

UNCLASSIFIED

# Guide to Securing Apache on Linux

Prepared by the  
Space and Naval Warfare Systems Center, San Diego  
for the  
Technical Support Working Group

**Authors:**  
Charles N. Long  
Carsten P. Gehrke



**Revision:**  
28 Oct 2004

Distribution Statement: This document is available for general release to all interested parties. The software associated with the Fort Knox for Linux (FKL) program along with this documentation is licensed under the terms of the GNU General Public License (GPL) and as such is considered open source. The software and this documentation is free; you can redistributed it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. You should have received a copy of the GNU General Public License along with the FKL program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

UNCLASSIFIED

UNCLASSIFIED

This page intentionally left blank.

UNCLASSIFIED

**Revision Information**

<b>Date</b>	<b>Comment</b>	<b>Author</b>
2004-10-28	First public release.	CNL

## Disclaimer

SOFTWARE AND DOCUMENTATION IS PROVIDED “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Trademark Information

Linux® is a registered trademark of Linus Torvalds.

“Red Hat” is a registered trademark of Red Hat, Inc.

“SUSE” is a registered trademark of SUSE AG, a Novell, Inc. Business.

Apache is a trademark of The Apache Software Foundation, and is used with permission.

All other trademarks are property of their respective owners.

## Abstract

This document is a guide to hardening an Apache Web server that is running on either Red Hat Enterprise Linux Server 3 or SUSE Linux Enterprise Server 9. While Apache itself is considered secure, errors made by an administrator may still cause the server to be compromised. This document addresses several issues which have been identified as potential security problems, and provides guidance in regard to mitigating their effects. The recommendations have been derived from many books, documents, and Web sites authored by security professionals and organizations that have been studied and analyzed to determine industry best practices. Furthermore, Department of Defense directives were consulted to determine the Department's specific requirements.

The objective of this document is to instill a better understanding of how to secure Apache. Among the topics covered are the threats and risks related to operating a Web server in general, as well as the installation and secure configuration of the Apache Web server software in particular. In addition, setup of `mod_security` and Secure Socket Layer (SSL) are discussed. The appendices include configuration file templates which can be used to set up an Apache server.

Being of a technical nature, this document is directed at system and Web server administrators as well as sophisticated users of computers running the Linux operating system. A degree of familiarity with Linux or a UNIX system is assumed. Some experience with Apache may be helpful. Knowledge of computer security is not required.

## Typographic Conventions

<b>Typeface</b>	<b>Meaning</b>
command	Command line input, the names of files, the contents of text files.
<i>variable</i>	A variable in a command line, to be replaced with an actual value such a real file name.

## Table of Contents

<b>Revision Information</b> .....	i
<b>Disclaimer</b> .....	ii
<b>Trademark Information</b> .....	iii
<b>Abstract</b> .....	iv
<b>Typographic Conventions</b> .....	v
<b>1 Introduction</b> .....	1
1.1 Services .....	1
1.2 Threats and Risks .....	1
1.2.1 Input Validation .....	1
1.2.2 Cross Site Scripting .....	2
1.2.3 Web Server Misconfiguration .....	2
1.2.4 Denial of Service attacks .....	2
1.2.5 Buffer Overflow Attacks .....	2
1.3 Verification of Integrity .....	2
<b>2 Securing Apache</b> .....	5
2.1 General Operating System Security .....	5
2.2 Permissions on Server Root Directories .....	5
2.3 CGI Scripts and Server Side Includes .....	6
2.3.1 Deleting Default CGI Scripts and Files .....	6
2.3.2 CGIwrap and suEXEC .....	6
2.4 Aliased Directory .....	6
<b>3 Configuration of Apache</b> .....	9
3.1 Basic Configuration .....	11
3.2 Options .....	12
3.3 Granting Access to Directories .....	13
3.4 htaccess and Password Authentication .....	13
3.5 Directory Index .....	13
3.6 Mime Types .....	13
3.7 Logging .....	13
3.8 ScriptAlias Directory .....	14
3.9 Information Revealed .....	14
3.10 Browser Identification .....	15
<b>4 Mod_Security</b> .....	17
<b>5 Secure Socket Layer</b> .....	19
5.1 Red Hat Enterprise Server .....	19
5.1.1 Creating a Private Key and Self-Signed Certificate .....	19
5.2 SUSE Enterprise Server .....	20
5.2.1 Creating a Private Key and Self-Signed Certificate .....	20
5.2.2 Yet Another Setup Tool (YaST) .....	21
<b>Appendix A</b> .....	23
<b>Appendix B</b> .....	27
<b>Appendix C</b> .....	31
<b>Appendix D</b> .....	35

<b>Appendix E</b> .....	37
License Information .....	37
<b>References</b> .....	43

UNCLASSIFIED

This page intentionally left blank.

# 1 Introduction

Securing the Apache httpd server requires the same thought process as for securing an Operating System (OS). The program must be regularly maintained and updated so that potential security holes are patched, and it must be properly configured. Apache should be installed on a hardened operating system which is configured to provide the least number of services as possible. The server should reveal the least amount of information possible to prevent supplying potential attackers with knowledge of possible vulnerabilities. The server should not run scripts as they are a weak area of security. If scripts are necessary, particular care should be taken to ensure that they are securely written. Three fundamental rules to remember about computer security are to keep your system up to date, keep it simple and keep it backed up. In this regard the Apache httpd server is no different.

This document will focus on the Apache 2 Web server though much of the information is also useful for Apache 1.3. Instructions regarding the securing of the underlying Linux OS can be found in the "Guide to Securing Linux", available at <http://fortknock.sourceforge.net/>. These guidelines were developed from the study of numerous documents, books, and other works on computer security. Where applicable, requirements of the U.S. Department of Defense (DoD) have been observed and implemented.

## 1.1 Services

A Web server increases its security if it only serves static Web pages. This limits the use of scripts written for a Common Gateway Interface (CGI). These interactive scripts open up greater areas of attack against the Web server. Dynamic Web pages are a better option as they do not make the server vulnerable to input validation attacks. Running minimal services on Apache limits overall exposure to the server. This means running minimal modules on Apache. In short, the fundamental aspect of security is to run only necessary services.

## 1.2 Threats and Risks

The major threats and risks against the Apache Web server are Input Validation, Cross Site Scripting (XSS), Web server misconfiguration, Denial of Service (DoS) attacks, and buffer overflows.

