Configure Apache Web Server for Client Auth

This page describes configuration of the Apache Web Server (httpd) on the Bamboo Services Platform (that is, on the host running the FUSE ESB instance to which centrally-hosted Bamboo services are deployed). Example configuration files, *httpd.conf* and *ssl.conf*, that were in active use as of March 2013 are in the Project Bamboo code repository at:

http://svn.code.sf.net/p/projectbamboo/code/platform-config/trunk/client-auth/

The last section of this page describes a number of client responsibilities in establishing trust via client auth. Client generation of X.509 certificates and configuration of selected clients are covered in the *Certificate Exchange* section of this page. (Note that additional client configuration necessary for the client to operate as a Shibboleth Service Provider is included in this documentation on a separate page: *Shibboleth SP Installation and Configuration for Bamboo Trust Federation Clients*.)

Trusted (known) client applications (those calling services on the Bamboo Services Platform) are expected to participate in client auth. This might be effected by running Apache Web Server (httpd) on the client application's host, or by other means. Modest suggestions are made in this document regarding generation and handling of X.509 certificates exchanged with the Bamboo Services Platform host in order to (physically) establish trust; these suggestions assume are given in the contexts of a simple test client (Firefox Poster); and a PHP application. Other clients' mileage will undoubtedly vary.

The (physical) establishment of trust effects the step described on the page *Identity and Access Management - Authentication and Authorization* as "*registering as a member of the Bamboo Trust Federation by exchanging and publishing appropriate metadata [...].*"

Clients that are *not* trusted (known) are treated as anonymous for purposes of policy enforcement (access control, permissions). Anonymous clients may not assert the identity of a known users, so requests made from an anonymous client application are treated as if an anonymous user were making the request.

Overview

These instructions are drawn from Project Bamboo's practical experience installing and configuring Apache Web Server (httpd) v2.2.22 to handle client authentication via exchange of X.509 certificates. The virtual machines on which these instructions were proven were Fedora 16-64bit VMs hosted by Linode.com.

The client configuration suggestions given at the end of this document reflect experience configuring Firefox Poster and the Drupal-based Account Services Module for participation in the Bamboo Trust Federation.

Apache Web Server (httpd) on the Bamboo Services Platform

Per the *info* bloc at the top of this page, the following documents configuration of the Apache Web Server (httpd) v2.2.22 on the Bamboo Services Platform (that is, on the host running the FUSE ESB instance to which centrally-hosted Bamboo services are deployed).

Install Apache Web Server (httpd)

Installation of httpd is well-documented for most platforms. YUM or similar installation is what's probably wanted for a Linux box / VM.

- install httpd, e.g., *sudo yum install httpd*
- install mod_ssl, e.g., *sudo yum install mod_ssl*

On the Linode VMs used by Project Bamboo, the GUI client Webmin further simplified installation of httpd; from that interface, it's a point-and-click operation.

Certificates and Certificate Directories

A certificate, private key, and intermediate certificate chain can be obtained from a certificate granting authority. GoDaddy was the commercial authority used by Project Bamboo.

These files should be placed in the location appropriate to the server on which httpd is installed. On the Fedora 16 VMs used by Project Bamboo, that directory was:

```
/etc/pki/tls/certs
```

Note that the key must be made **root readable/writable** only; the crt files should also be readable by group & other. This directory and the filenames will be referenced in configuration set in ssl.conf; the examples given use filenames of GoDaddy certs utilized by Project Bamboo, others' filenames will differ.

Configure Apache Web Server (httpd)

The following examples are taken from configuration for Apache Web Server (httpd) v. 2.2.22 on a server whose DNS name was *bsp-dev.projectbamboo.org*. Please consult the httpd project documentation for current configuration advice.

Specific instructions listed here include ONLY changes necessary to freshly-installed configuration files for httpd v2.2.22.

- Changes to these files that have to do directly with client auth; *plus* ProxyPass directives that come into play in forwarding calls to FUSE ESB (the Bamboo Services Platform).
- *Not* described here are ProxyPass directives that may come into play if Grouper is installed on the same server as FUSE ESB; those directives are described on the page *Grouper Install - Configure - Populate*.

Configure httpd.conf

Set ServerName to valid DNS name for the host

```
#ServerName www.example.com:80

# ==> PERSON mod DD MM YYYY
# set server name properly

ServerName bsp-dev.projectbamboo.org

# <== end of mod
```

Set UseCanonicalName to use the ServerName value as host name in self-referencing URLs

```
# UseCanonicalName Off

# ==> PERSON mod DD MM YYYY

UseCanonicalName On

# <== end of mod
```

Configure ssl.conf

Add logging node if mod_ssl is running

(your location for the SSL Engine log may differ)

```
# <==
# PERSON mod DD MM YYYY
#
<IfModule mod_ssl.c>
  ErrorLog /var/log/httpd/ssl_engine.log
  LogLevel debug
</IfModule>
# end mod ==>
```

Specify ServerName in SSL VirtualHost context

```
##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

# General setup for the virtual host, inherited from global configuration
#DocumentRoot "/var/www/html"
#ServerName www.example.com:443

# <==
# PERSON mod DD MM YYYY

ServerName bsp-dev.projectbamboo.org:443

# end mod of DD MM YYYY ==>
```

Specify certificate, private key, certificate chain, and Certificate Authority (CA) files and/or locations

Certificate, Certificate Key, and Certificate Chain files were purchased by Project Bamboo from GoDaddy.com (hence the *gd_bundle.crt* filename). A self-signed certificate may be used instead, provided clients and client applications do not require the assurance of server identity provided by a cert signed by a known Certificate Authority.

