

RED HAT SECURE WEB SERVER 3.0
DEVELOPER EDITION FOR COBALT NETWORKS SERVERS



Cobalt Secure Web Server (SSL) Cobalt Networks, Inc

Features:

- 128 bit Encryption*
- Based on Redhat's Secure Server
- Apache 1.3.6
- SSL 3.0
- Certificate Generation
- Secure Server Documentation
- mod_ssl Documentation
- Single Server Advanced Cryptography License from RSA Data Security
- BSAFE Encryption Engine from RSA Data Security

Limitations:

*Due to restrictions imposed by the US Government, this software may not be distributed outside the U.S. and Canada

© 1998, 1999 Cobalt Networks, Inc. Contains Software & Licenses from the following organizations/companies: RedHat Software, RSA Data Security.



Contents

- 1. Introduction**
- 2. Installing your Secure Server**
 - 1.1 Installing the Package File**
 - 1.2 Configuring your server**
 - 1.2.1 Default Configuration**
 - 1.2.2 Configuration files**
 - 1.2.3 Generating a test certificate**
 - 1.2.4 Starting the server**
 - 1.3 Obtaining a Certificate**
 - 1.3.1 About Passwords**
 - 1.3.2 Certificate Authorities**
 - 1.3.3 Gather Proof of Right Documents**
 - 1.3.4 Generate a CSR (Certificate Signing Request)**
 - 1.3.5 Submit your information**
 - 1.4 Installing your Certificate**
- 3. Keeping your Server Secure**
- 4. Customizing your Server**
 - 4.1 Directories & File Locations**
- 5. Links**
- 6. FAQ**
- 7. Contacting Cobalt**
- 8. Acknowledgements**
- 9. License & Warranties**

1. Introduction

Red Hat Secure Web Server 3.0 Developer Edition is designed for developers.

Cobalt Networks release of Red Hat Secure Web Server is designed for developers who wish to deploy applications that use encrypted communication conforming to SSL (Secure Sockets Layer) standards. The core technology behind the Secure Web Server is the Apache 1.3.6 web server with SSL extensions, an implementation of Netscape's Secure Sockets Layer. In order to use this software effectively, you will need to have some familiarity with the Unix operating system, specifically directory structures & telnet.

About our Implementation

The Red Hat Secure Web Server version 3 is an implementation of mod_ssl. There are two important differences between Cobalt Network's release of Red Hat Secure Web Server and the publicly available version of Apache with mod_ssl.

1. The Red Hat Secure Web Server contains a license for RSA's BSAFE encryption library. The BSAFE library contains routines that make SSL communications possible. It is necessary to have this license to deploy commercial SSL applications in the US and Canada.
2. RSA and Redhat Software have ported and optimized the BSAFE routines for the MIPS architecture used by the Cobalt Networks server family. Encryption performance is superior to standard SSLeay routines.

If you want to include other modules to customize the Apache server, you can use modules in DSO format that comply with Apache 1.3.6. You must have a separate license for each Cobalt Networks server. Cobalt Networks provides no support for you to recompile and customize Apache.

Installing the Red Hat Secure Web Server package

Installing the Package file

The procedure for installing the package is very easy.

Start by downloading the appropriate package to your hard disk. Then connect to the Admin server of the Cobalt Qube. Navigate to the install software tab in the maintenance section.

Type in the pathname and filename of the package file. Alternately, click on the browse button and select the package file from your hard disk. Then enter the admin password and click the "install a '.pkg' Package" button.

The install process takes about a minute. You will know that the software is installed when you see "Red Hat Secure Web Server Release 3" listed in the Software window.

Configuring your server

Default configuration:

The secure server runs from `/home/httpd/html/`

HTML pages are served from `/home/httpd/html/` by default. On the Cobalt RaQ 2, only the admin user may ftp files to this directory.

Configuration Files

The configuration file for the secure web server is:
`/etc/httpd/conf/httpsd.conf`

Make a backup before modifying `/etc/httpd/conf/httpsd.conf`. This file contains SSL specific configuration options.

HostnameLookups: `off` /`on`. This determines whether hostnames or IP addresses are logged. Setting this to `on` results in a nominal performance penalty since the DNS software must lookup the address of the visiting host.

User: `httpd`. This option sets the user that will be running the server process. Cobalt uses the `httpd` user for the public site. Do not attempt to run the server under any other user as server instability will result.

Group: `httpd`. This option sets the group that will be running the server process. Do not attempt to run the server under any other group as server instability will result.

TransferLog: `/var/log/httpd/access_log-ssl`. This determines the location of the host access log file. Web statistics and report software use the access log file to generate reports.

ErrorLog: `/var/log/httpd/error_log-ssl`. This determines the location and name of the error log file. If your server fails to start properly, this file will contain the error notice.

ServerAdmin: `root@localhost`. This determines the email address of your server's administrator. This is normally also the server administrator.

To learn more about other options, consult the Apache 1.3.6 documentation at <http://www.apache.org>.

Generating a Certificate and Private Key

In order to start your secure server, you must first generate a certificate.

1. Telnet to the server, authenticate as the user “admin”.
2. Re-authenticate as the user “root” using the same password as was used to authenticate as the user “admin” by typing ‘su -’ from the command line.
3. Change directories to the web server configuration directory by typing ‘cd /etc/httpd/conf’.
4. Make an SSL key by typing ‘make genkey’. You will need to type in a passphrase that will be used when generating certificates and when starting the secure web server. **Do not forget this passphrase!**
5. Make a certificate request by typing ‘make certreq’. You will need to specify some information about the web site and the exact server name to be used.

Country Name: the two-letter code for your country (ex. US, CA)

State or Province: The state or province name spelled out (ex. California)

Locality Name: The name of your city

Organization Name: Your company or organization name

Organization Unit: Your department or company section

Server Host Name: Your hostname & domain name (ex. ssl.cobaltnet.com)

Email Address: The webmaster administrator address (admin@xyz.com)

If you are only generating a test certificate, not a production (registered) certificate, you may jump to the next step. The certificate request file is placed in /etc/httpd/conf/ssl.csr/server.csr. The contents of this file must be submitted to a certificate authority such as Thawte or Verisign. The certificate generated by the certificate authority must be placed in the file /etc/httpd/conf/ssl.crt/server.crt. If you need to change the server name, then you will have to re-generate the certificate request and re-register the secure server certificate with certificate authority.

You may generate a test certificate to test your server while waiting for a real certificate from the certificate authority. If you choose to generate a test certificate yourself, type “make testcert”. This will place the temporary (unregistered) certificate in the file /etc/httpd/conf/ssl.crt/server.crt. You will need to replace the contents of this file once you obtain the registered certificate from the certificate authority.

Starting the Server

Always use the script **`/etc/rc.d/init.d/httpsd`** to start, stop and restart the server.

You can start the server manually by typing in:

`/etc/rc.d/init.d/httpsd start`

If there is no error, you can use your browser to connect to the server.

You can also verify that the server is running from the command line,

type: **`ps uax | grep httpsd`**

To stop the server, type:

`/etc/rc.d/init.d/httpsd stop`.

To restart the server, type:

`/etc/rc.d/init.d/httpsd restart`

When you first connect to the secure server from your browser, you will have to go through a certificate authentication process. This is because the server's digital ID is not guaranteed by any CA (Certificate Authority).

Obtaining a Certificate

About Passwords

If you encrypt your server with a PEM passphrase, the server will be unable to start the secure server by itself. This is because Apache-SSL requires that you enter the passphrase when you start the program. If you want the secure server to startup automatically, you should not use a PEM passphrase. Be aware that security is compromised if a PEM passphrase is not used.

Certificate Authorities

You can obtain a certificate from a number of Certificate Authorities. The two most common authorities are Verisign and Thawte.

Company	Certificate Cost	Term	Renewal Fee	Web Site
Verisign	\$349	1 year	\$249	www.verisign.com
Thawte	\$125	1 year	\$100	www.thawte.com

All modern browsers (IE 3 or newer & Netscape 3 or newer) work with certificates from either authority.

Step 1: Gather Proof of Right Documents

Certificate Authorities require some form of proof that you are who you claim you are. This can include a number of things. Check with your CA for details.

- Your DUNS Number (Verisign only)
To obtain a DUNS number, visit: <http://www.dnb.com/dunsno/whereduns.htm>
- Proof of your right to use the certified organization name (Articles of incorporation)
- Proof of your registration of the domain name (from the InterNIC whois database)
To obtain your domain details, visit: <http://rs.internic.net/>
- A letter of authorization from an agent of your company or organization
Thawte has an online form that generates this letter

Step 2: Generate a CSR (Certificate Signing Request)

Please refer to the instructions titled **Generating a Certificate**.

Important Configuration Files:

SSL Certificate File **/etc/httpd/conf/ssl.crt/server.crt** (will be replaced by CA key)
SSL Certificate Key File **/etc/httpd/conf/ssl.key/server.key**
Certificate Request File **/etc/httpd/conf/ssl.csr/server.csr** (submitted to the CA)
Apache Configuration **/etc/httpd/conf/httpd.conf**

The file **/etc/httpd/conf/ssl.crt/server.crt** is your self-signed certificate. You use it as a temporary certificate while you are waiting for a real certificate from your CA.