### 1.2.1 Input Validation

The lack of input validation is a threat to a server that can be fixed easily by not running interactive CGI scripts or Server Side Includes with CGI execution. A simple example of an input validation attack would be reverse directory traversal. If a Web form processes a request such as `script.cgi?file=database.txt` by returning the contents of the named file, and an attacker enters `script.cgi?file=../../../../etc/passwd`, the attacker now has the password file. The attacker can now go about trying to crack an account. One way to guard against this attack is to carefully validate the input, for instance by rejecting patterns such as `".."`. A better approach may be to reject any input that does not match a very restricted set of allowed patterns. A complete discussion of this topic is beyond the scope of this document, a good introduction is available at the following URL:

<http://www-106.ibm.com/developerworks/linux/library/l-sp2.html>

### 1.2.2 Cross Site Scripting

Cross Site Scripting (XSS) attacks are a server-side vulnerability, which can lead to a client computer unknowingly providing attackers with private or confidential data. These attacks are initiated when a client computer runs a cross-site script on a compromised server. These types of attacks involve three entities, a client's Web browser, the attacker's Web server, and the compromised Web site. The attacker uses the cross-site script on the compromised Web site with their server to gain access to the client's information. The attacker is then able to impersonate the client in future Web transactions. The risk involved here is loss of privacy and theft of private data, such as bank account number and credit card. An example of Cross Site Scripting can be found at this URL:

<http://crypto.stanford.edu/cs155/CSS.pdf>

### 1.2.3 Web Server Misconfiguration

The Apache server should be configured to avoid simple security mistakes. These simple mistakes include: installing default CGI scripts and files that are not required, specifying incorrect file permissions, allowing directory listings, allowing easy access to the logs, and running the Apache httpd as root. These types of mistakes can lead to defacement of Web sites, the installation of XSS or other malicious programs and theft of the server files, maybe even the password file. While following best practices will generally protect Web server administrators, care should be taken to avoid the simple mistakes.

### 1.2.4 Denial of Service attacks

Denial of Service (DoS) attacks try to make a service fail in carrying out its function by overloading the resources it uses. Examples of such resources are bandwidth, memory, and CPU cycles. DoS attacks or Distributed Denial of Service (DDoS) are very difficult to overcome without sufficient resources to overcome the burden that the attack is putting on the server. Remember that keeping a server simple and limiting the number of services restricts the effect these attacks have on these resources.

### 1.2.5 Buffer Overflow Attacks

While in the past there have been buffer overflow vulnerabilities in Apache, today it is actually considered a rather secure piece of software. Buffer overflow attacks from an administrative stand point are easy to secure against. The fundamental rule for stopping buffer overflow attacks is to keep your Web server and system as a whole patched.

## 1.3 Verification of Integrity

Since Linux is an open-source project, many distributions, patches, and additional software are available on public servers for download by anyone around the world. These servers are just as vulnerable to compromise as any other system. Therefore, verifying the integrity of the downloaded files is extremely important.

Perhaps the best method for performing such a test is through the use of Pretty Good Privacy (PGP) signatures. An open-source implementation of PGP is the GNU Project's Privacy Guard (GPG), which is included in most distributions of Linux. The use of GPG requires that

the downloaded file has been signed by the author or vendor's private key. You will need to obtain the corresponding public key. For some distributions, the key may be stored on the media. In other cases, it must be retrieved from another source. This can easily be done by using one of the many public key servers such as `keyserver.pgp.com`, `wwwkeys.pgp.net`, or `pgp.mit.edu`, e.g.:

```
/usr/bin/gpg --keyserver pgp.mit.edu --recv-keys key-id
```

Where *key-id* is the hexadecimal identification number of the public key, such as `0xE0002FC4`. You may also find keys on Web sites, as e-mail signatures, and a variety of other places. To test the signature of a file, issue the following command:

```
/usr/bin/gpg --verify file.sig signed.file.name
```

In the above example *file.sig* is the detached signature file, usually ending in `.sig` or `.asc`, and *signed.file.name* is the downloaded file that is to be checked for integrity. For RPM files, this command should be used:

```
/bin/rpm --checksig -v file.name
```

These are only a few examples of using PGP signatures; refer to `man gpg` or the many books and Web sites devoted to this topic for more information.

If a PGP signature is not available for the file in question, try to locate an MD5 fingerprint of the file that is cryptographically signed or from a different source. Do not use an unsigned MD5 fingerprint from the same server from which you obtained the file itself; if the file was replaced by a maliciously modified version, the attacker most likely replaced the MD5 fingerprint as well (this occurred in the fall of 2002 for the source code package of the `tcpdump` program, available at <http://www.tcpdump.org/>). Other sources may be a mirror site (provided it hasn't automatically mirrored the altered file and fingerprint) or an e-mail from the author or vendor directly.

UNCLASSIFIED

This page intentionally left blank.

## 2 Securing Apache

### 2.1 General Operating System Security

The Apache Web server must be run on a host operating system that has been hardened. This hardening process requires the computer to provide only the minimal set of functionality needed to perform its defined tasks. For Web servers, the minimal set of functionality is ssh, ntp and the Apache httpd server. Ssh functionality provides encrypted communication mechanisms so that users and administrators can interface with the server remotely, while ntp is used to keep and update time correctly. Users can remotely upload Web content by using scp and sftp, since these use encrypted protocols and thus make it much more difficult to discover passwords by analyzing network traffic. DoD Instruction 8500.2 requires that passwords not be transmitted or stored in clear text. This prohibits the use of protocols which do not encrypt such sensitive information, e.g. FTP. The “Guide to Securing Linux” can provide instructions on how to harden your system. Additionally, the Bastille hardening tool will automate many of the steps needed to secure your system.

A DeMilitarized Zone (DMZ) should be implemented, which is a small subnetwork that sits between a trusted internal network and an untrusted external network such as the Internet. The Web server should be hosted inside a DMZ or outside the firewall. Never run a publicly available service behind the firewall on an internal network.

### 2.2 Permissions on Server Root Directories

The default install of Apache in Red Hat Package Manager (RPM) format provides the correct file permissions. You will find the files pertaining to the Apache Web server by typing in the following command.

**For Red Hat Enterprise Server 3:**

```
/bin/rpm -qc httpd
```

**For SuSE Enterprise Server 9:**

```
/bin/rpm -qc apache2
```

The following set of commands is recommended by the Apache team when installing Apache from source. This information can be found at the following URL:

[http://httpd.apache.org/docs-2.0/misc/security\\_tips.html](http://httpd.apache.org/docs-2.0/misc/security_tips.html).

```
mkdir /usr/local/apache
cd /usr/local/apache
chown 0 . bin conf logs
chgrp 0 . bin conf logs
chmod 755 . bin conf logs
```

Except for the document root directory all Web directories should only be modifiable by root. This will prevent overwriting of log files and placing files that may accidentally be run by Apache. The reason the document directories can be modified by others is that Apache will not access them when it is starting up as root. Apache needs to start as root because it needs to bind to port 80. Only the root user has access to the ports numbered less than 1024. Following

startup the Web server changes its privilege level to the user specified in the `httpd.conf` file after it has bound to port 80. Apache reduces its security threat when executed as a normal nonprivileged user. This is an important reason to refrain from running Apache as root. Another step that may be taken to secure the Web site from defacement is to not give the Apache user write permissions on the document directories. This stops a compromised Web server from changing the contents of the document directory, and therefore the Web site content.

## 2.3 CGI Scripts and Server Side Includes

Security is strengthened by offering less functionality by the Web server. Do not offer Server Side Includes (SSI) nor CGI scripts if security is the number one priority. This can lead to exposure of the Web server to various attacks. These types of attacks allow the attackers to send arbitrary information into the CGI script with the intention that the CGI processor will execute their rogue commands or scripts. Another reason to refrain from running SSI is that Apache must parse every SSI-enabled file, which can lead to a large load increase on the server. If SSI must be used, do not allow it to execute CGI scripts with the `NOEXEC` option. If SSI with the `EXEC` option is used then at the very least limit the CGI to specific directories with `<!--#include virtual="/cgi-bin/blink.shtml"-->`. This will force the SSI to retrieve the file from a known directory. These, and other Apache configuration options, will be discussed in more detail later in this document.

