

## Re: How secure is software X?

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2006-05/msg00291.html>

---

- *From:* [Matt.Carpenter@xxxxxxxxxxx](mailto:Matt.Carpenter@xxxxxxxxxxx)
  - *Date:* Mon, 15 May 2006 09:44:35 -0400
- 

Fabian Becker <neonomicus@xxxxxx> wrote on 05/12/2006 03:12:32 PM:

Dear David

in my opinion a software can either be secure or not secure.

I think it's a bit like a woman cannot be "a bit pregnant".

But the protocol you are talking about can be used to tell the secure from the insecure pieces of software. By applying a test for these rules against systems, security will definitely be enhanced since software brandmarked with "insecure" will simply loose it's value.

Another question is how to verify that authors check their own software?

If they do not do it by now, why then? The only reason I could imagine would be a raise in value by beeing able to say "My software is a tested 'secure' one".

Hello Fabian,

Respectfully, to classify security like that would be to condemn every software as "insecure". What I see David proposing is more akin to "how far along in her pregnancy". It is a measurement. Hopefully we can all agree that with large applications (eg. Oracle, WebSphere, Windows, etc...) there are bugs. While the desired direction may be 100% security (much like the desired personal goal is perfection), we need to be able to qualify how difficult it is to break applications in a standardized fashion.

The one caveat I might bring up is the topic of false security.

It is difficult to prove, in a standardized methodology, that an application is difficult to break; only that our methodology has failed to do so. How in-depth a fuzzing to we apply for this standard? Does the standard include significant levels of reverse engineering? If so, who does this (since some are more proficient than others)? If not, what true value does this standard prove, except that the application can withstand yet another script?

In concept, I agree wholeheartedly that a security qualification could be beneficial. And perhaps, with all the brainpower involved, an relatively reliable automated method could be achieved. There are many details which would need to be sorted out. Some applications are more easily fuzzed

## Re: How secure is software X?

than others... For example, SMTP servers have a pretty standard interface, they have to. Database servers do not, although they do have underlying language similarities. Web app servers, such as WebSphere and Oracle app server, may have commonalities, but have such a breadth of testing required to give any comprehensive qualification, to do so seems rather overwhelming.

In my own little portable mind, such a standard would require an infrastructure of standards, with each "class" of application being represented and handled separately.

One alternative proposition would be to provide a difficulty rating for the security researchers to apply to their vulnerability reports/analysis. Simply an appendage to our normal bugtraq traffic. Let the researchers grade the difficulty. Perhaps this would be problematic as well, since it would take me far longer to find a vuln in Oracle than it would for someone like David. But it would be a start.

\$0x02