualquer diretiva dentro de <VirtualHost> controlarão terão efeito no site virtual especificado. Quando uma diretiva não for especificada dentro de <VirtualHost>, serão usados os valores padrões especificados no arquivo de configuração do Apache (como a diretiva `ServerAdmin` `webmaster@site.com.br` que será usado como padrão na configuração de `www.site2.com.br`).

Digite `apache -S` para ver suas configurações de virtual hosts atual.

OBS1: Desative a diretiva `UseCanonicalName` off quando utilizar o recurso de máquinas virtuais, esta diretiva faz que o nome do servidor retornado usando o valor em `ServerName` quando o cliente digita um endereço qualquer.

OBS2: Utilize sempre que possível endereços IP em configurações críticas, assim os serviços não serão tão vulneráveis a possíveis falsificações ou erros. Leia a seção “10.3Segurança no uso de IP's em Virtual Hosts”.

OBS3: Não permita que outros usuários a não ser o root e o dono do processo Apache (especificado pela diretiva `User`) tenham acesso de gravação aos logs gerados pelo servidor, pois os dados podem ser apagados ou criados links simbólicos para binários do sistema que serão destruídos quando o Apache gravar dados. Alguns binários e bibliotecas são essenciais para o funcionamento do sistema.

10.2 Virtual hosts baseados em nome

Este método é idêntico ao baseado em IP, em especial adicionamos a diretiva `NameVirtualHost` para dizer qual é o endereço IP do servidor que está servindo os virtual hosts baseados em nome. Veja o exemplo de configuração:

```
NameVirtualHost 200.200.200.10:80
```

```
<VirtualHost 200.200.200.10>
```

```
ServerName www.site1.com.br
```

```
ServerAdmin admin1@site1.com.br
```

```
DocumentRoot /var/www/www_site1_com_br
```

```
TransferLog /var/log/apache/site1/access.log
```

```
ErrorLog /var/log/apache/site1/error.log
```

```
</VirtualHost>
```

```
<VirtualHost 200.200.200.10>
```

```
ServerName www.site2.com.br
```

```
ServerAdmin admin2@site2.com.br  
  
DocumentRoot /var/www/www_site2_com_br  
  
TransferLog /var/log/apache/site2/access.log  
  
ErrorLog /var/log/apache/site2/error.log  
  
</VirtualHost>
```

A diretiva NameVirtualHost diz que será usado virtual hosts baseados em nome servidos pela máquina com IP 200.200.200.10. Os parâmetros dentro do bloco das diretivas <VirtualHost> são específicas somente no site virtual especificado, caso contrário os valores padrões definidos no arquivo de configuração serão usados.

Digite `apache -S` para ver suas configurações de virtual hosts atual. Se sua intenção é criar um grande número de virtual hosts que serão servidos pela mesma máquina, o uso da expansão %0 e diretivas `VirtualDocumentRoot` e `VirtualScriptAlias` são recomendados:

```
NameVirtualHost 200.200.200.10:80  
  
<VirtualHost 200.200.200.10>  
  
VirtualDocumentRoot /var/www/%0  
  
VirtualScriptAlias /var/www/%0/cgi-bin  
  
TransferLog log/apache/site1/access.log  
  
ErrorLog log/apache/site1/error.log
```

</VirtualHost>

Agora crie os diretórios em /var/www correspondentes aos nomes de domínios que serão servidos por sua máquina:

```
mkdir /var/www/www.site1.com.br
```

```
mkdir /var/www/www.site2.com.br.
```

Note que sua máquina deverá estar com o DNS configurado para responder por estes domínios .

ATENÇÃO É importante que os endereços especificados nas diretivas ServerName (www.site1.com.br) resolvam o endereço IP da diretiva VirtualHost (200.200.200.10). Isto deve ser feito via DNS ou nos arquivos /etc/hosts.

OBS1: Utilize sempre que possível endereços IP em configurações críticas, assim os serviços não serão tão vulneráveis a possíveis falsificações ou erros. Leia a seção “10.3Segurança no uso de IP's em Virtual Hosts”.

OBS2: Não permita que outros usuários a não ser o root e o dono do processo Apache (especificado pela diretiva *User*) tenha acesso de gravação aos logs gerados pelo servidor. Pois os dados podem ser apagados ou criados links para binários do sistema que serão destruídos quando o apache gravar dados para os logs. Alguns binários e bibliotecas são essenciais para o funcionamento do sistema.

10.3 Segurança no uso de IP's em Virtual Hosts

Quando você está colocando um nome na diretiva de configuração do seu virtual hosts, está assumindo que ele resolverá o endereço IP corretamente (como www.site1.com.br => 200.200.200.10). Se por algum motivo o servidor DNS for modificado (por outra pessoa que tem acesso a isto), o endereço IP resolvido para o site www.site1.com.br poderá ser modificado para 200.200.200.20, isto redirecionará as requisições para outra máquina ao invés da máquina correta. Este tipo de ataque é chamado "DNS Spoofing" e o uso de endereço IP

(ao invés de nomes) praticamente evita que isto aconteça. Esta situação pode acontecer com a diretiva abaixo:

```
<VirtualHost www.gms.com.br>  
  
ServerName www.gms.com.br  
  
ServerAdmin gleydson@cipsga.org.br  
  
DocumentRoot /var/www/www_gms_com_br  
  
</VirtualHost>
```

Outra situação, que impede o funcionamento do servidor Web, é quando o servidor DNS está em manutenção ou por algum outro motivo não pode resolver o endereço IP de um nome especificado (como `www.site1.com.br`). O apache precisa saber qual é o seu endereço IP para ser executado. Veja a próxima modificação:

```
<VirtualHost 192.168.1.1>  
  
ServerName www.gms.com.br  
  
ServerAdmin gleydson@cipsga.org.br  
  
DocumentRoot /var/www/www_gms_com_br  
  
</VirtualHost>
```

Na configuração acima usamos o IP do servidor para especificar o virtual host. O apache tentará fazer o DNS reverso para determinar qual nome é servido por aquele endereço IP (`www.site1.com.br`). Se ele falhar, somente a seção `<VirtualHost>` correspondente será desativada. Isto já é uma melhoria sobre a primeira

configuração. O nome do servidor na diretiva `ServerName` garante que o servidor responda com o nome correto.

Para evitar ataques baseados em DNS siga os seguintes procedimentos de segurança:

1. Preferencialmente utilize o arquivo `/etc/hosts` para a resolução de nomes em máquinas locais (principalmente quando existe somente um administrador). É um método que evita diversas consultas ao servidor DNS (que pode deixar o acesso lento) e este arquivo é gerenciado pelo usuário `root`, isto evita o acesso de qualquer usuário para a falsificação de endereços.

Este arquivo também é útil caso a pesquisa DNS falhe (quando a ordem de pesquisa for do servidor DNS para o arquivo `hosts` no arquivo `/etc/host.conf`), pois de qualquer forma o nome será resolvido e o servidor Apache será executado.

2. Evite dar poderes a outros administradores manipularem seu próprio domínio DNS, não há nada que possa impedi-lo de modificar o endereço "X" para ser servido pelo IP "Y" desviando o tráfego para seu próprio servidor web. Se isto não for possível, siga as dicas abaixo para diminuir possíveis problemas.
3. Utilize endereços IP na diretiva `<VirtualHost>`.
4. Use endereços IP na diretiva `Listen`.
5. Use um endereço IP na diretiva `BindAddress`.
6. Sempre utilize o parâmetro `ServerName` em todas as diretivas `<VirtualHost>`, isto evita o retorno incorreto de nomes (que pode evitar/revelar fraudes).
7. Quando utilizar virtual hosts, crie uma diretiva `<VirtualHost _default_L:*>` usando uma diretiva `DocumentRoot` que não aponte para lugar algum. Esta diretiva será acessada quando nenhuma diretiva `VirtualHost` servir a requisição, conferindo com o endereço/ip.

11 Uso de criptografia SSL

Adaptado do Guia Foca GNU/Linux Avançado – Capítulo 11.

Esta seção é uma referência rápida para configuração e uso do módulo `apache-ssl` com o servidor Apache. Este módulo realiza a comunicação segura de dados (criptografada) via porta 443 (que é usada como padrão quando especificamos uma url iniciando com `https://`). A transmissão criptografada de dados é importante quanto temos dados confidenciais que precisamos transmitir como movimentação bancária, senhas, número de cartões de crédito, fazer a administração remota do servidor, etc. SSL significa *Secure Sockets Layer* (camada segura de transferência) e TLS *Transport Layer Security* (camada segura de Transporte).