### 2.3.1 Deleting Default CGI Scripts and Files

When running CGI scripts, make certain that the default scripts included in the distributions are not in the `cgi-bin` directory when the server goes online. In Apache, these are `printenv.pl` and `test-cgi.pl`. These scripts will basically give away what version and type of Web server it is. This will show too much information about other exploits that may be used against the server. Delete `<DocumentRoot>/icons` and `<DocumentRoot>/manual`, and any other files that do not need to be on the server. Changing the default error pages to help obscure the Apache server is also recommended.

### 2.3.2 CGIwrap and suEXEC

Another aspect of CGI scripts is that they are executed as a user defined in the `httpd.conf` file, which has equal rights with all users. As such, anyone can write a malicious script which can ruin someone else's files on the same server. There are two programs that can be used to stop this, but they need to be compiled from source. They are `cgi-wrap`, available at this URL: <http://cgiwrap.unixtools.org/>

The other is `suEXEC`, which can be found at: <http://httpd.apache.org/docs/suexec.html>

The latter is part of the SUSE Linux Enterprise Server distribution, and thus does not need to be compiled for that platform.

## 2.4 Aliased Directory

CGI scripts should only be run from the special designated aliased directory. This allows the administrator to have more control over the CGI scripts. Also, this allows the administrator

## UNCLASSIFIED

to monitor the code for blatant security violations. The alternative is to allow CGI executables anywhere on the server which results in difficulty for the administrator to determine what scripts are on his server.

UNCLASSIFIED

This page intentionally left blank.

### 3 Configuration of Apache

This section will guide you through a simple Apache `httpd.conf` file. In the interest of clarity and security, the `httpd.conf` will be simple and to the point. A simple and concise `httpd.conf` file results in a much more secure server as everything is understood, and nothing is turned on unintentionally. Refer to the Appendices A and B for sample `httpd.conf` files. This server will only run the essential modules to limit the number of security vulnerabilities. We are using modules for ease of use, compatibility, and upgradability; however, another option is to compile the source with all the functionality needed, and not allow modules to be loaded at all. While this will be more secure, it is much harder to maintain.

- **ServerRoot:** The top of the directory tree under which the server's configuration, error and log files are kept.

```
ServerRoot "/etc/httpd"
```

- **Timeout:** The number of seconds before Apache sends a time out.

```
Timeout 300
```

- **KeepAlive:** Whether or not to allow persistent connections (more than one request per connection). Set to "off" to deactivate. If DoS attacks are a problem set `KeepAlive` to "off"; however, if performance is an issue set it to "on".

```
KeepAlive on
```

- **MaxKeepAliveRequests:** The maximum number of requests to allow during a persistent connection. Set to 0 to allow an unlimited amount. We recommend leaving this number high, for maximum performance.

```
MaxKeepAliveRequests 100
```

- **KeepAliveTimeout:** Number of seconds to wait for the next request from the same client on the same connection.

```
KeepAliveTimeout 15
```

- **Listen** tells Apache which port and IP address to bind to.

```
Listen 80
```

#### Rationale:

This will make Apache listen to requests on all addresses at port 80. If the server is used for development purposes then binding the IP address to `127.0.0.1:80` will provide additional security since it will not listen to any of the external addresses. If the Web server is for an internal network than it might make sense to bind the Apache daemon to a local address such as `Listen 192.168.1.0/24:80`.

- **File** where the server records the process ID of the daemon.

```
PidFile /var/run/apache.pid
```

- The LockFile directive sets the path to the lockfile. Do not put this file in a world writable file system as someone could create a DoS attack by creating a file with the same name and the server would not start.

```
LockFile /var/run/apache.lock
```

- Description of Apache modules

Uncomment the minimum loadable modules needed for a working Apache Web server with the `httpd.conf`. If additional functionality is needed then the modules that are needed should be uncommented. To find the modules that are compiled in you would type the following command on the `httpd` binary. For a full description of all modules please look at Appendix D. This guide will assume that the Apache server will be a Prefork server vs. a MPM server which is threaded. If SLES 9.0 and greater is being used then the `httpd` binary is called `httpd2`.

```
httpd -l
```

```
core.c
prefork.c
http_core.c
mod_so.c
```

- User/Group to run the Apache Web server as.

#### **For Red Hat Enterprise Server 3:**

```
User apache
Group apache
```

#### **For SuSE Enterprise Server 9:**

```
User wwwrun
Group www
```

- ServerAdmin: The e-mail address that should be used to send mail. Note do not use an e-mail of a user name on the system as this will be giving someone an account name they can try and hack.

```
ServerAdmin webmaster@localhost
```

- ServerName This tells the server how to identify itself. If the host does not have a registered DNS name, enter its IP address here. The server will have to be accessed by its address anyway, and this will make redirection work in a sensible way.

```
ServerName new.host.name
```

- UseCanonicalName Determines how Apache constructs self-referencing URLs and the `SERVER_NAME` and `SERVER_PORT` variables. When set "Off", Apache will use the Hostname and Port supplied by the client. When set "On" Apache will use the value of the `ServerName` directive. If password access to this Web site is needed leave canonical name on because some browsers cache authentication information by URL.

```
UseCanonicalName Off
```

## UNCLASSIFIED

- **DocumentRoot:** This is the default directory where the documents are served from.

```
DocumentRoot "var/www/html"
```

- **HostnameLookups:** Log the names of clients or just their IP addresses.

```
HostnameLookups Off
```

- **Errorlog:** The location of the error log file. If the `ErrorLog` directive is not setup within a `VirtualHost` container, error messages relating to that virtual hosts will be logged here. If an error log file for a `VirtualHost` container is defined then that host's errors will be logged there and not here.

```
ErrorLog logs/error_log
```

- **LogLevel:** Controls the number of messages logged to the error log. The possible values include: debug, info, notice, warn, error, crit, alert, and emerg.

```
LogLevel warn
```

### 3.1 Basic Configuration

One way to set up a Web server is to start from a default deny stance in the top directory, and after that open up access on a need only basis. This configuration style makes certain that a mistake in the configuration will not provide access to some areas on the Web document tree that were not intended to be served. The `Directory "/"` field tells Apache the area of the UNIX file system that the specific options within the container apply to. `Options` field tells Apache what options to use for that particular container. If options are not wanted, do not put it in the file as options are disabled by default. If options are turned on, you would make a list such as `Options Indexes FollowSymlinks ExecCGI`. There are other ways to do it, but this is the simplest. The following is an example of a standard Apache setup. It has no options and includes everything under the root directory.

<code>&lt;Directory "/"&gt;</code>	Marks the beginning of the block and specifies which directory the block applies to. In this case, it is the host's root directory.
<code>Options None</code>	This could be All, None, Indexes, FollowSymlinks/SymlinksIfOwnerMatch, Includes/IncludesNOEXEC, ExecCGI, MultiViews.
<code>Order allow,deny</code>	This means to do the allow statements, then the deny rules. The default is deny because it is second.
<code>Deny from all</code>	This will explicitly deny from all.
<code>Satisfy all</code>	Both access controls must exist and be valid.
<code>AllowOverride None</code>	.htaccess files are ignored as Apache doesn't allow a user to deviate from this configuration.
<code>&lt;/Directory&gt;</code>	Marks the end of the block.

## 3.2 Options

- The `Indexes` option tells Apache to show the contents of the current directory if it can not find an `index.html` file. Do not turn this on as it will show the contents of the directories with no `index.html` file, this will give away unnecessary information.
- The `FollowSymLinks` option tells the Web server to follow the symlinks even out of the normal directory tree. This is not good from a security standpoint because a remote user could get access to files that they should not. An example from the Apache Web site illustrates this. If the administrator issues the following commands:

```
cd /
ln -s / public_html
```

Then a remote user could request the URL [http://host\\_name/~root/](http://host_name/~root/) and the server would show the contents of the root user's home directory.

- Leave `FollowSymLinks` on for performance reasons. Since Apache needs to make sure that the file is not a symlink, it must look at every file to make sure it is not a symlink, which will degrade performance.
- The `SymlinksIfOwnerMatch` will only follow the link if the link owner and the file owner are the same. This is twice as much a performance hit as following symlinks, but gives a higher level of security if symlinks must be followed.
- The `IncludesNOEXEC` option allows the server to parse dynamic Web pages. As stated earlier do not turn this on from a security standpoint, but sometimes clients require certain services. This option will allow Apache to serve and parse `.shtml` files. The option `ExecCGI` will turn on CGI access. Do not enable this option as it will allow Server Side Includes to execute CGI scripts from anywhere on your server. CGI scripts should only be in the aliased CGI directory.
- The option `MultiViews` enables a Web server to give out the same document in different languages. Multiviews must be turned on implicitly to work.
- The field `Order allow,deny` really means to deny all as Apache always uses the second in the `Order` directive. Always remember that the second one in the list of the `Order` directive is the one that effects the Web server policy.
- The field `Deny from all` is the same as `Order allow,deny`.
- The field `Satisfy all` defines how any inconsistencies in access control will work out. `Satisfy all` requires the Web user to have the right password and come from the correct IP; however, if it is `satisfy any` then the user will need to be either from the right IP address, or have the correct password to login.
- The field `AllowOverride None` disables the Apache looking at `.htaccess` files.

### 3.3 Granting Access to Directories

This is how to give implicit access to `public_html` folders in users' home directories.

```
<Directory /home/*/public_html>
  Order Deny,Allow
  Allow from all
</Directory>

<Directory /usr/local/apache/htdocs>
  Order Deny,Allow
  Allow from all
</Directory>
```

### 3.4 htaccess and Password Authentication

This keeps `.htaccess` files from being served up by the Web server. People should not be able to look at these files as they have private information.

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

Password authentication requires a few small changes to the `httpd.conf` file, but it is important to remember that if basic authentication mode is used then the passwords are being sent unencrypted. If authentication is used then it should be sent over an SSL link. A good example on how to setup password protection may be found at <http://httpd.apache.org/docs-2.0/howto/auth.html>

### 3.5 Directory Index

Directory Index tells Apache what extension is the default for the index Web page.

```
DirectoryIndex index.html
```

### 3.6 Mime Types

Mime types control what Internet media types are sent to the client for any given file extension(s). Sending the correct media type to the client is important so they know how to handle the content of the file.

Mime types information:

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

### 3.7 Logging

It is important to have proper file permissions on the log directory. Someone with write access to the log directory may be able to gain access to the UID that the server is started as, which is usually root. No one should have access to the log files except root.

## UNCLASSIFIED

HostnameLookups off	This makes Apache resolve IP addresses to names.
ErrorLog logs/error_log	Apache diagnostic information.
LogLevel warn	The level Apache is logging at.

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

These LogFormat variables mean:

%h	The record of the IP address of the client.
%l	The identity of the client determined by identd on the client machine.
%u	The userid of the person asking for the document.
%t	The time stamp.
%r	The request line from the client. This tells how the client asked for the resource, what that resource was, and what protocol was used.
%>s	This is the status code that the server sends back to the client.
%b	This is the size of the object returned to the client.

### 3.8 ScriptAlias Directory

CGI scripts should be turned off and at the very least they need to be in an aliased script directory such as `/var/www/cgi-bin`.

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
<Directory "/var/www/cgi-bin">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
</Directory>
```

### 3.9 Information Revealed

`ServerTokens Prod` is the lowest setting for `ServerTokens` which will respond to a query with Apache. The only way to get Apache to not respond with Apache is to edit the source code.

```
ServerTokens Prod
ServerSignature off
```

`ServerSignature off` tells Apache to not put out a footer message with its version number.

#### Rationale

This should always be off as it gives away too much information.

Default charset should be set.

```
AddDefaultCharset UTF-8
```

#### Rationale

This helps prevent cross site scripting attacks.

### 3.10 Browser Identification

To accommodate the different behavior of the various user agents that might send requests to the Apache Web server, the configuration file allows environment variables to be set depending on the identification string sent by the agent. The following list is the default and should not be shortened to maintain compatibility.

```
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade1.0 forceresponse1.0
BrowserMatch "RealPlayer 4\.0" forceresponse1.0
BrowserMatch "Java/1\.0" forceresponse1.0
BrowserMatch "JDK/1\.0" forceresponse1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirectcarefully
BrowserMatch "^WebDrive" redirectcarefully
BrowserMatch "^WebDAVFS/1.[012]" redirectcarefully
BrowserMatch "^gnomevfs" redirectcarefully
```

UNCLASSIFIED

This page intentionally left blank.

## 4 Mod\_Security

Mod\_security adds another layer to security that if used correctly may be very useful. Author Ivan Ristic describes mod\_security as “an open source intrusion detection and prevention engine for applications”. It operates embedded within the server, acting as a powerful umbrella – shielding applications from attacks. The best place to learn about mod\_security is at the URL: <http://modsecurity.org/>

This module is useful, but it is not mandatory as you will need to have a compiler and other tools on your computer to install it.

Mod\_security will be compiled as a Dynamic Shared Object (DSO), or module with Red Hat or SUSE. To do this the `httpd-devel-2.0.46-38.ent.i386.rpm` package must be installed, and to have mod\_security compile you also need the following installed. If SUSE is being used then the best way to install these programs is through YaST.

```
glibc-kernheaders-2.4-8.34.i386.rpm
glibc-headers-2.3.2-95.3.i386.rpm
glibc-devel-2.3.2-95.3.i386.rpm
cpp-3.2.3-20.i386.rpm
gcc-3.2.3-20.i386.rpm
automake-1.6.3-5.noarch.rpm
autoconf-2.57-3.noarch.rpm
libtool-1.4.3-6.i386.rpm
```

The `httpd-develop-2.0.46-38.ent.i386.rpm` contains the APXS binary and other files, which are needed to build mod\_security. Download the latest mod\_security from this URL:

<http://modsecurity.org/download/index.html>

The following commands will compile the mod\_security module:

```
tar xvzf mod_security-1.8.4.tar.gz
cd mod_security-1.8.4
cd mod_security-1.8.4/apache2/
/usr/sbin/apxs -cia mod_security.c
```

If everything worked on Red Hat the following line should be displayed after the compile and install are finished.

```
activating module 'security' in /etc/httpd/conf/httpd.conf
```

Check your `httpd.conf` file to make sure that when you compiled mod\_security as a module it added this line:

```
LoadModule security_module /usr/lib/httpd/modules/mod_security.so
```

If this is a SUSE installation then this line must be added by hand, and an error will come up about not finding the `prefork httpd.conf` file, which can be safely ignored.

Included in Appendix C is a default `httpd.conf` file from the mod\_security website that may be appended to your `httpd.conf` file. One area to note about mod\_security is the chroot support. Many individuals prefer to chroot their Web server, so that it is in a jail. This means that if someone compromises a server, he will not have access to any tools or the host via exploitation of the server. Chrooting Apache server takes considerable effort, so Ivan has incorporated a way for mod\_security to do it with one line in the `httpd.conf` file. Two things to keep in mind when using this chroot function is that it is considered experimental, and if external files are needed for CGI scripts or system binaries, they need to be put in the chroot directory as well.

## UNCLASSIFIED

Using `mod_security` in this way, while experimental, will eliminate the work needed to chroot the `httpd` binary yourself. You must make sure that all of the paths are relative to Apache in jail.

```
mkdir -p /chroot/apache  
SecChrootDir /chroot/apache
```

## 5 Secure Socket Layer

Secure Socket Layer (SSL) is a complex topic that goes beyond the scope of this document in terms of hardening a server. This will be a quick overview of setting up SSL with a self-signed certificate to provide an encrypted connection through which users can perform transactions. DoD Instruction 8520.2 requires that Web sites that provide sensitive information use some form of authentication and employ encryption. This document will not cover setting up a server for more complex applications such as e-commerce.

### 5.1 Red Hat Enterprise Server

You will need to have the following rpms installed.

```
mod_ssl-2.0.46-25.ent.i386.rpm
openssl-0.9.7a-22.1.i386.rpm
```

After `mod_ssl-2.0.46-25.ent.i386.rpm` is installed the Web server will actually be able to support SSL transactions. All you have to do is point your Web browser to your host's IP address at port 443, so if the Web server's IP is 192.168.0.5 the URL would be `https://192.168.0.5`. The Web browser will give you a few warnings because the certificate is not signed by a trusted authority. This is expected behavior, and nothing to be alarmed about.

#### 5.1.1 Creating a Private Key and Self-Signed Certificate

The next step is to create a local private/public key for certificate requests. This command generates a secure private key in the execution directory. With this command, a passphrase is needed. This passphrase will be required when the server is started manually, but not by a reboot. In the command below *servername* can be whatever you want it to be. The best practice for certificate names is to name the key the same name as that of the domain it is used for, so if your domain name was `example.com` your key name should be `example.com.key` or `example.key`. If you leave off the `-des3` switch on the `openssl` command the private key will not have triple des encryption, and you will not be required to put in a passphrase

With passphrase

```
openssl genrsa -des3 -out servername.key 1024
```

Without passphrase

```
openssl genrsa -out servername.key 1024
```

The next step is to `chmod` your private key, and it should also be owned by root.

```
chmod 400 servername.key
chown root:root servername.key
```

The last step is to copy your private key to `/etc/httpd/conf/ssl.key`.

```
cp servername.key /etc/httpd/conf/ssl.key
```

To create a self-signed certificate a private key is needed, and the name of the server. The following command is used to create a certificate, which is followed by its output that the administrator needs to enter.

```
openssl req -new -key servername.key -x509 -out servername.crt
```

## UNCLASSIFIED

```
Country Name (2 letter code) [GB]:GB
State or Province Name (full name) [Berkshire]:
Locality Name (eg, city) [Newbury]:
Organization Name (eg, company) [My Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
```

After all of these steps are completed the self-signed certificate should be put in `/etc/httpd/conf/ssl.crt/`.

The Red Hat distribution inserts the following line in the `httpd.conf` so that it will read all of the configuration files in the `/etc/httpd/conf.d` directory.

```
Include conf.d/*.conf
```

The file `/etc/httpd/conf.d/ssl.conf` may be used, and if so then the following lines need to be changed for the self-signed certificate.

```
SSLCertificateFile /etc/httpd/conf/ssl.crt/servername.crt
SSLCertificateKeyFile /etc/httpd/conf/ssl.key/servername.key
```

Another simple way to setup SSL is by adding the relevant statements directly into the `httpd.conf` file. The following is an example of this approach.

```
Listen 443
<VirtualHost default:443>
DocumentRoot /srv/www/htdocs
ServerName servername
SSLEnging on
SSLCertificateFile /etc/apache2/ssl.crt/servername.crt
SSLCertificateKeyFile /etc/apache2/ssl.key/servername.key
SetEnvIf User-Agent ".*MSIT.*" nokeepalive ssl-unclean-shutdown
</VirtualHost>
```

## 5.2 SUSE Enterprise Server

These will be a quick start for SUSE Enterprise SSL. Red Hat and SUSE are different enough to warrant this. If you've installed Apache2 then `mod_ssl` is already included with that package. The easiest way to install Apache2 is through the YaST interface. Type in the command `yast`, and go to the YaST control center. Now follow the menus to software and then install and remove software. Tab to the [Search] button, which asks you the name of the file to type in under search phrase. At the Search phrase prompt type in Apache2. In the next menu Apache2 must be highlighted and have a + sign so that Apache will be installed after you tab to the accept button. If YaST shows an i next to Apache2 then it is already installed, if not, install it.

### 5.2.1 Creating a Private Key and Self-Signed Certificate

The next step is create a local private/public key to create certificate requests. This command generates a secure private key in the execution directory. With this command, a passphrase is needed. This passphrase will be required when the server is started manually, but not by a reboot. In the command below `servername` can be whatever you want it to be. The best practice for certificate names is to name it the same name as that of the domain it is used for, so if your domain name was `example.com` your key name should be `example.com.key` or maybe

example.key. If you leave off the `-des3` switch on the `openssl` command the private key will not have triple des encryption, and you will not be required to put in a passphrase.

With passphrase

```
openssl genrsa -des3 -out servername.key 1024
```

Without passphrase

```
openssl genrsa -out servername.key 1024
```

The next step is to `chmod` your private key, and it should also be owned by root.

```
chmod 400 servername.key
chown root:root servrname.key
```

The last step is to copy your private key to `/etc/httpd/conf/ssl.key`.

```
cp servername.key /etc/apache2/ssl.key/
```

To create a self-signed certificate a private key is needed, and the name of the server. The following command is used to create a certificate, which is followed by its output that the administrator needs to enter.

```
openssl req -new -key servername.key -x509 -out servername.crt
```

```
Country Name (2 letter code) [GB]:GB
State or Province Name (full name) [Berkshire]:
Locality Name (eg, city) [Newbury]:
Organization Name (eg, company) [My Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
```

After all of these steps are completed the self-signed certificate should be put in `/etc/apache2/ssl.crt/`.

The final step to incorporating a Secure Socket Layer into your server is to add the following lines to the `httpd.conf` file.

```
Listen 443
<VirtualHost default:443>
DocumentRoot /srv/www/htdocs
ServerName servername
SSLEngine on
SSLCertificateFile /etc/apache2/ssl.crt/servername.crt
SSLCertificateKeyFile /etc/apache2/ssl.key/servername.key
SetEnvIf User-Agent ".*MSIT.*" nokeepalive ssl-unclean-shutdown
</VirtualHost>
```

After this the server needs to be restarted. If a passphrase was used then that passphrase will be required on startup of the server; however, if the computer is restarted then a passphrase will not be required.

### 5.2.2 Yet Another Setup Tool (YaST)

YaST is available on SUSE distributions only, so this will not pertain to Red Hat installations. A lot of the configurations mentioned previously may be done with YaST. The moment you deviate from YaST by manually editing a file it is no longer useable. The following will show you what you can accomplish with YaST.

To access the http server from YaST type:

yast - network services - http server

From the http server configuration screen the httpd daemon may be enabled or disabled. The firewall rule sets can be set up so that they allow incoming traffic. The ports that Apache is listening on can be changed from 80 to whatever may be appropriate, and you are allowed to turn on 443 for SSL. Modules and httpd.conf options may be disabled. YaST even allows you to enable virtual hosts, and create a fake certificate authority from “snake oil” to enable SSL.

## Appendix A

Sample configuration file for Apache running on Red Hat Enterprise Linux.

```

ServerRoot "/etc/httpd"
Timeout 300
KeepAlive on
MaxKeepAliveRequests 100
KeepAliveTimeout 15
Listen 80
PidFile run/httpd.pid
LockFile run/apache.lock

#Load config files from the config directory "/etc/httpd/conf.d".
#uncomment if you are going to use SSL or setup your own.
Include conf.d/*.conf

#Prefork Server
StartServers      8
MinSpareServers  5
MaxSpareServers  20
MaxClients       150
MaxRequestsPerChild 1000

# Apache Modules
LoadModule access_module modules/mod_access.so
#LoadModule auth_module modules/mod_auth.so
#LoadModule auth_anon_module modules/mod_auth_anon.so
#LoadModule auth_dbm_module modules/mod_auth_dbm.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule include_module modules/mod_include.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule env_module modules/mod_env.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
#LoadModule expires_module modules/mod_expires.so
#LoadModule deflate_module modules/mod_deflate.so
#LoadModule headers_module modules/mod_headers.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule mime_module modules/mod_mime.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule status_module modules/mod_status.so
#LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule asis_module modules/mod_asis.so
#LoadModule info_module modules/mod_info.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
#LoadModule imap_module modules/mod_imap.so
#LoadModule actions_module modules/mod_actions.so
#LoadModule speling_module modules/mod_speling.so
#LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so

```

## UNCLASSIFIED

```
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule cache_module modules/mod_cache.so
#LoadModule suexec_module modules/mod_suexec.so
#LoadModule disk_cache_module modules/mod_disk_cache.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule mem_cache_module modules/mod_mem_cache.so
#LoadModule cgi_module modules/mod_cgi.so

# Change the server's owner
User apache
Group apache

# Server info
ServerAdmin root@localhost
ServerName new.host.name:80
UseCanonicalName Off
DocumentRoot "/var/www/html"

# Minimal permissions for any directory
<Directory />
  Options None
  AllowOverride None
  Order deny,allow
  Deny from all
</Directory>

# More permissive options
<Directory "/var/www/html">
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>

# Directory Index
DirectoryIndex index.html

# Security filters, saves .htaccess files
<Files ~ "^\.ht">
  Order allow,deny
  Deny from all
</Files>

# Mime types information
TypesConfig /etc/mime.types
DefaultType text/plain

# Logging
HostnameLookups Off
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b" common

# info given out
ServerTokens Prod
ServerSignature off

# CGI scripts turned off
# CGI directory exists, if you have that configured.
#ScriptAlias /cgibin/ "/var/www/cgibin/"
```

## UNCLASSIFIED

```
#<Directory "/var/www/cgi-bin">
# AllowOverride None
# Options None
# Order allow,deny
# Allow from all
#</Directory>

# Default charset, prevents XSS
AddDefaultCharset UTF-8

# For Compatability
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade1.0 forceresponse1.0
BrowserMatch "RealPlayer 4\.0" forceresponse1.0
BrowserMatch "Java/1\.0" forceresponse1.0
BrowserMatch "JDK/1\.0" forceresponse1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirectcarefully
BrowserMatch "^WebDrive" redirectcarefully
BrowserMatch "^WebDAVFS/1.[012]" redirectcarefully
BrowserMatch "^gnomevfs" redirectcarefully
```

UNCLASSIFIED

This page intentionally left blank.

## Appendix B

Sample configuration file for Apache when running on SUSE Linux Enterprise Server.

```

ServerRoot "/etc/httpd"

#Server's options
Timeout 300
KeepAlive on
MaxKeepAliveRequests 100
KeepAliveTimeout 15
PidFile /var/run/httpd.pid
LockFile /var/run/apache.lock

Listen 80

#This is a prefork server
LoadModule access_module /usr/lib/apache2-prefork/mod_access.so
#LoadModule auth_module /usr/lib/apache2-prefork/mod_auth.so
#LoadModule auth_anon_module
/usr/lib/apache2-prefork/usr/lib/apache2/g/mod_auth_anon.so
#LoadModule auth_dbm_module /usr/lib/apache2-prefork/mod_auth_dbm.so
#LoadModule auth_digest_module /usr/lib/apache2-prefork/mod_auth_digest.so
#LoadModule include_module /usr/lib/apache2-prefork/mod_include.so
LoadModule log_config_module /usr/lib/apache2-prefork/mod_log_config.so
#LoadModule env_module /usr/lib/apache2-prefork/mod_env.so
#LoadModule mime_magic_module /usr/lib/apache2-prefork/mod_mime_magic.so
#LoadModule cern_meta_module /usr/lib/apache2-prefork/mod_cern_meta.so
#LoadModule expires_module /usr/lib/apache2-prefork/mod_expires.so
#LoadModule deflate_module /usr/lib/apache2-prefork/mod_deflate.so
#LoadModule headers_module /usr/lib/apache2-prefork/mod_headers.so
#LoadModule usertrack_module /usr/lib/apache2-prefork/mod_usertrack.so
#LoadModule unique_id_module /usr/lib/apache2-prefork/mod_unique_id.so
LoadModule setenvif_module /usr/lib/apache2-prefork/mod_setenvif.so
LoadModule mime_module /usr/lib/apache2-prefork/mod_mime.so
#LoadModule dav_module /usr/lib/apache2-prefork/mod_dav.so
#LoadModule status_module /usr/lib/apache2-prefork/mod_status.so
#LoadModule autoindex_module /usr/lib/apache2-prefork/mod_autoindex.so
#LoadModule asis_module /usr/lib/apache2-prefork/mod_asis.so
#LoadModule info_module /usr/lib/apache2-prefork/mod_info.so
#LoadModule dav_fs_module /usr/lib/apache2-prefork/mod_dav_fs.so
#LoadModule vhost_alias_module /usr/lib/apache2-prefork/mod_vhost_alias.so
#LoadModule negotiation_module /usr/lib/apache2-prefork/mod_negotiation.so
LoadModule dir_module /usr/lib/apache2-prefork/mod_dir.so
#LoadModule imap_module /usr/lib/apache2-prefork/mod_imap.so
#LoadModule actions_module /usr/lib/apache2-prefork/mod_actions.so
#LoadModule speling_module /usr/lib/apache2-prefork/mod_speling.so
#LoadModule userdir_module /usr/lib/apache2-prefork/mod_userdir.so
LoadModule alias_module /usr/lib/apache2-prefork/mod_alias.so
LoadModule rewrite_module /usr/lib/apache2-prefork/mod_rewrite.so
#LoadModule proxy_module /usr/lib/apache2-prefork/mod_proxy.so
#LoadModule proxy_ftp_module /usr/lib/apache2-prefork/mod_proxy_ftp.so
#LoadModule proxy_http_module /usr/lib/apache2-prefork/mod_proxy_http.so
#LoadModule proxy_connect_module /usr/lib/apache2-prefork/mod_proxy_connect.so
#LoadModule cache_module /usr/lib/apache2-prefork/mod_cache.so
#LoadModule suexec_module /usr/lib/apache2-prefork/mod_suexec.so
#LoadModule disk_cache_module /usr/lib/apache2-prefork/mod_disk_cache.so
#LoadModule file_cache_module /usr/lib/apache2-prefork/mod_file_cache.so
#LoadModule mem_cache_module /usr/lib/apache2-prefork/mod_mem_cache.so

```

## UNCLASSIFIED

