Infrastructure - Bug #2411

knb MNs and CNs allow self-signed certificates to connect

2012-02-24 21:53 - Rob Nahf

Status:	In Progress	**Start date:**	2014-10-01
Priority:	Normal	**Due date:**	2014-10-01
Assignee:	Chris Jones	**% Done:**	30%
Category:	Environment.Development	**Estimated time:**	0.00 hour
Target version:	Release Backlog		
Milestone:	CCI-1.0.0	**Story Points:**	
Product Version:			

Description

I have a self-signed certificate that succeeds in mn.ping(), mn.listObjects() and mn.isAuthorized() against demo1.test.dataone.org.

However:
a) an expired certificate for the same user (signed by dataone CA) fails to establish a connection, and
b) my MobileMe certificate passed by the Safari browser fails to establish a connection

poses a security risk to the system.

also tests the same way for cn-dev, so maybe a problem with CN deployment packaging as well.

Subtasks:

Task # 2478: testConnectionLayer_SelfSignedCert() failing	**Rejected**

Related issues:

Related to Infrastructure - Story #2548: recasting untrusted certs to public ...	**New**	**2012-03-27**
Related to Java Client - Story #6570: libclient should give better indication...	**Closed**	

History

#1 - 2012-02-24 21:57 - Rob Nahf

- Subject changed from knb MNs (at least) allow self-signed certificates to connect to knb MNs and CNs allow self-signed certificates to connect

also a problem for cn-dev.

#2 - 2012-03-07 15:39 - Chris Jones

- Status changed from New to In Progress

- Target version set to Sprint-2012.09-Block.2.1

The SSL configuration for Apache has:

SSLVerifyClient optional

meaning that a client may or may not send a valid cert. If we set it to 'none', the behavior is the same. If we set it to 'require', no SSL connections will be allowed without a valid cert (limited to cilogon and dataone on the CNs, and any on the MN depending how we decide to limit the authorized CAs.

This seems black and white: either we force or don't force valid client certs - we can't have an in between that I can see.

#3 - 2012-03-10 00:32 - Ben Leinfelder

We have to keep it as optional or else there will not be any notion of public access -- all communication with the server will require a valid certificate. My understanding of client certificate processing was that only client certificates signed by a CA that the server trusted would be accepted and any

others would be ignored (treated as though there were no certificate at all). This makes it a little more difficult to immediately recognize why you don't have access to some protected resource, but I think it's not that bad in the grand scheme of things.
Just because you are "using a self signed cert" doesn't mean the server used it or even asked for it. At least that's what I was seeing back in the day when we first started down this client certificate road.

#4 - 2012-03-16 02:34 - Dave Vieglais

- Position set to 15

- Target version changed from Sprint-2012.09-Block.2.1 to Sprint-2012.11-Block.2.2

#5 - 2012-04-25 14:13 - Dave Vieglais

- Target version deleted (Sprint-2012.11-Block.2.2)

The issue is related to he SSL handshake process. The server indicates which CAs it will trust, if the (java) client does not have a certificate that matches, then it wil continue the connection with no certificate, and so will be connecting as public. This is not a security risk (since only public content will be seen), however there should be some feedback to the user that the connection is as a public user.

It is necessary for the certificate acceptance to be optional since otherwise, as Ben indicates, all connections will require a certificate which makes all interactions significantly more complex.

Moving this issue to the backlog, since notifying the user of who they are connected as is important, but not critical for the public release.

#6 - 2014-10-01 22:22 - Dave Vieglais

- Start date set to 2014-10-01

- Due date set to 2014-10-01

- Target version set to Release Backlog