A intenção aqui é fornecer explicações práticas para colocar um servidor Apache com suporte a SSL funcionando no menor tempo possível. Detalhes sobre funcionamento de certificados, métodos de criptografia, assinatura, etc. deverão ser buscadas na documentação deste módulo ou em sites especializados (é um assunto muito longo).

11.1 Servidor apache com suporte a ssl

Ao invés de utilizar o módulo `apache-ssl`, você poderá usar o pacote `apache-ssl`, ele nada mais é que um servidor Apache com o suporte SSL já incluso e não interfere no servidor Apache padrão, porque é executado somente na porta 443.

Se você tem um grande site com configurações de acesso personalizadas, ele trará mais trabalho de administração, pois as configurações e diretivas de restrições de acesso deverão ser copiadas para este servidor web. No entanto, ele é indicado para máquinas que serão servidores SSL dedicados ou quando não possui configurações especiais em seu servidor web principal.

Esta seção tem por objetivo a instalação do suporte ao módulo SSL (`mod_ssl`) no servidor Apache padrão.

11.2 Gerando um certificado digital

O certificado digital é a peça que garante a transferência segura de dados. Ele contém detalhes sobre a empresa que fará seu uso e quem o emitiu. Para gerar ou modificar um certificado digital, execute o comando `mod-ssl-makecert` e siga as instruções. O método de criptografia usado pelo certificado digital é baseado no conceito de chave pública/privada, a descrição sobre o funcionamento deste sistema de criptografia é feito em [Usando pgp \(gpg\) para criptografia de arquivos, Section 17.5](#).

OBS Não utilize acentos nos dados de seu certificado.

11.3 Exemplo de configuração do módulo `mod-ssl`

Abaixo uma configuração rápida para quem deseja ter um servidor com suporte a SSL funcionando em menor tempo possível (ela é feita para operar em todas as instalações e não leva em consideração o projeto de segurança de sua configuração atual do Apache). Note que todas as diretivas relacionadas com o módulo `mod_ssl` começam com o nome "SSL":

```
# Somente processa as diretivas relacionadas a SSL caso o módulo mod_ssl estiver

# carregado pela diretiva LoadModule

<IfModule mod_ssl.c>

# É necessário especificar as portas que o servidor Web aguardará conexões (normais e

# ssl).

Listen 80

Listen 443
```

Ativa o tratamento de conexões com o destino na porta 443 pela diretiva

VirtualHost abaixo

<VirtualHost _default_:443>

Ativa ou desativa o módulo SSL para este host virtual

SSLEngine on

Certificado do servidor

SSLCertificateFile /etc/apache/ssl.crt/server.crt

Chave privada de certificado do servidor.

SSLCertificateKeyFile /etc/apache/ssl.key/server.key

A linha abaixo força o fechamento de conexões quando a

conexão com o navegador Internet Explorer é interrompida. Isto

Apache

```
# viola o padrão SSL/TLS mas é necessário para este tipo de
# navegador. Alguns problemas de conexões de navegadores também
# são causados por não saberem lidar com pacotes keepalive.
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
```

```
</VirtualHost>
```

```
</IfModule>
```

```
#####
```

```
# Adicionalmente poderão ser especificadas as seguintes opções para modificar #
```

```
# o comportamento da seção SSL (veja mais detalhes na documentação do mod-ssl) #
```

```
#####
```

```
# Formato e localização do cache paralelo de processos da seção. O cache de seção é
```

```
# feito internamente pelo módulo mas esta diretiva acelera o processamento
```

```
# de requisições paralelas feitas por modernos clientes navegadores. Por padrão
# nenhum cache é usado ("none").
SSLSessionCache    dbm:/var/run/ssl-cache

# Localização do arquivo de lock que o módulo SSL utiliza para
# sincronização entre processos. O padrão é nenhum.
SSLMutex file:/var/run/ssl-mutex

# Especifica o método de embaralhamento de dados que será utilizado
# durante o início de uma seção SSL (startup) ou durante o processo
# de conexão (connect). Podem ser especificados "builtin" (é muito rápido
# pois consome poucos ciclos da CPU mas não gera tanta combinação aleatória), um
# programa que gera números aleatórios (com "exec") ou os dispositivos aleatórios
# /dev/random e /dev/urandom (com "file"). Por padrão nenhuma fonte
# adicional de números aleatórios é usada.
SSLRandomSeed startup builtin
```

```
SSLRandomSeed connect builtin
```

```
#SSLRandomSeed startup file:/dev/urandom 512
```

```
#SSLRandomSeed connect file:/dev/urandom 512
```

```
#SSLRandomSeed connect exec:/pub/bin/NumAleat
```

```
# Tipos MIME para download de certificados
```

```
AddType application/x-x509-ca-cert .cert
```

```
AddType application/x-pkcs7-crl .crl
```

```
# Tempo máximo de permanência dos objetos do cache acima. O valor padrão é
```

```
# 300 segundos (5 minutos).
```

```
SSLSessionCacheTimeout 300
```

```
# Versão do protocolo SSL que será usada. Podem ser especificadas
```

```
# SSLv2, SSLv3 TLSv1 ou all. O mais compatível com os navegadores atuais
```

```
# é o "SSLv2". Por padrão "all" é usado.
```

```
#SSLProtocol all
```

```
#SSLProtocol -all +SSLv3
```

```
# Registra detalhes sobre o tráfego neste arquivo. Mensagens de erro
```

```
# também são armazenadas no arquivo de registro padrão do Apache
```

```
SSLLog /var/log/apache/ssl-mod.log
```

```
# Nível das mensagens de log registradas por SSLLog
```

```
SSLLogLevel info
```

Algumas diretivas deste módulo podem fazer parte tanto da configuração global do servidor como diretivas de acesso (Directory, Location, .htaccess, veja a opção "Context" na documentação do mod_ssl).

11.4 Autorizando acesso somente a conexões SSL

Existem casos que precisa restringir o uso de conexões normais e permitir somente conexões via SSL (como por exemplo, dentro da diretiva de acesso que controla seu acesso a uma página com listagem de clientes). A opção `SSLRequireSSL` é usada para tal e deve ser usada dentro das diretivas de controle acesso:

```
<Directory /var/www/secure/clientes>
```

```
Options Indexes
```

```
Order deny,allow
```

```
Deny from evil.cracker.com
```

```
SSLRequireSSL
```

```
</Directory>
```

A diretiva acima *requer* que sejam feitas conexões SSL (porta 443 – https://) para acesso ao diretório /var/www/secure/clientes, qualquer conexão padrão não criptografada (feita na porta 80) será rejeitada com o erro 403.

OBS: A diretiva SSLRequireSSL podia ser colocada entre as condicionais "IfModule mod_ssl.c" mas o servidor web permitiria conexões não criptografadas se por algum motivo esse módulo não estivesse carregado. Na configuração acima, ocorrerá um erro e impedirá o funcionamento do servidor web caso ocorra algum problema com o mod_ssl.

11.5 Iniciando o servidor Web com suporte a SSL

Verifique se a configuração do Apache está ok com `apache -t`. Caso positivo, reinicie o servidor usando um dos métodos descritos na seção “2.1 Iniciando o servidor/reiniciando/recarregando a configuração”. O servidor web lhe pedirá a FraseSenha para descriptografar a chave privada SSL (esta senha foi escolhida durante o processo de criação do certificado).

Esta senha garante uma segurança adicional caso a chave privada do servidor seja copiada de alguma forma. Somente quem tem conhecimento da FraseSenha poderá iniciar o servidor com suporte a transferência segura de dados. Verifique se o virtual host está servindo as requisições na porta 443 com `apache -S`.

