strongSwan - Bug #2843

scepclient.c:1120: use of out of scope variable ?

30.11.2018 09:40 - David Binderman

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	scepclient		
Target version:	5.7.2		
Affected version:	5.7.1	**Resolution:**	Fixed

Description

[scepclient.c:1117] -> [scepclient.c:1108] -> [scepclient.c:1120]: (error) Using pointer to local variable 'buf' that is out of scope.

Source code is

```
if (distinguishedName == NULL)
{
    char buf[BUF_LEN];
    int n = sprintf(buf, DEFAULT_DN);

    /* set the common name to the hostname */
    if (gethostname(buf + n, BUF_LEN - n) || strlen(buf) == n)
    {
        exit_scepclient("no hostname defined, use "
                        "--dn <distinguished name> option");
    }
    distinguishedName = buf;
}

DBG2(DBG_APP, "dn: '%s'", distinguishedName);
subject = identification_create_from_string(distinguishedName);
```

Associated revisions

Revision 631abb68 - 03.12.2018 11:54 - Tobias Brunner

scepclient: Don't use a block-scope buffer for the default DN

The correct behavior will depend on the compiler.

Fixes #2843.

History

#1 - 30.11.2018 10:48 - Tobias Brunner

- Tracker changed from Issue to Bug

- Description updated

- Category set to scepclient

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

- Target version set to 5.7.2

Yes, that's not ideal. I suppose it depends on the compiler (and perhaps the flags) whether it's actually a problem or not. I pushed a fix to the *2843-scepclient-buf* branch. The code generated by GCC 7.3.0 with -O2 is exactly the same with or without the fix, though.

```
[scepclient.c:1117] -> [scepclient.c:1108] -> [scepclient.c:1120]: (error) Using pointer to local variable 'buf' that is out of scope.
```

What tool/compiler reported this? Because none of GCC 7.3.0, Clang 6.0.0 (both with -Wextra), Coverity or SonarCube reported this (the latter two operate on binary output so that might be related to the compiler).

#2 - 30.11.2018 11:11 - David Binderman

> The code generated by GCC 7.3.0 with -O2 is exactly the same with or without the fix,

Code that depends on certain version numbers of certain compilers is quite fragile compared to code that conforms with defined language standards and so is expected to work everywhere.

> What tool/compiler reported this?

Latest development version of cppcheck, a static analyser for C/C++.

#3 - 03.12.2018 12:04 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed