strongSwan - Bug #3357

Avoid log message if mixed PKIs (RSA/ECDSA) are used with the openssl plugin

04.03.2020 06:01 - Glen Huang

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libstrongswan		
Target version:	5.8.3		
Affected version:	5.8.2	**Resolution:**	Fixed

Description

Currently, you if request a ECDSA certificate from Let's Encrypt, the certificate signature scheme is RSA_EMSA_PKCS1_SHA2_256, which is only supported if the certificate is RSA if I'm not wrong, and charon keeps complaining that signature scheme is not supported in EC.

Any chance this combination of certificate type and signature can be supported?

Associated revisions

Revision 61769fd1 - 06.03.2020 11:12 - Tobias Brunner

openssl: Don't check signature if issuer doesn't match always

Doing this for the self-signed check also (i.e. if this and issuer are the same) is particularly useful if the issuer uses a different key type. Otherwise, we'd try to verify the signature with an incompatible key that would result in a log message.

Fixes #3357.

History

#1 - 04.03.2020 10:37 - Tobias Brunner

- Status changed from New to Feedback

> Currently, you if request a ECDSA certificate from Let's Encrypt, the certificate signature scheme is RSA_EMSA_PKCS1_SHA2_256

What are you talking about? Why would that be the case? Did you explicitly configure a signature scheme in your config?

#2 - 04.03.2020 10:54 - Glen Huang

Tobias Brunner wrote:

> > Currently, you if request a ECDSA certificate from Let's Encrypt, the certificate signature scheme is RSA_EMSA_PKCS1_SHA2_256
>
> What are you talking about? Why would that be the case? Did you explicitly configure a signature scheme in your config?

I didn't do anything special, the certificate is issued by LE and they don't offer configs. They haven't achieved full ECDSA yet, so that the intermedia cert and root ca are RSA based, only the end cert is ECDSA, but the signature is RSA + SHA 256.

I found out if I install both x509 and pkcs1 plugins, the error message goes away. But not sure if it really solves the problem.

#3 - 04.03.2020 11:49 - Tobias Brunner

> I found out if I install both x509 and pkcs1 plugins, the error message goes away. But not sure if it really solves the problem.

Which plugins did you use before? The *pkcs1* plugin is definitely required by the x509 plugin.

#4 - 04.03.2020 11:54 - Glen Huang

Tobias Brunner wrote:

> I found out if I install both x509 and pkcs1 plugins, the error message goes away. But not sure if it really solves the problem.

Which plugins did you use before? The *pkcs1* plugin is definitely required by the x509 plugin.

With these compile options, the error will display:

```
--with-capabilities=native
--disable-defaults
--disable-shared
--enable-static
--enable-monolithic
--enable-nonce
--enable-openssl
--enable-pem
--enable-vici
--enable-charon
--enable-swanctl
--enable-ikev2
--enable-kernel-netlink
--enable-socket-default
```

Adding these make it disappear:

```
--enable-pkcs1
--enable-x509
```

But from the code it seems to hide the error, the certificate is longer verified as a EC one.

#5 - 04.03.2020 13:05 - Tobias Brunner

> With these compile options, the error will display:
> [...]
>
> Adding these make it disappear:
> [...]

Does it actually fail? Or are you just worried about the log message?

#6 - 05.03.2020 09:20 - Glen Huang

It doesn't fail. I just thought the info might help you identify an issue or something.

But if it's normal, feel free to close the issue.

#7 - 05.03.2020 12:53 - Tobias Brunner

- Tracker changed from Issue to Bug

- Category set to libstrongswan

- Assignee set to Tobias Brunner

- Target version set to 5.8.3

> It doesn't fail. I just thought the info might help you identify an issue or something.

It's not an issue, but I found the reason for it. When a certificate is loaded, there is a check whether it is self-signed or not. In the *openssl* plugin, as compared to the *x509* plugin, there was no check whether the checked certificate's issuer and the issuer certificate's subject match before verifying the signature. Which means that in such mixed PKIs an ECC public key (that of the end-entity certificate) could be used to verify an RSA signature (generated by the issuing CA), which will fail with the message you saw. I pushed a fix for that to the *3357-openssl-issued-by* branch.

#8 - 05.03.2020 13:02 - Glen Huang

Tobias Brunner wrote:

> It doesn't fail. I just thought the info might help you identify an issue or something.

It's not an issue, but I found the reason for it. When a certificate is loaded, there is a check whether it is self-signed or not. In the *openssl* plugin, as compared to the *x509* plugin, there was no check whether the checked certificate's issuer and the issuer certificate's subject match before verifying the signature. Which means that in such mixed PKIs an ECC public key (that of the end-entity certificate) could be used to verify an RSA signature (generated by the issuing CA), which will fail with the message you saw. I pushed a fix for that to the *3357-openssl-issued-by* branch.

Thanks for the quick reply and detailed explanation!

So if I don't use CRLs or OCSP, I guess I can safely remove the x509 and pkcs1 plugins?

Also why x509 definitely depends on pkcs1? If I'm not wrong, pkcs1 basically means RSA support? If I only use ECDSA certificates, will it still be needed?

#9 - 05.03.2020 13:11 - Tobias Brunner

> So if I don't use CRLs or OCSP, I guess I can safely remove the x509 and pkcs1 plugins?

Sure.

> Also why x509 definitely depends on pkcs1? If I'm not wrong, pkcs1 basically means RSA support?

No, it parses other public (subjectPublicKeyInfo) and private keys too (only the basic structure, it doesn't do any crypto).

> If I only use ECDSA certificates, will it still be needed?

I guess it depends on the loaded plugins and the format of the private keys (with *openssl* loaded it should work fine without the plugin).

#10 - 05.03.2020 13:28 - Glen Huang

Tobias Brunner wrote:

> > So if I don't use CRLs or OCSP, I guess I can safely remove the x509 and pkcs1 plugins?
>
> Sure.
>
> > Also why x509 definitely depends on pkcs1? If I'm not wrong, pkcs1 basically means RSA support?
>
> No, it parses other public (subjectPublicKeyInfo) and private keys too (only the basic structure, it doesn't do any crypto).
>
> > If I only use ECDSA certificates, will it still be needed?
>
> I guess it depends on the loaded plugins and the format of the private keys (with *openssl* loaded it should work fine without the plugin).

Good to know. Looking forward to version 5.8.3

#11 - 06.03.2020 11:13 - Tobias Brunner

- Subject changed from Support RSA SHA 256 signature for ECDSA certificate? to Avoid log message if mixed PKIs (RSA/ECDSA) are used with the openssl plugin

- Status changed from Feedback to Closed

- Resolution set to Fixed