strongSwan - Bug #453

constraints_validator's check_policy is too strict

21.11.2013 09:31 - Christian Zwettler

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Martin Willi	**Estimated time:**	0.00 hour
Category:	libstrongswan		
Target version:	5.2.2		
Affected version:	5.1.1	**Resolution:**	Fixed

Description

Hi,

we are using strongSwan and since we enabled the constraints plugin we get the error policy x.x.xxx.x.xx.x.xxx missing in issuing certificate 'C=xxx, O=xxx, OU=xxx, CN=xxx'
I tried to replicate the error using openSSL with the optione -policy_check but that works just fine.

So my question is, how can i replicate the issue to reproduce the error?

Related issues:

Related to Issue #1237: CRL not validating correctly	**Closed**	**14.12.2015**
Has duplicate Issue #489: constraints_validator's check_policy is too strict	**Closed**	**16.01.2014**

Associated revisions

Revision 8131d180 - 30.10.2014 11:42 - Martin Willi

Merge branch 'policy-constraints'

Fixes handling of invalid policies in end entity certificates by not rejecting
the full certificate, but just invalidating the affected policy. Additionally
adds a bunch of unit tests for the constraints plugin, and some minor fixes
to the nameConstraints handling.

Currently we still reject CAs that use invalid policy mapping; we should accept
such certificates and just invalid affected policies in a next iteration.

Fixes #453.

History

#1 - 21.11.2013 10:18 - Martin Willi

- Category set to libstrongswan

- Status changed from New to Assigned

- Assignee set to Martin Willi

- Affected version changed from 5.0.4 to 5.1.1

Hi Christian

For a certificate policy to be valid, it must be contained in the issuing certificate (or the issuer certificate must have anyPolicy or an appropriate policy mapping).

Some CAs issue certificates with policies not confirmed by the CA. According to RFC 5280, all policies must be explicitly allowed by the CA:

> In an end entity certificate, these policy information terms indicate the policy under which the certificate has been issued and the purposes for which the certificate may be used. In a CA certificate, these policy information terms limit the set of policies for certification paths that include this certificate. When a CA does not wish to limit the set of policies for certification paths that include this certificate, it MAY assert the special policy anyPolicy, with a value of { 2 5 29 32 0 }.

So if I read RFC 5280 correctly, the certificate policy won't be valid in such a case. But feel free to correct me if you see this differently.

The actual problem now is that the constraints plugin is a little too strict. While checking trust chains, it should just invalidate that policy. Instead, it currently rejects the certificate at all, making it unusable, whether the policy is required or not. This is certainly wrong.

Fixing this issue is on my TODO list, but it's not a top priority item.

If you need the constraints plugin, you may use a different certificate fulfilling these requirements. If you don't need constraints checking, just disable it until the issue is fixed.

Regards
Martin

#2 - 21.11.2013 11:28 - Christian Zwettler

Hi Martin,

thank you very much for your fast reply.

My conclusion was the same as yours, including the part about it being too strict. I just wanted to make sure that that is actually the case.

I still wonder if I can reproduce the error using other tools (e.g. openSSL), so far I did not succeed.

Regards,
Christian

#3 - 07.10.2014 11:39 - D B

Just ran into the same issue. A fix would be welcome!

#4 - 30.10.2014 11:49 - Martin Willi

- Tracker changed from Issue to Bug

- Status changed from Assigned to Closed

- Target version set to 5.2.2

- Resolution set to Fixed

Finally fixed with the referenced merge commit.

#5 - 05.01.2015 10:18 - Tobias Brunner

- Subject changed from Using Constraints Plugin to constraints_validator's check_policy is too strict

#6 - 17.12.2015 09:48 - Tobias Brunner

- Related to Issue #1237: CRL not validating correctly added