```
#LoadModule cgi_module /usr/lib/apache2-prefork/mod_cgi.so
#LoadModule security_module /usr/lib/apache2-prefork/mod_security.so

# Change the server's owner
User wwwrun
Group www

# Server info
ServerAdmin root@localhost
ServerName new.host.name:80
UseCanonicalName Off

DocumentRoot "/srv/www/htdocs/"

#Minimal permissions for any directory
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

# More permissive options for sub-directories

<Directory "/srv/www/htdocs">
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

DirectoryIndex index.html

#Security filters, saves .htaccess files
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

# Mime types information
TypesConfig /etc/mime.types
DefaultType text/plain

# Logging
HostnameLookups Off
ErrorLog /var/log/apache2/error_log

LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog /var/log/apache2/accesse_log common

#Info given out. It can be Full,OS,Minor,Minimal,Major,Prod
ServerTokens Prod
ServerSignature off

# CGI SCRIPTS
ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
<Directory "/srv/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
```

## UNCLASSIFIED

```
    Allow from all  
</Directory>
```

```
# Set the Default charset, prevents XSS  
AddDefaultCharset UTF-8
```

```
#Important hacks
```

```
BrowserMatch "Mozilla/2" nokeepalive  
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0  
BrowserMatch "RealPlayer 4\.0" force-response-1.0  
BrowserMatch "Java/1\.0" force-response-1.0  
BrowserMatch "JDK/1\.0" force-response-1.0  
BrowserMatch "Microsoft Data Access Internet Publishing Provider"  
redirect-carefully  
BrowserMatch "^WebDrive" redirect-carefully  
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully  
BrowserMatch "^gnome-vfs" redirect-carefully
```

UNCLASSIFIED

This page intentionally left blank.

## Appendix C

Sample configuration of the mod\_security module. These lines should be included in the main Apache configuration file httpd.conf.

```
<IfModule mod_security.c>

# Turn the filtering engine On or Off
SecFilterEngine On

# Make sure that URL encoding is valid
SecFilterCheckURLEncoding On

# Only allow bytes from this range
SecFilterForceByteRange 32 126

# The audit engine works independently and
# can be turned On or Off on the per-server or
# on the per-directory basis
SecAuditEngine RelevantOnly

# The name of the audit log file
SecAuditLog logs/audit_log

# If this is SUSE comment out previous log file and add this one
# SecAuditLog /var/log/apache2/audit_log

SecFilterDebugLog logs/modsec_debug_log
SecFilterDebugLevel 0

# If this is SUSE comment out previous log file and add this one
# SecFilterDebugLevel 0

# Should mod_security inspect POST payloads
SecFilterScanPOST On

# Action to take by default
SecFilterDefaultAction "deny,log,status:406"

# Redirect user on filter match
SecFilter xxx redirect:http://www.webkreator.com

# Execute the external script on filter match
SecFilter yyy log,exec:/home/ivanr/apache/bin/report-attack.pl

# Simple filter
SecFilter 111

# Only check the QUERY_STRING variable
SecFilterSelective QUERY_STRING 222

# Only check the body of the POST request
SecFilterSelective POST_PAYLOAD 333

# Only check arguments (will work for GET and POST)
SecFilterSelective ARGS 444
```

## UNCLASSIFIED

```
# Test filter
SecFilter "/cgi-bin/keyword"

# Another test filter, will be denied with 404 but not logged
# action supplied as a parameter overrides the default action
SecFilter 999 "deny,nolog,status:404"

# Prevent OS specific keywords
SecFilter /etc/password

# Prevent path traversal (..) attacks
SecFilter "\.\./"

# Weaker XSS protection but allows common HTML tags
SecFilter "<( |\n)*script"

# Prevent XSS attacks (HTML/Javascript injection)
SecFilter "<(.\|\n)+>"

# Very crude filters to prevent SQL injection attacks
SecFilter "delete[[:space:]]+from"
SecFilter "insert[[:space:]]+into"
SecFilter "select.+from"

# Require HTTP_USER_AGENT and HTTP_HOST headers
SecFilterSelective "HTTP_USER_AGENT|HTTP_HOST" "^$"

# Forbid file upload
SecFilterSelective "HTTP_CONTENT_TYPE" multipart/form-data

# Only watch argument p1
SecFilterSelective "ARG_p1" 555

# Watch all arguments except p1
SecFilterSelective "ARGS|!ARG_p2" 666

# Only allow our own test utility to send requests (or Mozilla)
SecFilterSelective HTTP_USER_AGENT "!(mod_security|mozilla)"

# Do not allow variables with this name
SecFilterSelective ARGS_NAMES 777

# Do now allow this variable value (names are ok)
SecFilterSelective ARGS_VALUES 888

# Stop spamming through FormMail
# note the exclamation mark at the beginning
# of the filter - only requests that match this regex will
# be allowed
<Location /cgi-bin/FormMail>
    SecFilterSelective "ARG_recipient" "!@webkreator.com$"
</Location>
```

## UNCLASSIFIED

```
# when allowing upload, only allow images
# note that this is not foolproof, a determined attacker
# could get around this
<Location /fileupload.php>
    SecFilterInheritance Off
    SecFilterSelective POST_PAYLOAD "!image/(jpeg|bmp|gif)"
</Location>