The file **/etc/httpd/conf/ssl.csr/server.csr** is your CSR (certificate request). The important bit looks something like this:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBPTCB6AIBADCBhDELMAkGA1UEBhMCWkExFTATBgNVBAgTDFdlc3Rlcm4gQ2Fw
ZTESMBAGA1UEBxMJQ2FwZSBUb3duMRQwEgYDVQQKEwtPcHBvcnR1bml0aTEYMBYG
A1UECxMPT25saW5lIFNlcnZpY2VzMR0wGAYDVQQDExF3d3cuZm9yd2FyZC5jby56
YTBaMA0GCSqGSIb3DQEBAQUAA0kAMEYCCQDT5oxxeBWu5WLHD/G4BJ+PobiC9d7S
6pDvAjuyC+dPanL0d91tXdm2j190D1kgDoSp5ZyGSgwJh2V7diuuPlHDagEDoAAw
DQYJKoZIhvcNAQEEBQADQQBf8ZHIu4H8ik2vZQngXh8v+iGnAXD1AvUjuDPCWzFu
pReiq7UR8Z0wiJBeaqiuvtDnTFMz6oCq6htdH7/tvKhh
-----END NEW CERTIFICATE REQUEST-----
```

(This section reprinted/modified with permission of Thawte Consulting: <http://www.thawte.com/certs/server/keygen/apachessl.html>)

Step 3: Submit your information

Once you have your information prepared, decide which Certificate Authority you want to register with. The two most common CA's in the US are Verisign & Thawte. Here are links to their registration pages:

Verisign - <http://digitalid.verisign.com/server/enrollIntro.htm>
Thawte - <http://www.thawte.com/certs/server/request.html>

Each Service requires that you generate and submit a Certificate Signing Request (CSR), which you created in Step 2. Copy the contents of **/etc/httpd/conf/ssl.csr/server.csr** into your Certificate Authority web submission form. Submit the other necessary data and wait for the certificate to be returned to you.

Installing your Certificate

Once you receive your certificate back from the CA, you should replace the contents of the certificate file: `/etc/httpd/conf/ssl.crt/server.crt`

Restart your web server, `/etc/rc.d/init.d/httpd restart`, and you are ready to server secure pages to everyone on the Internet!

Keeping your Server Secure

If you're planning to develop commerce applications on a Cobalt Networks server, you want your transaction environment to be as secure as possible. It's important to follow a few design guidelines.

IMPORTANT: Running a secure server means that the connection to your customer/client is secure. Be proactive with your site design. Don't let other data-flow processes interrupt that security.

1. Never store unencrypted credit card numbers on the server - This holds true for customer data in general.
2. Never send credit card or other transaction data over the network unencrypted
If you send transactions over the net for latter processing via email, be sure to encrypt the data before you send it. Consider using ssh to send data.
3. If at all possible, don't store your PEM password on the server
This may not be practical if you must restart your server automatically. Cobalt Networks servers are designed to run for months (even years) at a time without rebooting. Invest in a UPS.
4. Maintain a checksum on your server data (html, cgi, static databases) to warn you of attacks.
5. Backup your PEM password and key in a safe place.

Customizing your Server

Assumption: You understand how to administer the Apache web server. For documentation about the Apache server, see <http://www.apache.org>.

Cobalt Networks servers run several web servers simultaneously.

1. The Public server (port 80)
2. The Admin server (port 81)
3. The Secure server (port 443)

Why would I want to customize my server?

You may want to customize the web root directory. Customizing the server may be necessary to produce commerce applications.

Customization Directions

Note: The Public Server configuration files are also in /etc/httpd/conf - don't edit these by mistake! The secure web server configuration files are /etc/httpd/conf/httpsd.conf (a symbolic link to httpd-ssl.conf), /etc/httpd/conf/srm-ssl.conf (not used by Apache 1.3.6) and /etc/httpd/conf/access-ssl.conf (not used by Apache 1.3.6.)

Each configuration file (httpsd.conf, access-ssl.conf, srm-ssl.conf) may be edited using your own Apache directives. You can review the entire list of Apache directives at: <http://www.apache.org/docs/mod/directives.html>.

Links

Cobalt Networks: <http://www.cobaltnet.com/>

Apache-SSL: <http://www.apache-ssl.org/>

Apache Quick Reference Card: <http://www.ford-mason.co.uk/resources/apache-refcard/>

Apache Perl Integration Project: <http://perl.apache.org/>

PHP HyperText Processor: <http://www.php.net/>

Apache-SSL SRPMs: <http://www.replay.com/redhat/apache.html>

Mod_SSL: http://www.engelschall.com/sw/mod_ssl/

Contacting Cobalt Networks, Inc.

Cobalt Networks, Inc
555 Ellis St.
Mountain View, Calif. 94043
Tel: (650) 930-2500
Fax: (650) 930-2501
Web: <http://www.cobaltnet.com/>

Acknowledgements

©1998, 1999 Cobalt Networks, Inc. All rights reserved.

Cobalt Networks, Cobalt Qube and Cobalt RaQ are trademarks of Cobalt Networks, Inc. All other company, brand, and product names may be registered trademarks or trademarks of their respective companies and are hereby recognized.

License & Warranties

©1998, 1999 Cobalt Networks, Inc., RedHat Software, RSA Data Security All rights reserved.

PORTIONS OF THIS PRODUCT ARE COVERED UNDER THE GNU GENERAL PUBLIC LICENSE

THIS PRODUCT MAY NOT BE EXPORTED TO, OR SOLD TO A NATIONAL OF, ANY COUNTRY OTHER THAN THE UNITED STATES AND CANADA.

THIS SOFTWARE IS PROVIDED BY COBALT NETWORKS ``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 DISCLAIMED. IN NO EVENT SHALL COBALT NETWORKS OR 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, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This publication and the information herein is furnished AS IS, subject to change without notice, and should not be construed as a commitment by Cobalt Networks, Inc. Furthermore, Cobalt Networks, Inc., assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes and noninfringement of third party right.