strongSwan - Bug #854

libipsec support for NULL encryption?

20.02.2015 12:13 - Peter Whisker

Status:	Closed	**Start date:**	20.02.2015
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libipsec		
Target version:	5.3.0		
Affected version:	5.2.1	**Resolution:**	Fixed

Description

Hi

We have a requirement for NULL encryption (eg esp=null-sha256-modp4096) - the application needs to ensure integrity but also requires that the content is in the clear. I also need to use libipsec for other reasons relating to kernel separation. I have no problem with the above esp string in standard kernel mode but I get lots of errors if I use it in libipsec mode. Removing the "esp=null-sha256-modp4096" line from ipsec.conf makes libipsec behave but it defaults to AES.

Is it still the case (as per issue 377) that AES and AES-GCM are the **only** crypto algorithms supported and even NULL encryption does not work with libipsec?

Thanks
Peter

Associated revisions

Revision 4e236a7e - 23.02.2015 11:29 - Tobias Brunner

openssl: Return the proper IV length for OpenSSL crypters

For instance, the NULL cipher has a block size of 1 but an IV length
of 0.

Fixes #854.

History

#1 - 20.02.2015 12:44 - Peter Whisker

BTW I see the following errors:

Feb 20 11:41:06 IrisP-L-1 charon: 05[ESP] unsupported IP version
Feb 20 11:41:06 IrisP-L-1 charon: 05[ESP] parsing ESP payload failed: unsupported payload

#2 - 20.02.2015 13:37 - Tobias Brunner

- File 0001-openssl-Return-the-proper-IV-length-for-OpenSSL-cryp.patch added

- Tracker changed from Issue to Bug

- Status changed from New to Feedback

- Target version set to 5.3.0

> Is it still the case (as per issue 377) that AES and AES-GCM are the **only** crypto algorithms supported and even NULL encryption does not work with libipsec?

As you can see in #377 the limitation to AES/AES-GCM has been lifted with 5.1.1. Since then *libipsec* supports all the algorithms provided by *libstrongswan*. And there is a NULL cipher provided by the *openssl* plugin, otherwise the CHILD_SA negotiation would fail in the first place.

```
Feb 20 11:41:06 IrisP-L-1 charon: 05[ESP] unsupported IP version
Feb 20 11:41:06 IrisP-L-1 charon: 05[ESP] parsing ESP payload failed: unsupported payload
```

The problem is that the *crypter_t* implementation of the *openssl* plugin always returns the cipher's block size as IV length. While the block size for the

NULL cipher is 1 the IV size should be 0. So connections between two instances using *libipsec* work because they both will send/expect a 1 byte IV, but if one of the peers is implementation that does handle this correctly you'll see the messages above because the ESP packets are parsed with a 1 byte offset (similarly are the sent packets invalid).

While we added more flexibility in regards to the IV length with f7c04c5b37 (before that we always queried the block size externally), using *block_size* here (source:src/libstrongswan/plugins/openssl/openssl_crypter.c#L135) is incorrect. Actually, the EVP_CIPHER struct has an *iv_len* member for exactly this purpose. The attached patch fixes the problem.

#3 - 20.02.2015 15:25 - Peter Whisker

Wow! What a quick response. I have patched the code and rebuilt and it works great! Thank you so much!

Peter

#4 - 23.02.2015 11:31 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed

Great, thanks for testing.

Files

0001-openssl-Return-the-proper-IV-length-for-OpenSSL-cryp.patch	962 Bytes	20.02.2015	Tobias Brunner