The *SSLCACertificatePath* directive is utilized (instead of the more-familiar *SSLCACertificateFile*) to more cleanly manage multiple trusted client certificates, as described below.

(Note that in-line comments included in the standard ssl.conf file are omitted here for brevity and clarity.)

```
# <==
# PERSON mod DD MM YYYY
SSLCertificateFile /etc/pki/tls/certs/WILDCARD.projectbamboo.org.crt
SSLCertificateKeyFile /etc/pki/tls/certs/WILDCARD.projectbamboo.org.key
SSLCertificateChainFile /etc/pki/tls/certs/gd_bundle.crt
SSLCACertificatePath /etc/pki/tls/certs/ca-certs-directory/
# end mod of DD MM YYYY ==>
```

Set Client Authentication (Type) directives

```
#SSLVerifyClient require
#SSLVerifyDepth  10
# <==
# PERSON mod DD MM YYYY
SSLVerifyClient optional
SSLVerifyDepth 1
SSLOptions +StdEnvVars +ExportCertData
# end mod of DD MM YYYY ==>
```

Set SSL Options directive

```
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire

# <==
# PERSON mod DD MM YYYY
#
SSLOptions +StdEnvVars +FakeBasicAuth +ExportCertData
# end mod of DD MM YYYY ==>
```

Set Proxy directives to forward service requests to BSP over AJP

These settings assume:

- FUSE ESB is listening on port localhost:8009
- an environment variable HOSTNAME_SERVICE_QUALIFIER is set to the value *services* (cf. *Developer Workbench Environment for BSP Service Developers*)

```
#
# <== PERSON mod DD MM YYYY
#
# add proxy pass directives to forward "/services" traffic to Bamboo Services Platform -- over AJP
# example of request URL: https://localhost/services/bsp/persons

   ProxyIOBufferSize 65536
   ProxyPass /services ajp://localhost:8009

# end mod DD MM YYYY ==>
```

Firewall and testing

- Ports 80 and 443 must be open on the server where httpd is running in order to enable access from another networked location.
- It is probably worthwhile to construct a simple test page to test that httpd is running
  - the test page should be located in the DocumentRoot directory specified in httpd.conf
  - if a different DocumentRoot is specified in the VirtualServer context of the SSL virtual host, a test page should be located there in order to test basic access to the server over https://

Certificate Exchange

Clients are required to supply an X.509 Certificate, which may be self-signed

To physically establish trust between the Bamboo Services Platform and a client application, X.509 certificates are exchanged. The client supplies a Bamboo Services Platform administrator with a PEM-formatted public X.509 certificate, while retaining control and confidentiality of the corresponding key.

Generate a self-signed key with openssl

A client may generate a self-signed X.509 certificate and key using openssl (or other software of your choice). The following console dialog may be illustrative for those who wish to use openssl.

Due to unresolved problems encountered with certificates in which an e-mail address was part of the Subject DN, it is recommended that *no e-mail address* be supplied in the console-dialog that occurs when generating a self-signed certificate. Cf. JIRA issue IAM-103, marked as "Won't Fix" as of 22 April 2013.

```
$ openssl req -x509 -nodes -days 3660 -newkey rsa:2048 -keyout my_selfsigned.key -out my_selfsigned.pem
Generating a 2048 bit RSA private key
..............................+++
..............+++
writing new private key to 'my_selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:Berkeley
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UC Berkeley
Organizational Unit Name (eg, section) []:Project Bamboo
Common Name (eg, YOUR name) []:Steve Masover
Email Address []:
$
$ ls
my_selfsigned.key       my_selfsigned.pem
$
```

You can view the newly created public cert using this openssl command:

```
openssl x509 -in my_selfsigned.pem -noout -text
```

Export self-signed key in PKCS12 format (e.g., for browser import)

If you wish to use a web browser as a trusted client to make calls to the Bamboo Services Platform, you will need to identify your browser with the key just generated above, to prove that the browser owns/controls the key corresponding to the public X.509 certificate. To do this with Firefox, for example, in order to use the Poster plugin to send requests to the BSP endpoint, you will need to export the key to PKCS12 format for import into Firefox. Here's how:

```
$ openssl pkcs12 -export -out my_selfsigned.pfx -inkey my_selfsigned.key -in my_selfsigned.pem
Enter Export Password:
Verifying - Enter Export Password:
$ ls -l
total 24
-rw-r--r--  1 masover  staff  1675 Mar 18 13:18 my_selfsigned.key
-rw-r--r--  1 masover  staff  1541 Mar 18 13:18 my_selfsigned.pem
-rw-r--r--  1 masover  staff  2701 Mar 18 13:20 my_selfsigned.pfx
$
```

A Bamboo Service Platform administrator places a trusted client's X.509 cert in the proper directory

Assumption: the trusted client's administrator has generated or otherwise-produced a confidential key and corresponding public X.509 certificate, the latter in PEM format; and has supplied a Bamboo Services Platform administrator with the public, PEM-formatted X.509 cert.