O único método para fazer o servidor web evitar de pedir a senha para descriptografar a chave privada é colocando uma senha em branco. Isto só é recomendado em ambientes seguros e o diretório que contém a

chave privada deverá ter somente permissões para o dono/grupo que executa o servidor Web. Qualquer outra permissão poderá por em risco a segurança da instalação caso a chave privada seja roubada. Depois disso, execute o comando:

```
# entre no diretório que contém a chave privada

cd /etc/apache/ssl.key

# renomeie a chave privada para outro nome

ren server.key server.key-Csenha

openssl rsa -in server.key-Csenha -out server.key
```

Digite a senha quando pedido. A chave original (com senha) estará gravada no arquivo `server.key-Csenha` e poderá ser restaurada se necessário. Reinicie o servidor Apache, desta vez ele não pedirá a senha.

OBS1: Tire uma cópia de segurança da chave privada original antes de executar esta operação.

OBS2: Não se esqueça de ajustar as permissões de acesso no diretório `/etc/apache/ssl.key` caso não utilize senha para proteger seu certificado digital.

12 Exemplo comentado de um arquivo de configuração do Apache

Adaptado do [Guia Foca GNU/Linux Avançado – Capítulo 11](#).

O exemplo abaixo foi retirado da distribuição Debian GNU/Linux, fiz sua tradução, modificações e inclui alguns comentários sobre as diretivas para deixá-lo mais de acordo com o conteúdo abordado pelo guia e mais auto-explicativo.

A configuração do Apache está distribuída nos arquivos `httpd.conf`, `srm.conf` e `access.conf`, e podem ser usados como modelo para a construção da configuração de seu servidor.

12.1 O arquivo `httpd.conf`

```
##  
  
## httpd.conf -- Arquivo de configuração do servidor httpd Apache  
  
##  
  
#  
  
# Baseado nos arquivos de configuração originais do servidor NCSA por Rob McCool.  
  
# Modificado para distribuição junto ao guia Foca GNU/Linux Avançado  
  
# http://www.metainfo.org/focalinux <gleydson@cipsga.org.br>
```

Apache

```
#  
  
# Este é o arquivo de configuração principal do servidor Apache. Ele contém as  
# diretivas de configuração que dão ao servidor suas instruções.  
# Veja <http://www.apache.org/docs/> para informações detalhadas sobre as  
# diretivas.  
  
#  
# NÃO leia simplesmente as instruções deste arquivo sem entender o que significam  
# e o que fazem, se não tiver certeza do que está fazendo consulte a documentação  
# on-line ou leia as seções apropriadas do guia. Você foi avisado.  
  
#  
# Após este arquivo ser processado, o servidor procurará e processará o arquivo  
# /etc/apache/srm.conf e então /etc/apache/access.conf  
# a não ser que você tenha modificado o nome dos arquivos acima através das  
# diretivas ResourceConfig e/ou AccessConfig neste arquivo.  
  
#  
# Configuração e nomes de arquivos de log: Se os nomes de arquivos que
```

Apache

```
# especificar para os arquivos de controle do servidor iniciam com uma
# "/", o servidor usará aquele caminho explicitamente. Se os nomes *não*
# iniciarem com uma "/", o valor de ServerRoot é adicionado — assim
# "logs/foo.log" com ServerRoot ajustado para "/usr/local/apache" será
# interpretado pelo servidor como "/usr/local/apache/logs/foo.log".
#
# Originalmente por Rob McCool
# modificado por Gleydson Mazioli da Silva para o guia Foca GNU/Linux

# Carga dos Módulos de Objetos Compartilhados:
# Para você ser capaz de usa a funcionalidade de um módulo que foi construído como
# um módulo compartilhado, será necessário adicionar as linhas 'LoadModule'
# correspondente a sua localização, assim as diretivas que os módulos contém
# estarão disponíveis _antes_ de serem usadas.

# Exemplo:
```

Apache

#

ServerType pode ser inetd, ou standalone. O modo Inetd somente é suportado nas

plataformas Unix. O modo standalone inicia o servidor como um daemon.

#

ServerType standalone

Se estiver executando a partir do inetd, vá até a diretiva "ServerAdmin".

Port: A porta que o servidor standalone escutará. Para portas < 1023, será

necessário o servidor funcionando como root inicialmente.

Port 80

#

HostnameLookups: Registra os nomes DNS dos clientes ou apenas seus endereços

Apache

IP's

ex., www.apache.org (on) ou 204.62.129.132 (off).

O valor padrão é off porque permitirá menos tráfego na rede. Ativando

esta opção significa que cada acesso de um cliente resultará em

NO MÍNIMO uma requisição de procura ao servidor de nomes (DNS).

#

HostnameLookups off

Caso desejar que o servidor http seja executado como um usuário ou grupo diferente

você deve executar o httpd inicialmente como root e ele modificará sua ID para a

especificada.

User/Group: O nome (ou #número) do usuário/grupo que executará o servidor httpd.

No SCO (ODT 3) use "User nouser" e "Group nogroup"

No HPUX você pode não será capaz de usar memória compartilhada como nobody, e

Apache

é sugerido que seja criado um usuário www e executar o servidor httpd como

este usuário, adequando as permissões onde necessárias.

User www-data

Group www-data

ServerAdmin: Seu endereço de e-mail, onde os problemas com o servidor devem ser

enviadas. Este endereço aparecerá nas mensagens de erro do servidor.

ServerAdmin gleydson@cipsga.org.br

#

ServerRoot: O topo da árvore de diretórios onde os arquivos de configuração do

servidor, erros, e log são mantidos.

#

NOTA: Se tiver a intenção de colocar isto em um sistema de arquivos montado

em um servidor NFS (ou outra rede) então por favor leia a documentação do

```
# LockFile  
  
# (disponível em <http://www.apache.org/docs/mod/core.html#lockfile>);  
# e se salvará de vários problemas.  
  
#  
# Não adicione uma barra no fim do caminho do diretório.  
  
#  
  
ServerRoot /etc/apache  
  
# BindAddress: Você pode usar esta opção em virtual hosts. Esta  
# opção é usada para dizer ao servidor que endereço IP escutar. Ele pode  
# conter ou "*", um endereço IP, ou um nome de domínio completamente qualificado  
# (FQDN). Veja também a diretiva VirtualHost.  
  
BindAddress *
```

Apache

```
#  
# Suporte a Objetos Compartilhados Dinamicamente (DSO – Dynamic Shared Object)  
#  
# Para ser capaz de usar a funcionalidade de um módulo que foi compilado como  
# um módulo DSO, você terá que adicionar as linhas 'LoadModule' correspondentes  
# nesta localização, assim as diretivas contidas nela estarão disponíveis  
# _antes_ de serem usadas. Por favor leia o arquivo README.DSO na distribuição  
# 1.3 do Apache para mais detalhes sobre o mecanismo DSO e execute o comando  
# "apache -l" para a lista de módulos já compilados (estaticamente linkados e  
# assim sempre disponíveis) em seu binário do Apache.  
#  
# Please keep this LoadModule: line here, it is needed for installation.  
# LoadModule vhost_alias_module /usr/lib/apache/1.3/mod_vhost_alias.so  
# LoadModule env_module /usr/lib/apache/1.3/mod_env.so  
LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config.so  
# LoadModule mime_magic_module /usr/lib/apache/1.3/mod_mime_magic.so
```

Apache

```
LoadModule mime_module /usr/lib/apache/1.3/mod_mime.so

LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so

LoadModule status_module /usr/lib/apache/1.3/mod_status.so

# LoadModule info_module /usr/lib/apache/1.3/mod_info.so

# LoadModule includes_module /usr/lib/apache/1.3/mod_include.so

LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so

LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so

LoadModule php3_module /usr/lib/apache/1.3/libphp3.so

LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so

# LoadModule asis_module /usr/lib/apache/1.3/mod_asis.so

# LoadModule imap_module /usr/lib/apache/1.3/mod_imap.so

# LoadModule action_module /usr/lib/apache/1.3/mod_actions.so

# LoadModule speling_module /usr/lib/apache/1.3/mod_speling.so

LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so

LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so

LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so
```

Apache

```
LoadModule access_module /usr/lib/apache/1.3/mod_access.so

LoadModule auth_module /usr/lib/apache/1.3/mod_auth.so

# LoadModule anon_auth_module /usr/lib/apache/1.3/mod_auth_anon.so

# LoadModule dbm_auth_module /usr/lib/apache/1.3/mod_auth_dbm.so

# LoadModule db_auth_module /usr/lib/apache/1.3/mod_auth_db.so

# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so

# LoadModule digest_module /usr/lib/apache/1.3/mod_digest.so

# LoadModule cern_meta_module /usr/lib/apache/1.3/mod_cern_meta.so

LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so

# LoadModule headers_module /usr/lib/apache/1.3/mod_headers.so

# LoadModule usertrack_module /usr/lib/apache/1.3/mod_usertrack.so

LoadModule unique_id_module /usr/lib/apache/1.3/mod_unique_id.so

LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so

# LoadModule sys_auth_module /usr/lib/apache/1.3/mod_auth_sys.so

# LoadModule put_module /usr/lib/apache/1.3/mod_put.so

# LoadModule throttle_module /usr/lib/apache/1.3/mod_throttle.so
```

Apache

```
# LoadModule allowdev_module /usr/lib/apache/1.3/mod_allowdev.so
# LoadModule auth_mysql_module /usr/lib/apache/1.3/mod_auth_mysql.so
# LoadModule pgsql_auth_module /usr/lib/apache/1.3/mod_auth_pgsql.so
# LoadModule eaccess_module /usr/lib/apache/1.3/mod_eaccess.so
# LoadModule roaming_module /usr/lib/apache/1.3/mod_roaming.so

#
# ExtendedStatus: Controla de o Apache gerará detalhes completos de status
# (ExtendedStatus On) ou apenas detalhes básicos (ExtendedStatus Off) quando o
# manipulador (handler) "server-status" for usado. O padrão é Off.
#
ExtendedStatus on

#
# ErrorLog: A localização do arquivo de log de erros.
# Se não estiver especificando a diretiva ErrorLog dentro de <VirtualHost>,
```

Apache

```
# as mensagens de erros relativas aos hosts virtuais serão registradas neste
# arquivo. Se definir um arquivo de log de erros para <VirtualHost>, as
# mensagens relativas ao servidor controlados por ela serão registradas lá e
# não neste arquivo.

#

ErrorLog /var/log/apache/error.log

#

# LogLevel: Controla o número de mensagens registradas no ErrorLog.
# Facilidades possíveis incluem: debug, info, notice, warn, error, crit,
# alert, emerg.

# Veja as facilidades na seção do guia sobre o syslog para detalhes

#

LogLevel warn

# As seguintes diretivas definem alguns formatos de nomes que serão usadas com a
```

diretiva CustomLog (veja abaixo).

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %T %v" full
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %P %T" debug
```

```
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
```

#

A localização e formato do arquivo de log de acesso (definida pela diretiva

LogFormat acima).

Se não definir quaisquer arquivos de log de acesso dentro de um

<VirtualHost>, elas serão registradas aqui. Se for definida dentro

de <VirtualHost> o arquivo de log de acesso será registrado no

arquivo especificado na diretiva e não aqui.

```
#  
  
#CustomLog /var/log/apache/access.log common  
  
# Se você desejar ter um arquivo de log separado para o agent (navegador usado)  
# e referer, descomente as seguintes diretivas.  
  
#CustomLog /var/log/apache/referer.log referer  
#CustomLog /var/log/apache/agent.log agent  
  
# Se preferir um arquivo de log simples, com os detalhes de acesso, agent, e  
# referer (usando o formato combined da diretiva LogFile acima), use a seguinte  
# diretiva.  
  
CustomLog /var/log/apache/access.log combined  
  
#
```

Apache

```
# Incluir uma linha contendo a versão do servidor e um nome de host virtual
# para as páginas geradas pelo servidor (documentos de erro, listagens
# de diretórios FTP, saída dos módulos mod_status e mod_info, etc., exceto
# para documentos gerados via CGI). Use o valor "EMail" para também incluir
# um link mailto: para o ServerAdmin. Escolha entre "On", "Off" ou "EMail".
#
ServerSignature On

#

# PidFile: O arquivo que o servidor gravará os detalhes sobre seu PID quando
# iniciar.
#
PidFile /var/run/apache.pid

#

# ScoreBoardFile: Arquivo usado para armazenar detalhes do processo interno do
```

Apache

```
# servidor. Nem todas as arquiteturas requerem esta diretiva, mas se a sua
# requerer (você saberá porque este arquivo será criado quando executar o
# Apache) então você *deverá* ter certeza que dois processos do Apache não
# utilizam o mesmo arquivo ScoreBoardFile.

#
ScoreBoardFile /var/run/apache.scoreboard

#
# Na configuração padrão, o servidor processará este arquivo, o
# srm.conf e o access.conf neste ordem. Você pode fazer o servidor
# ignorar estes arquivos usando "/dev/null".

#
ResourceConfig /etc/apache/srm.conf
AccessConfig /etc/apache/access.conf

#
```

Apache

```
# A diretiva LockFile define o caminho do lockfile usado quando o servidor
# Apache for compilado com a opção USE_FCNTL_SERIALIZED_ACCEPT ou
# USE_FLOCK_SERIALIZED_ACCEPT. Esta diretiva normalmente deve ser deixada em seu
# valor padrão. A razão principal de modificá-la é no caso do diretório de logs
# for montado via um servidor NFS< pois o arquivo especificado em LockFile
# DEVE SER ARMAZENADO EM UM DISCO LOCAL. O PID do processo do servidor principal
# é automaticamente adicionado neste arquivo.
#
LockFile /var/run/apache.lock

# ServerName permite ajustar o nome de host que será enviado
# aos clientes, caso for diferente do nome real (por exemplo, se desejar usar
# www ao invés do nome real de seu servidor).
#
# Nota: Você não pode simplesmente inventar nomes e esperar que funcionem. O nome
# que definir deverá ser um nome DNS válido para sua máquina.
```

```
ServerName debian.meudominio.org
```

```
# UseCanonicalName: Com esta opção ligada, se o Apache precisar construir uma  
# URL de referência (uma url que é um retorno do servidor a uma requisição) ele  
# usará ServerName e Port para fazer o "nome canônico". Com esta opção desligada,  
# o Apache usará computador:porta que o cliente forneceu, quando possível.  
# Isto também afeta SERVER_NAME e SERVER_PORT nos scripts CGIs.  
#  
# Dependendo de sua configuração, principalmente em virtual hosts, é recomendável  
# deixá-la desativada ou com o valor DNS. O valor DNS obtém o nome do servidor  
# através de uma requisição DNS reversa do endereço IP (muito útil para virtual  
# hosts baseados em IP).  
UseCanonicalName off
```

```
# CacheNegotiatedDocs: Por padrão, o Apache envia Pragma: no-cache com cada
```

Apache

```
# documento que foi negociado na base do conteúdo. Isto permite dizer a  
# servidores proxy para não fazerem cache do documento. Descomentando a  
# seguinte linha desativa esta característica, e os proxies serão capazes  
# de fazer cache dos documentos.
```

```
#CacheNegotiatedDocs
```

```
# Timeout: O número de segundos antes de receber e enviar um time out
```

```
Timeout 300
```

```
# KeepAlive: Se vai permitir ou não conexões persistentes (mais que uma requisição  
# por conexão). Mude para "Off" para desativar.
```

```
KeepAlive On
```

```
# MaxKeepAliveRequests: O número máximo de requisições que serão permitidas
```

Apache

```
# durante uma conexão persistente. Mude para 0 para permitir uma quantidade  
# ilimitada. Nós recomendamos deixar este número alto, para obter a máxima  
# performance
```

```
MaxKeepAliveRequests 100
```

```
# KeepAliveTimeout: Número de segundos que aguardará a próxima requisição
```

```
KeepAliveTimeout 15
```

```
# Regulagem do tamanho de pool do servidor. Ao invés de fazer você adivinhar  
# quantos processos servidores precisará, o Apache adapta dinamicamente  
# de acordo com a carga que ele vê ---- isto é, ele tenta manter o número de  
# processos o bastante para manipular a carga atual, mas alguns poucos  
# servidores esparsos para manipular requisições transientes (ex. requisições  
# simultâneas múltiplas de um navegador Netscape simples).
```

Apache

```
# Ele faz isto verificando periodicamente quantos servidores estão  
# aguardando por uma requisição. Se lá existe menos que MinSpareServers,  
# ele cria um novo processo. Se existe mais que MaxSpareServers, ele  
# fecha alguns processos. Os valores abaixo estão adequados para muitos  
# sites
```

```
MinSpareServers 5
```

```
MaxSpareServers 10
```

```
# Número de servidores que serão iniciados — deve conter um valor razoável.
```

