strongSwan - Bug #747

5.2.1: test suite failure in 'test_chunk_internet_checksum' on big-endian hosts

22.10.2014 08:41 - Anonymous

Status:	Closed	**Start date:**	22.10.2014
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libstrongswan		
Target version:	5.2.2		
Affected version:	5.2.1	**Resolution:**	Fixed

Description

Hi,

I updated the Debian packages of strongswan to version 5.2.1, and the test suite fails on big-endian architectures in the new 'test_chunk_internet_checksum' test, with the following:

```
Running case 'chunk_internet_checksum': -
      Failure in 'test_chunk_internet_checksum': 0x442e != ntohs(sum) (17454 != 17455) (suites/tes
t_chunk.c:799, i = 0)
```

Full build logs are here:
- powerpc: https://buildd.debian.org/status/fetch.php?pkg=strongswan&arch=powerpc&ver=5.2.1-1&stamp=1413927358
- s390x: https://buildd.debian.org/status/fetch.php?pkg=strongswan&arch=s390x&ver=5.2.1-1&stamp=1413929351

Thanks.

Associated revisions

Revision eb218ff8 - 23.10.2014 11:13 - Tobias Brunner

chunk: Fix internet checksum calculation on big-endian systems

ntohs() might be defined as noop (#define ntohs(x) (x)) so we have
to manually shorten the negated value (gets promoted to an int).

Fixes #747.

Revision e7c582e6 - 23.10.2014 11:13 - Tobias Brunner

unit-tests: Fix internet checksum tests on big-endian systems

We actually need to do a byte-swap, which ntohs() only does on
little-endian systems.

Fixes #747.

History

#1 - 22.10.2014 19:46 - Tobias Brunner

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

```
    Running case 'chunk_internet_checksum': -
    Failure in 'test_chunk_internet_checksum': 0x442e != ntohs(sum) (17454 != 17455) (suites/test_chunk.c:799, i = 0)
```

I ran the test suite in QEMU (mips) and got the same result (17455 instead of 17454). I tried to replicate the problem in a simple test program, but wasn't able to. That is, until I included arpa/inet.h (ntohs() might be a compiler built-in as it worked fine without that include).

Further analysis showed that the problem was caused by the definition of ntohs() and friends in netinet/in.h. On big-endian systems they are defined as noops (e.g. #define ntohs(x) (x)), so there is no function call (or cast) involved that would shorten the argument (or result) of ntohs() to 16-bit. On the first line here

```
u_int16_t chunk_internet_checksum_inc(chunk_t data, u_int16_t checksum)
{
    u_int32_t sum = ntohs(~checksum);
```

checksum is promoted to 32-bit when it is negated, so with an initial value of 0xffff and without shortening to 16-bit we end up with sum = 0xffff0000. If ntohs() was a function call or if there was a cast to 16-bit everything would be fine, but since that doesn't happen the calculation later is incorrect.

There was also a bug in the unit test on big-endian systems. I pushed fixes to the *inet-checksum* branch.

#2 - 23.10.2014 07:51 - Anonymous

Thanks. I included both commits in the 5.2.1-2 version of the package, and the libstrongswan test suite now passes.

#3 - 23.10.2014 11:28 - Tobias Brunner

- Tracker changed from Issue to Bug

- Status changed from Feedback to Closed

- Target version set to 5.2.2

- Resolution set to Fixed