The directory in which Apache Web Server (httpd) expects to find a client's PEM-formatted cert is specified with the Apache directive *SSLCACertificatePath* . In the example configuration given above, the directive looks like this in the *ssl.conf* file:

```
SSLCACertificatePath /etc/pki/tls/certs/ca-certs-directory/
```

Having received the client administrator's PEM-formatted X.509 cert, the Bamboo Services Platform admin has a two step process to complete:

(1) copy PEM file to directory specified in SSLCACertificatePath

- copy the PEM file to the correct directory
- for a Unix/Linux machine or VM, permissions on the PEM ought to be *-rw-r--r--* (644)

(2) create a symbolic link to the PEM file as required by httpd

Certificates in this directory must have filenames (or, better yet, symbolic links pointing to original files w/ filenames) that are hashes of the cert contained in the file, with a suffix of <period><zero>. To derive the appropriate hash, use openssl. Then create a symbolic link. Example (console output redacted for brevity):

```
# openssl x509 -noout -hash -in my_selfsigned.pem
af21f59e
#
# ln -s my_selfsigned.pem af21f59e.0
# ls -l
# ls -l
lrwxrwxrwx 1 root root   15 Jan 14 16:14 af21f59e.0 -> my_selfsigned.pem
-rw-r--r-- 1 root root 1493 Mar 18 17:34 my_selfsigned.pem
#
```

For more detail, including circumstances in which a different numeral than zero is appended to the symbolic link or renamed cert, a description of configuring certs in this directory is given in the httpd-users mailing list archives of a message from Mark Montague of 20 July 2011: *Configuring SSLCACertificatePath in httpd-ssl.conf*.

A Bamboo administrator creates a corresponding record in a catalog of trusted client applications

Registration of known client applications requires some administrative prestidigitation in Grouper, explained in detail on the page *Maintaining Application Catalog Data for Trusted Clients*.

A client application administrator configures the client with BSP and Client X.509 Certs

Configuring Firefox Poster as a test client

Poster is a Firefox add-on "*that lets you make HTTP requests, set the entity body, and content type*." For convenience, the add-on was utilized heavily by developers of Bamboo services during the course of the Bamboo Technology Project.

Configure Firefox to identify web service requests with the trusted client's key

The following instructions are valid for Firefox v.19.0.2, and were tested on a Mac.

1. Open Firefox Preferences (*Tools* : *Options* on a Windows box; *Firefox* : *Preferences* on a Mac)
2. Choose the *Advanced* icon at the top of the Options / Preferences dialog
3. Choose the *Encryption* tab on the *Advanced* dialog.
4. Choose *View Certificates*. If you are using a Mac, choose *Your Certificates* within the Certificate Manager.
5. Choose *Import*
6. Navigate to the directory where you stored your PKCS12 formatted export of the self-signed key created above.
7. Chose the PKCS12-formatted file (e.g., *my_selfsigned.pfx*) to import.
8. Import, entering a password when asked if you used a password when creating the key (otherwise leave the password field empty).

Configure Firefox to trust the Bamboo Services Platform when establishing an SSL connection

This step is not necessary if the Bamboo Services Platform identifies itself with a certificate signed by a known certificate authority. This was the case in all testing performed by project teams in the course of the Bamboo technology Project.

However, configuration of a self-signed cert to be trusted by the browser and poster client should mimic the configuration process described above for the client-side cert. The differences are:

1. This is an external SERVER's cert that is being imported
2. The import is of the server's PUBLIC certificate, not a private key

Configuring a PHP client

This documentation reflects practical, early experience configuring a Drupal-hosted client module (cf. *Account Services UI - Bamboo IAM Client - Drupal Module PoC*). These are not general instructions for configuring clients, or even for configuring PHP clients. Inclusion of these notes is indented to be a pointer to an early example, not a general prescription.

The PHP Client URL Library (cURL) may be used to construct RESTful calls to BSP-hosted services.

Examples of code that constructs such calls can be found in the bamboo_as.module code for Bamboo's (partially-implemented) Account Services UI.

That module includes the code in *bamboo_as_rest.inc*, which includes execution of the *curl_setopt* function to assemble or point to the appropriate X.509 certificate .key and .pem files, and the file that includes the BSP's public X.509 cert, to be utilized in the request. This code is not well developed (e.g., it hard codes location of the relevant files), but it does represent a skeleton for how a better-developed codebase might be written to leverage the PHP cURL library.

A code snippet that sets key and cert values might look something like this:

```
curl_setopt($session, CURLOPT_SSLKEY, "/my/location/mykey.key");
curl_setopt($session, CURLOPT_SSLCERT, '/my/location/mypubliccert.pem');
curl_setopt($session, CURLOPT_CAINFO, '/my/location/ca-bundle.crt');
```