```
StartServers 5
```

```
# Limita o número total de servidores rodando, i.e., limita o número de clientes  
# que podem conectar simultaneamente — se este limite é sempre atingido,
```

Apache

os clientes podem ser BARRADOS, assim este valor NÃO DEVE SER MUITO PEQUENO.

Ele tem a intenção principal de ser um freio para manter um em execução com

uma performance aceitável de acordo com os requerimentos de construção e

carga calculada no servidor.

MaxClients 150

#

MaxRequestsPerChild: O número de requisições que cada processo tem permissão

de processar antes do processo filho ser finalizado. O filho será finalizado

para evitar problemas após uso prolongado quando o Apache (e talvez as

bibliotecas que utiliza) tomar memória e outros recursos. Na maioria dos

sistemas, isto realmente não é necessário, exceto para alguns (como o

Solaris) que possuem ponteiros notáveis em suas bibliotecas. Para estas

plataformas, ajuste para algo em torno de 10000 ou algo assim; uma

configuração de 0 significa ilimitado.

Apache

```
#  
# NOTA: Este valor não inclui requisições keepalive após a requisição  
# inicial por conexão. Por exemplo, se um processo filho manipula  
# uma requisição inicial e 10 requisições "keepalive" subsequentes,  
# ele somente contará 1 requisição neste limite.  
#  
MaxRequestsPerChild 30  
  
# Listen: Permite fazer o Apache escutar um IP determinado e/ou porta, em  
# adição a padrão. Veja também o comando VirtualHost  
  
#Listen 3000  
#Listen 12.34.56.78:80  
  
# VirtualHost: Permite o daemon responder a requisições para mais que um
```

Apache

```
# endereço IP do servidor, se sua máquina estiver configurada para aceitar pacotes
# para múltiplos endereços de rede. Isto pode ser feito com a opção de aliasing
# do ifconfig ou através de patches do kernel como o de VIF.

# Qualquer diretiva httpd.conf ou srm.conf pode ir no comando VirtualHost.
# Veja também a entrada BindAddress.

#<VirtualHost host.some_domain.com>
#ServerAdmin webmaster@host.some_domain.com
#DocumentRoot /var/www/host.some_domain.com
#ServerName host.some_domain.com
#ErrorLog /var/log/apache/host.some_domain.com-error.log
#TransferLog /var/log/apache/host.some_domain.com-access.log
#</VirtualHost>
```

Apache

```
# VirtualHost: Se você quiser manter múltiplos domínios/nomes de máquinas em sua
# máquina você pode ajustar o conteúdo de VirtualHost para eles.
# Por favor veja a documentação em <http://www.apache.org/docs/vhosts/>
# para mais detalhes antes de tentar configurar seus hosts virtuais.
# Você pode usar a opção de linha de comando '-S' para verificar sua configuração
# de hosts virtuais.

#
# Se desejar usar hosts virtuais baseados em nome, será necessário definir no
# mínimo um endereço IP (e número de porta) para eles.
#
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78

#
# Exemplo de um Host Virtual:
```

```
# Praticamente qualquer diretiva do Apache pode entrar na condicional
# VirtualHost.
#
#<VirtualHost ip.address.of.host.some_domain.com>
#  ServerAdmin webmaster@host.some_domain.com
#  DocumentRoot /www/docs/host.some_domain.com
#  ServerName host.some_domain.com
#  ErrorLog logs/host.some_domain.com-error.log
#  CustomLog logs/host.some_domain.com-access.log common
#</VirtualHost>

#<VirtualHost _default_:*>
#</VirtualHost>
```

12.2 O arquivo srm.conf

Neste arquivo são definidos o espaço de nomes que os usuários visualizarão no

Apache

```
# seu servidor http. Este arquivo também define configurações do servidor que  
# afetam como as requisições são servidas e como os resultados deverão ser  
# formatados.
```

```
# Veja os tutoriais em http://www.apache.org/ para mais detalhes
```

```
# DocumentRoot: O diretório principal onde você servira seus documentos.  
# Por padrão, todas as requisições são tomadas através deste diretório,  
# exceto links simbólicos e aliases que podem ser usados para apontar para  
# outras localizações no sistema de arquivos.
```

```
DocumentRoot /var/www
```

```
#
```

```
# UserDir: O nome do diretório que será adicionado ao diretório home do usuário  
# caso uma requisição ~usuário for recebida.
```

```
#
```

```
<IfModule mod_userdir.c>  
  
    # Linha abaixo por recomendação de segurança do manual do Apache  
  
    UserDir disabled root  
  
    UserDir public_html  
  
</IfModule>  
  
#  
  
# DirectoryIndex: Nome do arquivo ou arquivos que serão usados como índice do  
# diretório. Especifique mais de um arquivos separados por espaços ao invés  
# de um só um nome (como "index") para aumentar a performance do servidor.  
#  
  
<IfModule mod_dir.c>  
  
    DirectoryIndex index.html index.htm index.shtml index.cgi  
  
</IfModule>
```

```
#  
# Diretivas que controlam a exibição de listagem de diretórios geradas pelo servidor.  
#  
  
<IfModule mod_autoindex.c>  
  
#  
# FancyIndexing: se você deseja o padrão fancy index ou padrão para a indexação  
#     de arquivos no diretório. Usando FancyIndexing o servidor  
#     apache gerará uma listagem de arquivos que poderá ser  
#     ordenada, usar tipos de ícones e encoding, etc. Veja as  
#     próximas opções  
IndexOptions FancyIndexing  
  
#  
# As diretivas AddIcon* dizem ao servidor que ícone mostrar para um determinado
```

```
# arquivo ou extensão de arquivos. Estes somente são mostrados para os
# diretórios classificados através da opção FancyIndexing.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
```

```
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif */core
AddIcon /icons/deb.gif .deb Debian

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
```

```
# DefaultIcon é o ícone que será mostrado para aplicativos que não tiverem um  
# ícone explicitamente definido.
```

```
DefaultIcon /icons/unknown.gif
```

```
#
```

```
# AddDescription: isto lhe permite colocar uma curta descrição após um arquivo  
# nos índices gerados pelo servidor. Estes somente são mostrados para diretórios  
# com índices organizados usando a opção FancyIndexing.
```

```
# Formato: AddDescription "descrição" extensão
```

```
#
```

```
#AddDescription "GZIP compressed document" .gz
```

```
#AddDescription "tar archive" .tar
```

```
#AddDescription "GZIP compressed tar archive" .tgz
```

```
# ReadmeName é o nome do arquivo LEIAME que o servidor procurará como  
# padrão. Estes serão inseridos no fim da listagem de diretórios.
```

```
Formato: ReadmeName nome
```

```
#
```

```
# O servidor procurará primeiro por nome.html, incluído se ele for encontrado,  
# e então procurará pelo nome e incluirá ele como texto plano se encontrado..
```

```
ReadmeName README
```

```
# HeaderName é o nome do arquivo que deve ser colocado no topo do índice  
# de diretórios. As regras de procura de nome são as mesmas do arquivo
```

```
# README
```

```
HeaderName HEADER
```

```
#
```

```
# IndexIgnore: um conjunto de nomes de arquivos que a listagem de diretórios
# deve ignorar e não incluir na listagem. É permitido o uso de coringas
# como no interpretador de comandos.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

```
</IfModule>
```

```
# AccessFileName: O nome do arquivo que será procurado em cada diretório
# que contém detalhes sobre as permissões de acesso a um determinado
# diretório e opções de listagem. Tenha cuidado ao modificar o nome
# deste arquivo, muitas definições que trabalham em cima do nome
# .htaccess nos arquivos de configuração deverão ser modificados para
# não comprometer a segurança de seu servidor.
# Uma falta de atenção neste ponto poderá deixar este arquivo visível
# em qualquer listagem de diretórios facilmente...
```

AccessFileName .htaccess

TypesConfig especifica o arquivo de configuração que contém os tipos

usados pelo servidor

TypesConfig /etc/mime.types

#

DefaultType é o tipo MIME padrão que o servidor utilizará para um documento

caso ele não possa determinar seu conteúdo, como através de extensões

de arquivos. Se o servidor contém em sua maioria texto ou documentos em HTML,

"text/plain" é um bom valor. Caso a maioria do conteúdo seja binários, tal

como aplicativos ou fotos, o tipo mais adequado ao seu caso poderá ser

"application/octet-stream" para evitar que navegadores tentem exibir

aplicativos binários como se fossem texto.

