



How to set up Apache
to use Raak Smart Card certificates
for user authentication

Table of contents

Introduction	3
Requirements	3
Configure Apache to accept certificates issued by the Raak CA.....	3
<i>If you don't have a pre-existing CA Certificate File:</i>	<i>4</i>
<i>If you do have a pre-existing CA Certificate File:.....</i>	<i>4</i>
Set Apache to require client certificates	4
Set which users are allowed access.....	5
<i>Example 1: Allow access to anyone from your company and department.....</i>	<i>5</i>
<i>Example 2: Block access to a specific user</i>	<i>5</i>
<i>Example 3: Allow access to a specific user.....</i>	<i>6</i>
<i>Example 4: Using the .passwd file.....</i>	<i>6</i>
<i>Example 5: Using the httpd.conf file</i>	<i>7</i>
References	8

Now you must point Apache to the certificate. This is done by adding the Raak certificate to the "CA Certificate File". This file is defined by the "SSLCACertificateFile" configuration directive in the httpd.conf file.

Note: If your web site is hosted by an ISP you may not have access to the httpd.conf file. Contact your ISP help desk to find out how to proceed.

If you don't have a pre-existing CA Certificate File:

Place the Raak root certificate file in a sub directory of the apache folder (for example in apache/conf/ssl).

Open the httpd.conf file (The httpd.conf file is usually found in the apache/conf/ directory). Add the following to the file:

Content of the httpd.conf file:

```
SSLCACertificateFile conf/ssl/raakroot.cer
```

Stop and Start the apache server, so that the changes take effect.

Note: If Apache refuses to start after making changes to your httpd.conf file, check the text carefully for spelling mistakes.

If you do have a pre-existing CA Certificate File:

Append the contents of the raakroot.cer file to the end of your file.

Stop and Start the apache server, so that the changes take effect.

Warning: If there are multiple Root certificates in the CA Certificate file, certificates signed by those certificates will also be accepted by the examples below. Make sure you trust these root certificates not to sign fraudulent user certificates.

Set Apache to require client certificates

Now you must configure the directories that you want to secure so that they will require SSL user authentication. A typical web site will be structured so that one branch of the site will require access control. This is done by adding a ".htaccess" file to the top directory of the branch that you want to secure.

Add the following directives to the .htaccess file:

Content of the .htaccess file:

```
SSLCipherSuite HIGH:MEDIUM
SSLVerifyClient require
SSLVerifyDepth 2
SSLRequireSSL
```

SSLCipherSuite : require strong encryption of the data.
SSLVerifyClient : require client authentication
SSLVerifyDepth 2: set Apache to accept an intermediate CA certificate
SSLRequireSSL : set Apache to require HTTPS, and disallow HTTP

The settings for this directory will be valid for all the sub directories. It sets Apache to require a client certificate to access any files in the directory. Anyone with a valid Raak certificate will be able to access the directory.

Set which users are allowed access

With the above setting anyone with a valid certificate will be allowed access to the controlled directories. Now you want to specify which specific users will be allowed access. This is done by requiring Apache to check for specific data in the “distinguished name” (DN) of the client certificate. The distinguished name has the following components that can be checked:

	ID	Name	Directive identifier	Example
Name	CN	Certificate Name	SSL_CLIENT_S_DN_CN	<i>Bob Smith</i>
Email address	Email	Email	SSL_CLIENT_S_DN_Email	<i>bob@acmetech nologies.com</i>
Company name	O	Organization	SSL_CLIENT_S_DN_O	<i>Acme Technologies</i>
Department	OU	Organization Unit	SSL_CLIENT_S_DN_OU	<i>Sales</i>
City	L	Location	SSL_CLIENT_S_DN_L	<i>Austin</i>
State	ST	State	SSL_CLIENT_S_DN_ST	<i>Texas</i>
Country	C	Country	SSL_CLIENT_S_DN_O	<i>US</i>

The following are some examples of ways that these are used.

Note: The first four examples describe how to manage access through the .htaccess file. For a discussion of how to use the httpd.conf file see example 5.

Example 1: Allow access to anyone from your company and department

The following example limits access to certificates where

- Organization = Acme Technologies and
- Organization Unit = sales or admin

Content of the .htaccess file:

```

SSLCipherSuite HIGH:MEDIUM
SSLVerifyClient require
SSLVerifyDepth 2
SSLRequireSSL

SSLOptions +FakeBasicAuth
SSLRequire %{SSL_CLIENT_S_DN_O} eq "Acme Technologies" and \
           %{SSL_CLIENT_S_DN_OU} in {"sales", "admin"}
    
```

Example 2: Block access to a specific user

The example above shows how you can use certificates to effectively control groups of users based on the content of their certificates. The advantage of this approach is that you do not have to change any information to add a user. All that is needed to add a user is to issue them a Raak

Smart card or USB token.

In certain situations, however, you may want to block a specific user from a group of users that are otherwise acceptable. For example, this is the case when you have revoked a user but have not received back the smart card, or when a smart card has been lost or stolen.

The following example limits access to certificates where

Organization = Acme Technologies and
Certificate Name NOT Alice Jones or Bob Smith

Content of the .htaccess file:

```
SSLCipherSuite HIGH:MEDIUM
SSLVerifyClient require
SSLVerifyDepth 2
SSLRequireSSL

SSLOptions +FakeBasicAuth
SSLRequire %{SSL_CLIENT_S_DN_O} eq "Acme Technologies" and \
          !( %{SSL_CLIENT_S_DN_CN} in {"Alice Jones", "Bob
Smith"})
```

Example 3: Allow access to a specific user

The above two examples will allow you to manage most common scenarios. However, in certain situations you may want to provide access to a specific user that is not part of a larger group.

The following example limits access to certificates where

Organization = Acme Technologies and
Certificate Name = Alice Jones or Bob Smith

Content of the .htaccess file:

```
SSLCipherSuite HIGH:MEDIUM
SSLVerifyClient require
SSLVerifyDepth 2
SSLRequireSSL

SSLOptions +FakeBasicAuth
SSLRequire %{SSL_CLIENT_S_DN_O} eq "Acme Technologies" and \
          %{SSL_CLIENT_S_DN_CN} in {"Alice Jones", "Bob Smith"}
```

Example 4: Using the .passwd file

There may be certain scenarios where you want to control access to a large group of specific users. For example, this may be the case if the user certificates don't provide a straightforward way of grouping users by company name or department. If this is the case it is possible to use the standard Apache authentication methods using a .passwd file.

The .passwd file contains an entry for each specific user, and is referenced in the .htaccess file using the AuthUserFile directive. In this example the .passwd file is located in the "conf" directory.

The .passwd file is structured as a list of "username:password". In using it with certificates, the username is replaced with the "Distinguished Name" field of the certificate contents. The password is not used for authentication, but the word "password" is required as a placeholder.

Note: The username has to have the exact data contained in the Distinguished Name otherwise authentication will fail. An easy way of finding the exact Distinguished Name is looking at the access log if the authentication fails.

Note: If your server encrypts passwords you will have to provide the encrypted version of the word "password" otherwise authentication will fail.

Note: If you receive a login dialog requesting a username and password in trying this method, then there is a problem with your .passwd file. Check the username data, the path, and the password.

Content of the .htaccess file:

```
SSLCipherSuite HIGH:MEDIUM
SSLVerifyClient require
SSLVerifyDepth 2
SSLRequireSSL

SSLOptions +FakeBasicAuth
AuthName "Acme Technologies PW Authentication"
AuthType Basic
AuthUserFile /usr/local/apache/conf/httpd.passwd
require valid-user
```

```
AuthName (not used for certificate authentication)
AuthType Use .passwd file based authentication
AuthUserFile Pointer to the file, and directive to check validity
Require All valid users can access the directory
```

Content of the .passwd file:

```
/Email=jane@acmetechnologies.com/C=US/ST=Texas/L=Austin/O=Acme
Technologies/OU=admin/CN=Alice Jones:password
/Email=bob@acmetechnologies.com/C=US/ST=Texas/L=Austin/O=Acme
Technologies/OU=sales/CN=Bob Smith:password
```

Example 5: Using the httpd.conf file

The examples above are well suited to virtual host environments where the administrator determining access privileges does not have access to the httpd.conf file. If you do have access to this file, it is recommended that you place the access control directives in the httpds.conf file using <location> or <directory> directives.

The following example limits access to the sub directory "secure" to certificates where

Organization = Acme Technologies

Content of the httpd.conf

```
<Location /secure>
SSLCipherSuite HIGH:MEDIUM
SSLVerifyClient require
SSLVerifyDepth 2
SSLRequireSSL

SSLOptions +FakeBasicAuth
SSLRequire %{SSL_CLIENT_S_DN_O} eq "Acme Technologies"
</Location>
```

References

For support with Raak Smart Cards and USB Tokens see:
<http://www.raaktechnologies.com>

For details on Apache see:

Authentication: <http://httpd.apache.org/docs-2.0/howto/auth.html>

HTAccess: <http://httpd.apache.org/docs-2.0/howto/htaccess.html>

For more information on mod_ssl see:

Reference: http://www.modssl.org/docs/2.8/ssl_reference.html

How to: http://www.modssl.org/docs/2.8/ssl_howto.html