</IfModule>
```

UNCLASSIFIED

This page intentionally left blank.

## Appendix D

### Description of loadable modules.

LoadModule access_module	Provides access control based on client hostname, IP address, or other characteristics of the client request.
#LoadModule auth_module	User authentication using text files
#LoadModule auth_anon_module	Allows "anonymous" user access to authenticated areas
#LoadModule auth_dbm_module	Provides for user authentication using DBM files
#LoadModule auth_digest_module	User authentication using MD5 Digest Authentication
#LoadModule include_module	Server-parsed html documents (Server Side Includes)
LoadModule log_config_module	Logging of the requests made to the server
#LoadModule env_module	Modifies the environment which is passed to CGI scripts and SSI pages
#LoadModule mime_magic_module	Determines the MIME type of a file by looking at a few bytes of its contents
#LoadModule cern_meta_module	CERN httpd metafile semantics
#LoadModule expires_module	Generation of Expires and Cache-Control HTTP headers according to user-specified criteria
#LoadModule deflate_module	Compress content before it is delivered to the client
#LoadModule headers_module	Customization of HTTP request and response headers
#LoadModule usertrack_module	Clickstream logging of user activity on a site
#LoadModule unique_id_module	Provides an environment variable with a unique identifier for each request
LoadModule setenvif_module	Allows the setting of environment variables based on characteristics of the request
LoadModule mime_module	Associates the requested extensions with the file's behavior (handlers and filters) and content (mime-type, language, character set and encoding)
#LoadModule dav_module	Authoring and Versioning (WebDAV) functionality
#LoadModule status_module	Information on server activity and performance
#LoadModule autoindex_module	Generates directory indexes, automatically, similar to the UNIX <code>ls</code> command or the Win32 <code>dir</code> shell command
#LoadModule asis_module	Sends files that contain their own HTTP headers
#LoadModule info_module	Provides a comprehensive overview of the server configuration

## UNCLASSIFIED

<code>#LoadModule dav_fs_module</code>	filesystem provider for mod_dav
<code>#LoadModule vhost_alias_module</code>	Provides for dynamically configured mass virtual hosting
<code>#LoadModule negotiation_module</code>	Provides for content negotiation
<code>LoadModule dir_module</code>	Provides for "trailing slash" redirects and serving directory index files
<code>#LoadModule imap_module</code>	Server-side imagemap processing
<code>#LoadModule actions_module</code>	This module provides for executing CGI scripts based on media type or request method.
<code>#LoadModule speling_module</code>	Attempts to correct mistaken URLs that users might have entered by ignoring capitalization and by allowing up to one misspelling
<code>#LoadModule userdir_module</code>	User-specific directories
<code>LoadModule alias_module</code>	Provides for mapping different parts of the host filesystem in the document tree and for URL redirection
<code>LoadModule rewrite_module</code>	Provides a rule-based rewriting engine to rewrite requested URLs on the fly
<code>#LoadModule proxy_module</code>	HTTP/1.1 proxy/gateway server
<code>#LoadModule proxy_ftp_module</code>	FTP support module for mod_proxy
<code>#LoadModule proxy_http_module</code>	HTTP support module for mod_proxy
<code>#LoadModule proxy_connect_module</code>	mod_proxy extension for CONNECT request handling
<code>#LoadModule cache_module</code>	Content cache keyed to URIs
<code>#LoadModule suexec_module</code>	Allows CGI scripts to run as a specified user and Group
<code>#LoadModule disk_cache_module</code>	Content cache storage manager keyed to URIs
<code>#LoadModule file_cache_module</code>	Caches a static list of files in memory
<code>#LoadModule mem_cache_module</code>	Content cache keyed to URIs
<code>#LoadModule cgi_module</code>	Execution of CGI scripts

## Appendix E

### License Information

#### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect

## UNCLASSIFIED

making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

### GNU GENERAL PUBLIC LICENSE

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under

## UNCLASSIFIED

these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

## UNCLASSIFIED

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

## UNCLASSIFIED

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD

UNCLASSIFIED

PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## References

- Anonymous. Maximum Linux Security. Indianapolis: Sams Publishing, 2000.
- Bauer, Michael D. Building Secure Servers with Linux: Tools & Best Practices for Bastion Hosts. Sebastopol: O'Reilly & Associates, 2003.
- Mourani, Gerhard. Securing and Optimizing Linux: Red Hat Edition. Version 1.3. n.p.: OpenDocs Publishing, 2000.
- Mourani, Gerhard. Securing and Optimizing Linux: The Hacking Edition. Version 3.0. n.p.: OpenDocs Publishing, 2002.
- National Security Agency. Guide to the Secure Configuration of Solaris 8. Fort Meade: National Security Agency, 2003.
- Red Hat, Inc. Red Hat Linux 9: Red Hat Linux Security Guide. Raleigh: Red Hat, Inc., 2002.
- Mobility, Tony. Hardening Apache. New York, New York: Apress, 2004.
- Ristic, Ivan. mod\_security: Reference Manual v1.8.4. n.p.: n.p. 2004  
<http://modsecurity.org/documentation/modsecurity-manual.pdf>
- Ristic, Ivan. Introducing mod\_security  
[http://www.onlamp.com/pub/a/apache/2003/11/26/mod\\_security.html](http://www.onlamp.com/pub/a/apache/2003/11/26/mod_security.html)
- Maj, Artur. Securing Apache: Step-by-Step n.p.: n.p. 2003.  
<http://www.securityfocus.com/infocus/1694>
- Apache.org  
<http://httpd.apache.org/docs-2.0/>
- Jay Beale's Unix Security Site  
<http://www.bastille-linux.org/jay/>
- Coar, Ken. Security and Apache: An Essential Primer  
<http://linuxplanet.com/linuxplanet/print/1527/>
- Coar, Ken. Using Apache with Suexec on Linux: Executing CGI Scripts as Other Users.  
<http://www.linuxplanet.com/linuxplanet/tutorials/1445/1/>
- Matilla, Jan. Chrooting Apache2 howto  
<http://www.cgisecurity.com/webserver/apache/chrootapache2-howto.html>
- Lowe, Scott. Five ways to address Apache CGI security concerns  
<http://techrepublic.com.com/5100-6264-1058416.html>
- The World Wide Web Security FAQ  
<http://www.cgi-security.com/>  
<http://www.w3.org/Security/Faq/wwwsf4.html>
- Barnett, Ryan. SECURING APACHE STEP BY STEP  
[http://www.cgisecurity.com/lib/ryan\\_barnett\\_gcux\\_practical.html](http://www.cgisecurity.com/lib/ryan_barnett_gcux_practical.html)
- Stein, Lincoln Apache Security from A-Z  
[http://stein.cshl.org/~lstein/talks/perl\\_conference/apache\\_security/](http://stein.cshl.org/~lstein/talks/perl_conference/apache_security/)
- Cox, Mark. Apache Security Secrets: Revealed. 2002.  
<http://www.cgisecurity.com/webserver/apache/tu04-handout.pdf>
- Hal Pomeranz and Deer Run Associates. WU-FTPD and Apache Security Basics. 2002  
<http://www.cgisecurity.com/webserver/apache/BayLISAApacheWUFTP.pdf>
- Klein, Amit. Cross Site Scripting Explained  
<http://crypto.stanford.edu/cs155/CSS.pdf>  
<http://www.lbl.gov/ITSD/Security/systems/apache-server.html#sec1>

UNCLASSIFIED

Warrene, Blane. Securing Your Apache 2 Server with SSL. 2004

<http://www.sitepoint.com/article/securing-apache-2-server-ssl>

<http://www.faqs.org/docs/Linux-HOWTO/SSL-RedHat-HOWTO.html#ss1.1>

<http://www.openssl.org/>