Se desejar uma referência rápida sobre tipos mime, consulte o arquivo

```
# /etc/mime.types
#
DefaultType text/plain

#
# Document types.
#
<IfModule mod_mime.c>

# AddEncoding permite que alguns navegadores (Mosaic/X 2.1+, Netscape, etc)
# descompactem dados durante sua abertura. N
# Nota: Nem todos os navegadores suportam isto. Esqueça os nomes parecidos,
# as seguintes diretivas Add* não tem nada a ver com personalizações
# da opção FancyIndexing usada nas diretivas acima.

AddEncoding x-compress Z
```

```
AddEncoding x-gzip gz tgz
```

```
#
```

```
# AddLanguage: permite especificar o idioma do documento. Você pode
```

```
# então usar a negociação de conteúdo para dar ao navegador um
```

```
# arquivo no idioma solicitado.
```

```
#
```

```
# Nota 1: O sufixo não precisa ser o mesmo da palavra chave do
```

```
# idioma ——— estes com o documento em Polonês (no qual o
```

```
# código padrão da rede é pl) pode desejar usar "AddLanguage pl .po"
```

```
# para evitar confusão de nomes com a extensão comum de scripts
```

```
# scripts em linguagem Perl.
```

```
#
```

```
# Nota 2: As entradas de exemplos abaixo mostram que em alguns casos
```

```
# as duas letras de abreviação do 'Idioma' não é idêntico as duas letras
```

```
# do 'País' para seu país, como 'Danmark/dk' versus 'Danish/da'.
```

#

Nota 3: No caso de 'ltz' nós violamos a RFC usando uma especificação de

três caracteres. Mas existe um 'trabalho em progresso' para corrigir isto

e obter os dados de referência para limpar a RFC1766.

#

Danish (da) – Dutch (nl) – English (en) – Estonian (ee)

French (fr) – German (de) – Greek–Modern (el)

Italian (it) – Portugese (pt) – Luxembourggeois* (ltz)

Spanish (es) – Swedish (sv) – Catalan (ca) – Czech(cz)

Polish (pl) – Brazilian Portuguese (pt-br) – Japanese (ja)

#

AddLanguage da .dk

AddLanguage nl .nl

AddLanguage en .en

AddLanguage et .ee

AddLanguage fr .fr

```
AddLanguage de .de
AddLanguage el .el
AddLanguage it .it
AddLanguage ja .ja
# AddCharset ISO-2022-JP .jis
AddLanguage pl .po
# AddCharset ISO-8859-2 .iso-pl
AddLanguage pt .pt
AddLanguage pt-br .pt-br
AddLanguage ltz .lu
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz

# LanguagePriority: permite definir a prioridade para a exibição de
```

```
# documentos caso nenhum documento confira durante a negociação de
# conteúdo.
#
# Para fazer isto, especifique os idiomas em ordem de preferência de exibição
# de idiomas.
#
<IfModule mod_negotiation.c>
    LanguagePriority pt-br pt es en da nl et fr de el it ja pl ltz ca sv
</IfModule>

#
# AddType permite modificar o mime.types sem editar o arquivo, ou fazer
# a associação de arquivos a certos tipos de conteúdo.
#
# Por exemplo, o módulo PHP 3.x (que não faz parte da distribuição do
# Apache – veja http://www.php.net) tipicamente utiliza isto:
```

```
#  
#AddType application/x-httpd-php3 .php3  
#AddType application/x-httpd-php3-source .phps  
#  
# E para arquivos PHP 4.x use:  
#  
#AddType application/x-httpd-php .php  
#AddType application/x-httpd-php-source .phps  
  
AddType application/x-tar .tgz  
AddType image/bmp .bmp  
  
# hhtml  
AddType text/x-hhtml .hhtml  
  
#
```

```
# AddHandler permite mapear certas extensões de arquivos a programas
# "manipuladores" adequados a seu conteúdo. Estes podem ser construídos
# no servidor ou adicionados com o comando Action (veja abaixo).
#
# Se desejar usar includes no lado do servidor, ou servir diretórios
# com scripts CGI para fora, descomente as seguintes linhas.
#
# Para usar scripts CGI:
#
#AddHandler cgi-script .cgi .sh .pl
#
# Para usar arquivos html gerados através do servidor
#
#AddType text/html .shtml
#AddHandler server-parsed .shtml
```

```
#  
  
# Descomente as seguintes linhas para ativar a características de arquivos  
  
# send-as-is HTTP do servidor Apache  
  
#  
  
#AddHandler send-as-is asis  
  
  
#  
  
# Se desejar usar arquivos de mapas de imagens processadas no servidor, use  
  
#  
  
#AddHandler imap-file map  
  
  
#  
  
# Para ativar tipo de mapas, você poderá usar  
  
#  
  
#AddHandler type-map var
```

```
</IfModule>
```

```
# Fim dos tipos de documentos
```

```
# Preferências padrões de exibição de caracteres (veja http://www.apache.org/info/css-security/).
```

```
AddDefaultCharset on
```

```
AddDefaultCharsetName iso-8859-1
```

```
# Redirect permite dizer aos cliente que documentos não existem mais no seu servidor
```

```
# e a nova localização do documento.
```

```
# Format: Redirect nomeurl url
```

```
# "nomeurl" é o caminho especificado na url e "url" é a nova localização do
```

```
# documento
```

```
# Aliases: Inclua aqui quantos apelidos você desejar (sem limite) o formato é:
```

```
# Alias nomeurl nomereal  
  
# "nomeurl" é o caminho especificado na url e "nomereal" é a localização  
# do documento no sistema de arquivos local  
  
# Note que se você incluir uma / no fim de "nomeurl", então o servidor  
# requisitará que também esteja presente na URL.  
  
Alias /icons/ /usr/share/apache/icons/  
  
Alias /doc/ /usr/doc/  
  
Alias /focalinux /var/www/focalinux  
  
Alias /debian-br /var/www/debian-br/htdocs  
  
Alias /debian /pub/mirror/debian  
  
# ScriptAlias: Esta diretiva controla que diretórios contém scripts do servidor.  
  
# Format: ScriptAlias fakename realname
```

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

#

# Action: permite definir os tipos de mídia que executarão um script quando um
# arquivo que conferir for chamado. Isto elimina a necessidade de caminhos de URLs
# repetidas para processadores de arquivos CGI freqüentemente usados.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location

#

#
# MetaDir: especifica o nome do diretório no qual o apache procurará arquivos de
# detalhes do módulo mod_cern_meta. Os módulos meta contém cabeçalhos HTTP
# adicionais que serão incluídos durante o envio do documento.

#
#MetaDir .web
```

```
#  
# Resposta de erros personalizada (no estilo do Apache)  
# estas podem ser 3 tipos:  
#  
# 1) texto plano  
#ErrorDocument 500 "O servidor fez boo boo."  
# n.b. a aspa (") marca como texto, ela não será exibida  
#  
# 2) redirecionamentos locais  
#ErrorDocument 404 /missing.html  
# para redirecionar para a URL local /missing.html  
#ErrorDocument 404 /cgi-bin/missing_handler.pl  
# N.B.: É também possível redirecionar a um script o documento usando includes  
# do lado do servidor (server-side-include).  
#
```

```
# 3) redirecionamentos externos

#ErrorDocument 402 http://algum.outro_servidor.com/inscricao.html

# N.B.: Muitas das variáveis de ambientes associada com a requisição atual *não*
# estarão disponíveis para tal script.

#

# O módulo mod_mime_magic permite o servidor usar várias dicas através do conteúdo
# do arquivo para determinar o seu tipo. A diretiva MIMEMagicFile diz ao módulo
# onde as definições de dicas estão localizadas. O módulo mod_mime_magic não é
# parte do servidor padrão Apache (você precisará adicioná-lo manualmente com
# uma linha LoadModule (veja o parágrafo DSO na seção Ambiente Global no
# arquivo httpd.conf), ou recompile o servidor e inclua mod_mime_magic como
# parte de sua configuração), por este motivo ele está entre as condicionais
# <IfModule>. Isto significa que a diretiva MIMEMagicFile somente será processada
# caso o módulo estiver ativo no servidor.

#
```

```
<IfModule mod_mime_magic.c>
```

```
    MIMEMagicFile conf/magic
```

```
</IfModule>
```

```
<IfModule mod_setenvif.c>
```

```
    #
```

```
    # As seguintes diretivas modificam o funcionamento da resposta normal do
```

```
    # servidor HTTP.
```

```
    # A primeira diretiva desativa o keepalive para o Netscape 2.x e navegadores que
```

```
    # as falsificam. Existem problemas conhecidos com estas implementações de
```

```
    # navegadores. A segunda diretiva é para o MS IE 4.0b2 que tem uma implementação
```

```
    # defeituosa do HTTP/1.1 e não suporta adequadamente o keepalive quando ele
```

```
    # utiliza as respostas de redirecionamento 301 e 302.
```

```
    #
```

```
    BrowserMatch "Mozilla/2" nokeepalive
```

```
BrowserMatch "MSIE 4\0b2;" nokeepalive downgrade=1.0 force-response=1.0

#
# As seguintes diretivas desativam as respostas HTTP/1.1 para navegadores que
# violam a especificação HTTP/1.0 não sendo capaz de enviar uma resposta
# 1.1 básica.
#
BrowserMatch "RealPlayer 4\0" force-response=1.0
BrowserMatch "Java/1\0" force-response=1.0
BrowserMatch "JDK/1\0" force-response=1.0
</IfModule>

# Se o módulo Perl está instalado, isto será ativado.
<IfModule mod_perl.c>
Alias /perl/ /var/www/perl/
<Location /perl>
```

```
Options +ExecCGI  
SetHandler perl-script  
PerlHandler Apache::Registry  
</Location>  
</IfModule>
```

12.3 O arquivo access.conf

```
# access.conf: Configuração de acesso Global  
  
# Documentos on-line em http://www.apache.org/  
  
# Este arquivo define as configurações do servidor que afetam que tipos de  
# serviços são permitidos e em quais circunstâncias.  
  
# Cada diretório que o Apache possui acesso, pode ser configurado respectivamente  
# com quais serviços e características que podem ser permitidas e/ou bloqueadas  
# no diretório (e seus subdiretórios).
```

```
#  
# Primeiro a configuração restringe uma série de permissões  
<Directory />  
    Options SymLinksIfOwnerMatch  
    AllowOverride None  
# Order deny,allow  
# Deny from all  
</Directory>  
  
# Desse ponto em diante, é necessário especificar o que será permitido  
# caso contrário será bloqueado pelo bloco acima  
  
# Esta parte deve ser modificada para a localização do documento raiz do servidor.  
  
<Directory /var/www>
```

```
# A opção Options pode conter os valores "None", "All", ou quaisquer combinação
# de "Indexes", "Includes", "FollowSymLinks", "ExecCGI", ou "MultiViews".
#
# Note que "MultiViews" deve ser *explicitamente* especificada — "Options All"
# não a ativa (pelo menos não ainda).
```

```
Options Indexes FollowSymLinks Includes MultiViews
```

```
# Esta opção controla que opções os arquivos .htaccess nos diretórios podem ser
# substituídas. Pode também conter "All", ou qualquer combinação de "Options",
# "FileInfo", "AuthConfig", e "Limit"
```

```
AllowOverride None
```

```
# Controla quem pode obter materiais deste servidor. Leia a seção adequada no
# guia para mais explicações sobre a ordem de acesso, padrões e valores permitidos.
```

```
order allow,deny
```

```
allow from all
```

```
</Directory>
```

```
#
```

```
# O diretório "/usr/lib/cgi-bin" deve ser modificado para o diretório que
```

```
# possuem seus scripts CGI, caso tenha configurado o suporte a CGI's no
```

```
# servidor.
```

```
#
```

```
<Directory /usr/lib/cgi-bin/>
```

```
    AllowOverride None
```

```
    Options ExecCGI
```

```
    Order allow,deny
```

```
    Allow from all
```

```
</Directory>

#

# Permite ver relatórios de status e funcionamento do servidor web e
# processos filhos, através da URL http://servidor/server-status
# isto requer o módulo status_module (mod_status.c) carregado no arquivo
# httpd.conf

#

#<Location /server-status>

#  SetHandler server-status

#  Order deny,allow

#  Deny from all

#  Allow from .meudominio.org

#</Location>

#
```

```
# Permite relatório de configuração remota do servidor, através da URL
# http://servername/server-info
# Isto requer o módulo info_module (mod_info.c) carregado no arquivo
# httpd.conf
#
#<Location /server-info>
#  SetHandler server-info
#  Order deny,allow
#  Deny from all
#  Allow from .meudominio.org
#</Location>

# Visualização do diretório de ícones
<Directory /usr/share/apache/icons>
    Options Indexes MultiViews
    AllowOverride None
```

```
Order allow,deny

Allow from all

</Directory>

# O Debian Policy assume que /usr/doc é "/doc/" e linkado com /usr/share/doc,
# pelo menos para localhost.

<Directory /usr/doc>

Options Indexes FollowSymLinks

order deny,allow

deny from all

allow from 192.168.1.10/24

</Directory>

# Esta define a localização visualizável do monitor de status mod_throttle
```

```
#  
  
<location /throttle-info>  
    SetHandler throttle-info  
</location>  
  
#  
# As seguintes linhas previnem os arquivos .htaccess de serem mostrados nos  
# clientes Web. Pois os arquivos .htaccess freqüentemente contém detalhes  
# de autorização, o acesso é desabilitado por razões de segurança. Comente  
# estas linhas se desejar que seus visitantes vejam o conteúdo dos arquivos  
# .htaccess. Se modificar a diretiva AccessFileName acima, tenha certeza de  
# fazer as modificações correspondentes aqui.  
#  
# As pessoas também tendem a usar nomes como .htpasswd nos arquivos de senhas  
# a diretiva abaixo os protegerá também.  
#
```

```
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

#
# Controla o acesso a diretórios UserDir. As seguintes diretivas são um exemplo
# para um site onde estes diretórios estão restritos a somente-leitura. Veja
# detalhes sobre as opções de acesso, e limites na seção sobre controle
# de acesso do guia
#
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
```

```
    Allow from all

</Limit>

<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>

    Order deny,allow

    Deny from all

</Limit>

</Directory>

#

# As vezes ocorrem relatos de pessoas tentando abusar de uma falha antiga nos
# dias do Apache 1.1 (muitas páginas na Net documentam isso). Esta falha envolve
# um script CGI distribuído como parte do Apache. Descomentando estas linhas você
# poderá redirecionar estes ataques a um script de registro em phf.apache.org. Ou
# poderá gravar em sua própria máquina, usando o script support/phf_abuse_log.cgi.
#

#<Location /cgi-bin/phf*>
```

```
# Deny from all
# ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>

# Acesso aos serviços proxy do apache
#<Directory proxy:*>
# Order deny,allow
# Deny from all
# Allow from .your_domain.com
#</Directory>

# a seguinte diretiva permite o acesso a todos os usuários ao conteúdo da página
# do guia Foca GNU/Linux exceto os que possuem navegadores MSIE ;- )
# Veja a seção sobre restrições de acesso para detalhes sobre a diretiva de
# controle de acesso baseado no user-agent
```

```
SetEnvIf User-Agent MSIE EXPLoder
```

```
<Directory /var/www/focalinux>
```

```
Options Indexes
```

```
Order allow,deny
```

```
allow from all
```

```
deny from env=EXPLoder
```

```
ErrorDocument 403 "Explorer não entra, página com o conteúdo potencialmente perigoso ao Windows, use um navegador seguro para ter acesso a esta página ;-)"
```

```
</Directory>
```

```
# A diretiva abaixo somente permite acesso a leitura do arquivo
```

```
# h-supor-fonte.txt a pessoas que fornecerem o nome/senha corretos
```

```
# que constam no arquivo passwd1
```

```
# Este bloco contém um erro que é a localização do arquivo da senha em um
```

```
# diretório público, você deverá adaptá-lo se não quiser se ver em apuros.
```

```
#
```

```
# A permissão do diretório de nível superior prevalece sobre seus
# sub-diretórios no caso as permissões de /focalinux, a menos que
# sejam definidas opções de acesso específicas ao arquivo abaixo
<Location /focalinux/humor/h-supor-fonte.txt>
    AuthName "Piada de fonte de alimentação"
    AuthType basic
    AuthUserFile /home/gleydson/public_html/passwd1
    Require valid-user
# Satisfy all
</Location>

# Libera o acesso a localização /debian (acessada através de /pub/mirror/debian,
# definida no Alias acima)
<Location /debian>
    Options Indexes
    Order deny,allow
```

allow from all

deny from all

</Location>

Apêndice B. Licença de Publicação Livre

Esta é uma tradução não-oficial da Open Publication License versão 1.0, de 8 de junho de 1999, e não é substituto legal para a Licença original, disponível em <http://www.opencontent.org/openpub>. Entretanto, esta tradução poderá auxiliar pessoas que falem Português a entender melhor a licença. É permitido a qualquer pessoa copiar e distribuir cópias desse documento de licença, desde que sem a implementação de qualquer mudança.

OPEN PUBLIC LICENSE

Draft v1.0, 8 June 1999

I. Requisitos comuns às versões modificadas e não modificadas

Os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) podem ser reproduzidos e distribuídos no todo ou em parte, em qualquer meio físico ou eletrônico, desde que os termos desta licença estejam incluídos, e que esta licença ou uma incorporação dela por referência (com quaisquer das opções escolhidas pelo autor ou editor) estejam presentes na reprodução.

A forma apropriada para uma incorporação por referência deste livro é:

Copyright© 2002 Alfamídia Ltda. Este material somente poderá ser distribuído se sujeito aos termos e condições firmados na Licença de Livre Publicação (Open Publication License), versão 1.0 ou superior (a versão mais atual encontra-se disponível em <http://www.opencontent.org/openpub/>).

Esta referência, devidamente preenchida com os dados da publicação, deve ser seguida imediatamente com quaisquer opções escolhidas pelos autores ou editor do documento (consultar a seção *Termos opcionais*).

É permitida a redistribuição comercial de material licenciado pela Licença de Livre Publicação (Open Publication License).

Qualquer publicação no formato livro padrão (papel) requer obrigatoriamente a citação dos autores e editor originais. Os nomes dos autores e do editor devem aparecer em todas as superfícies externas do livro. Em todas as faces externas do livro, o nome do editor original deve estar impresso em tamanho tão grande quanto o título do trabalho, e citado como proprietário em relação àquele título.

II. Copyright

O *copyright* de todo trabalho protegido pela Licença de Livre Publicação (Open Publication License) pertence aos autores ou proprietários.

III. Escopo da licença

Os termos de licença a seguir aplicam-se a todos os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License), a não ser que explicitamente indicado no trabalho.

A mera adição de trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) ou partes de trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) em uma mesma mídia que contenha outros trabalhos ou programas não protegidos por essa licença não decorre em aplicação da Licença de Livre Publicação (Open Publication License) para esses outros trabalhos. O trabalho resultante deve explicitamente conter uma nota especificando a inclusão do material protegido pela Licença de Livre Publicação (Open Publication License) e o aviso de copyright apropriado.

APLICABILIDADE. Se alguma parte desta licença não puder ser aplicada em alguma jurisdição, as partes restantes deste documento continuam sendo aplicadas.

AUSÊNCIA DE GARANTIA. Os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) são fornecidos "como estão", sem garantias de qualquer tipo, explícita ou implícita, incluindo, mas não limitado a, as garantias implícitas de comercialização e conveniência para um propósito particular, ou garantia de não-infração.

IV. Requisitos para trabalhos modificados

Todas as versões modificadas de documentos cobertos por esta licença, incluindo traduções, antologias, compilações e documentação parcial, deve seguir os requisitos abaixo:

A versão modificada deve ser indicada como tal.

As pessoas que fizerem as modificações e as datas de modificação devem ser identificadas.

O reconhecimento dos autores e editor originais (se aplicável) deve ser mantido de acordo com as práticas acadêmicas usuais de citação.

O local da versão não-modificada do documento deve ser indicado.

Os nomes originais dos autores não devem ser utilizados para indicar ou garantir seu endosso ao documento resultante sem a autorização expressa dos autores.

V. Práticas recomendadas

Em adição aos requisitos desta licença, é solicitado e extremamente recomendado aos redistribuidores que:

Se os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) estiverem sendo distribuídos em impressos ou CD-ROM, os autores sejam informados por email, ao menos trinta dias antes, para que os autores tenham tempo de providenciar documentação atualizada. Esta notificação deve descrever as modificações introduzidas no documento, se existirem.

Todas as modificações substanciais (incluindo exclusões) devem ser marcadas claramente no documento, ou então descritas em um anexo ao documento.

Finalmente, mesmo não sendo obrigatório sob esta licença, é considerado de bom tom oferecer uma cópia sem ônus de todo o material modificado (impresso e CD-ROM) para os autores originais.

VI. Termos opcionais

Os autores e editores de documentos protegidos pela Licença de Livre Publicação (Open Publication License) podem escolher certas opções de licença simplesmente incluindo alguns parágrafos após a cópia da licença ou sua referência. Estas opções são consideradas parte da licença e devem ser incluídas com ela (ou com a referência a ela) nos trabalhos derivados.

As opções que se aplicam a este trabalho são:

A:É vedada a distribuição de versões com modificações substanciais deste documento sem a expressa permissão dos proprietários do direito autoral.

B:É vedada a distribuição deste trabalho ou qualquer derivado seu em qualquer formato de livro padrão (papel) sem a prévia autorização dos proprietários do direito autoral.

Políticas de Publicação Livre

(O texto a seguir não é considerado parte da licença.)

Os trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) estão disponíveis e podem ser acessados na home page da Open Publication <http://works.opencontent.org/>.

Os autores de trabalhos protegidos pela Licença de Livre Publicação (Open Publication License) podem incluir suas próprias licenças nesses trabalhos, desde que os termos dessa licença não sejam mais restritivos que os da Licença de Livre Publicação (Open Publication License).

Em caso de dúvidas sobre a Licença de Livre Publicação (Open Publication License), contactar David Wiley <dw2@opencontent.org> ou a lista de autores de publicações <opal@opencontent.org> via email.

Para se inscrever na lista de autores de publicações livres (Open Publication Author's List), mande um email para <opal-request@opencontent.org> com a palavra **subscribe** no corpo da mensagem.

Para enviar mensagens para a lista de autores de publicações livres (Open Publication Author's List), mande um email para **opal@opencontent.org** ou simplesmente responda a uma mensagem postada.

Para se desinscrever na lista de autores de publicações livres (Open Publication Author's List), mande um email para **opal-request@opencontent.org** com a palavra **unsubscribe** no corpo da mensagem.

Apêndice C. Códigos de retorno HTTP

Este apêndice pode ser uma interessante referência para a programação e configuração da diretiva *ErrorDocument*, etc.

- 2xx – Sucesso
 - o 200 OK
 - o 201 Criado
 - o 202 Aceito
 - o 203 Informação não–autoritativa *
 - o 204 Nenhum conteúdo
 - o 205 Conteúdo resetado *
 - o 206 Conteúdo parcial *
- 3xx – Redirecionamento
 - o 300 Múltiplas escolhas
 - o 301 Movido Permanentemente
 - o 302 Movido Temporariamente
 - o 303 Veja outra *

- o 304 Não modificada
- o 305 Use o Proxy (redirecionamento proxy) *
- 4xx – Erros no Cliente
- o 400 Requisição incorreta
- o 401 Não autorizado
- o 402 Pagamento Requerido *
- o 403 Bloqueado
- o 404 Não encontrada
- o 405 Método não permitido *
- o 406 Não aceitável *
- o 407 Autenticação via proxy requerida *
- o 408 Tempo limite da requisição expirado *
- o 409 Conflito *
- o 410 Gone *
- o 411 Tamanho requerido *
- o 412 Falha na pré-condição *

- o 413 A requisição parece ser grande *
- o 414 A URL requisitada é muito longa *
- o 415 Tipo de mídia não suportado
- 5xx – Erros no Servidor
- o 500 Erro Interno no Servidor
- o 501 Não implementado
- o 502 Gateway incorreto
- o 503 Serviço não disponível
- o 504 Tempo limite no gateway *
- o 505 Versão HTTP não suportada *

Os códigos de erros marcados com "*" são suportados apenas pelo padrão HTTP 1.1