Infrastructure - Task #7822

Should we configure clients to use Expect 100-Continue header in 1.1 requests

2016-06-02 04:07 - Rob Nahf

Status:	New	**Start date:**	2016-06-02
Priority:	Normal	**Due date:**	
Assignee:	Rob Nahf	**% Done:**	0%
Category:	d1_libclient_java	**Estimated time:**	0.00 hour
Target version:	CCI-2.4.0		
Milestone:	None	**Story Points:**	
Product Version:	*		

Description

When the apache directive SSLVerifyClient is set within a or directive, then the ssl module will attempt to renegotiate the connection.

> Apache HTTP Server will request SSL renegotiation any time an SSL
> session is already established but a request is made for a per-location
> context which requires different security -- for example, if you have
> the SSLVerifyClient directive in a Directory or Location block.
>
> http://mail-archives.apache.org/mod_mbox/httpd-users/201210.mbox/%3C50741683.6050306%40catseye.org%3E

During the renegotiation, the apache webserver will buffer any body content submitted with the request.

> If an SSL renegotiation is required in per-location context, for example,
> any use of SSLVerifyClient in a Directory or Location block, then mod_ssl
> must buffer any HTTP request body into memory until the new SSL handshake
> can be performed. This directive can be used to set the amount of memory
> that will be used for this buffer.
>
> Note that in many configurations, the client sending the request body will
> be untrusted so a denial of service attack by consumption of memory must be
> considered when changing this configuration setting.
>
> http://httpd.apache.org/docs/2.2/mod/mod_ssl.html#sslrenegbuffersize

If the size of the request body buffered exceeds the size of the buffer, then an apache error is thrown as noted by Chris J and verified by Robert W.

One solution is to not add SSLVerifyClient directives in a Location or Directory block.

A work around to the problem is to increase the size of SSL renegotiation buffer using the directive:

SSLRenegBufferSize

On the apache webserver itself, but that may cause problems as noted in the documentation. It is also likely that we may not wish to limit the maximum size of any content submitted by a user.

Another solution is to use the Expect: 100-Continue header in the request object, such that the request body is sent after the renegotiation has occurred.

> The purpose of the Expect: 100-Continue handshake is to allow the client
> that is sending a request message with a request body to determine if
> the origin server is willing to accept the request (based on the request

headers) before the client sends the request body. The use of the Expect: 100-continue handshake can result in a noticeable performance improvement for entity enclosing requests (such as POST and PUT) that require the target server's authentication. The Expect: 100-continue handshake should be used with caution, as it may cause problems with HTTP servers and proxies that do not support HTTP/1.1 protocol.

https://hc.apache.org/httpcomponents-client-4.2.x/tutorial/html/fundamentals.html

While doing that, we need to make sure that use of the Expect: 100-Continue header works for:

http 1.0 servers

non-compliant HTTP 1.1 servers

compliant HTTP 1.1 servers

Apache HTTPClient has a default timeout of 3 seconds on waiting for the 100-continue interim response, after which it sends the message body. This should cover communication with HTTP 1.0 servers (who don't send interim statuses). However, This timeout system might breakdown if an HTTP 1.1 compliant server is swamped and just plain slow to respond. That's to ask, under heavy load could the server buffer still overflow?

If a server returns HTTP Error 417 (Expectation failed), the the request should be retried without the header.

https://support.urbanairship.com/entries/59909909--Expect-100-Continue-Issues-and-Risks

See

https://www.w3.org/Protocols/rfc2616/rfc2616-sec8.html

https://issues.apache.org/jira/browse/HTTPCLIENT-889

Related issues:

Related to Python Libraries - Task #7821: Verify Expect: 100-Continue header ...	**New**	**2016-05-31**

History

#1 - 2016-06-03 19:46 - Robert Waltz

- Subject changed from configure clients to use Expect 100-Continue header in 1.1 requests to Should we configure clients to use Expect 100-Continue header in 1.1 requests

- Description updated

- Parent task deleted (#6539)

- Product Version deleted (*)

#2 - 2016-06-03 19:49 - Robert Waltz

- Related to Task #7821: Verify Expect: 100-Continue header on POST or PUT requests added