

Re: What's so funny? WAS Re: rotor replacement

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-01/5167.html>

From: Paul Rubin (//phr.cx_at_NOSPAM.invalid)

Date: 01/31/05

Date: 30 Jan 2005 19:42:55 -0800

Skip Montanaro <skip@pobox.com> writes:

- > *While it might be convenient to not have to distribute some third*
- > *party library in addition to Python, there is a fundamental problem*
- > *implementing a crypto algorithm from scratch for inclusion into*
- > *Python. There is always the problem that the new code has to be*
- > *more rigorously tested than typical code*

Actually and surprisingly, that's not really true. Crypto algorithms are pretty straightforward, so if you examine the code and check that it passes a bunch of test vectors, you can be pretty sure it's correct. It's much harder to check something like a compiler, which has a much bigger state space, far more decision points, etc. The usual bugs in crypto apps (and there are lots of such bugs) are in how the primitives are used, not in the primitives themselves.

- > *and new bugs means a new distribution of Python, not just a*
- > *replacement library.*

Why would that be true of a crypto module and not true of, say, the socket module? If the socket module has a bug that allows a remote takeover of the application, that's as bad as a crypto bug.

- > *A bug in code that is not security-related generally means something*
- > *doesn't work and only rarely means a security hole has been opened*
- > *on the computer. A bug in security-related code more often means*
- > *the latter as well.*

People often don't understand that that almost all code is security-related. Any code that touches data that came from the internet is security related. If the math library arctangent function has a buffer overflow bug triggered by a certain input number, and someone uses math.arctan in an image viewing program, then maybe a specially concocted image designed to set off the bug can take over the user's computer. So even something like math.arctan is security related.

- > *While I imagine the changes were fairly small, the guys involved are*
- > *all very smart, and the code is fairly straightforward (little, if*

comp.lang.python: Re: What's so funny? WAS Re: rotor replacement

- > *any, memory allocation going on), there is still the possibility*
- > *that a bug lurks in either the incorporated code or in the changes*
- > *to it. How quickly could the Python community respond if a bug was*
- > *found and fixed in the public domain SHA code? How much harder*
- > *would it be for people to adapt if they had to reinstall Python*
- > *instead of just an external library?*

If they're able to install external libraries, what stops them from reinstalling a patched sha module?

The hazards of using a crypto module are sort of like the hazards of using the threading module. Unless you know what you're doing and are very careful, it's easy to make an error. But the resulting bugs happen because the module did exactly what you asked for, not because it did something different from what you asked for.

If you ever write code that uses the Internet, I highly recommend the book "Security Engineering", by Ross Anderson. It will give you some idea of what you are up against.