DCMTK - Bug #581

Security escalation problem in dcmnet

2013-11-25 09:32 - Marco Eichelberg

Status:	Closed	**Start date:**	2013-11-25
Priority:	High	**Due date:**	
Assignee:		**% Done:**	100%
Category:	Library and Apps	**Estimated time:**	0:00 hour
Target version:	3.6.1		
Module:	dcmnet	**Compiler:**	
Operating System:			

Description

> While I was checking part of dcmtk source code I fond a> security violation.
>
> The problem is that the 'setuid()' call is not checked in
> dcmtk-3.6.0/dcmwlm/libsrc/wlmactmg.cc line 254,
> dcmtk-3.6.0/dcmqrdb/apps/dcmqrscp.cc line 677,
> dcmtk-3.6.0/dcmnet/libsrc/scp.cc line 249,
> etc,.
>
> if setuid() fails for any reason, for instance if an environment limits
> the number of processes a user can have and the target uid already is at
> the limit, the following will be run as root.
>
> I CVE is comming ... but I would like coordinate the
> disclosure with you.
>
> Are you agree about this is a "security issue" ?
> I am writing a small report/note about this.
> Since you know better than me your software, please could you
> tell me which are the security implications in this case ?
> I just want to describe in a sentence the security risk of this bug.
> The problem is basically that the inn application assume that is running
> as nobody etc., but is no true for the case that I described above.

Thank you for this bug report. I agree with your analysis.
setuid() can fail with either EPERM or EAGAIN as errno error code.
EPERM is "harmless" as this will only happen when the application
is already run by a user with limited privileges.

EAGAIN, to my understanding, can only occur in the situation you
describe: If a system has been configured to only run a certain number
of processes for the given uid, and this number of processes is already
running, then the application, when run with setuid root, will fail to
give up the root privileges and continue to run as root.

This only affects systems that a) run the tools as setuid root
(which is only needed if you want to operate on a TCP port < 1024),
and that are configured to limit the number of processes of the
non-privileged user ("nobody") the tools are using. A simple stop-gap
measure is, therefore, to run the tools using a non-privileged port
(such as the official TCP port for DICOM, 11112) and revoke the setuid
bit of the binary, which is not needed in that case.

Basically, the tools using this code should check for EAGAIN, and
if present, terminate the application. Affected applications are
wlmscpfs and dcmqrscp (I don't think dcmnet/libsrc/scp.cc is used by
any of the applications that are part of the public DCMTK yet).

History

#1 - 2013-11-26 10:45 - Jörg Riesmeier

> (I don't think dcmnet/libsrc/scp.cc is used by any of the applications that are part of the public DCMTK yet).

Actually, DcmStorageSCP is used by dcmrecv.

#2 - 2014-02-19 18:58 - Jörg Riesmeier

- Status changed from New to Closed

- % Done changed from 0 to 100

Should be fixed now: http://git.dcmtk.org/web?p=dcmtk.git;a=commit;h=beaf5a5c24101daeeafa48c375120b16197c9e95