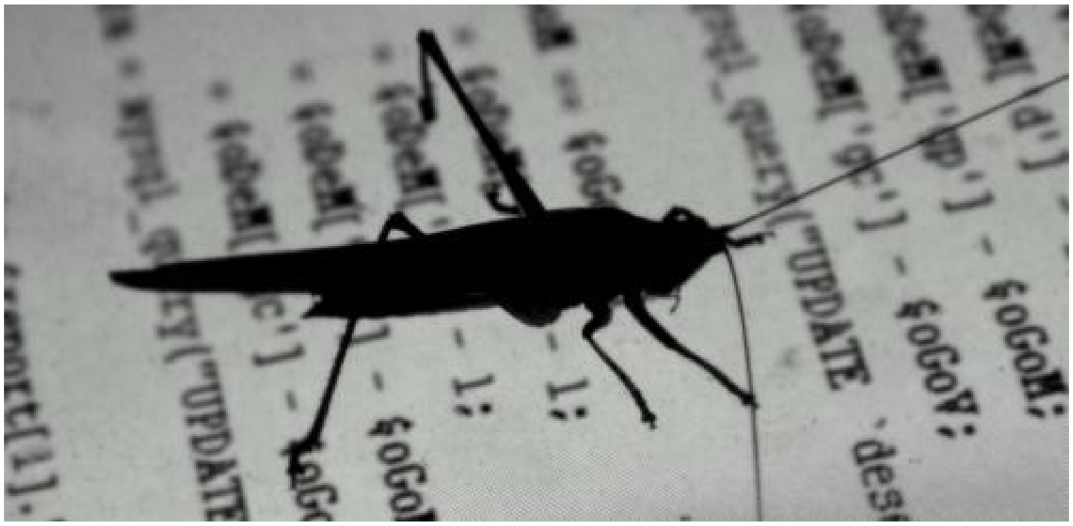
Six more bugs found in popular OpenSSL security tool

June 6 2014, by Robert Merkel



New bugs in the code for OpenSSL. Credit: Flickr/Guilherme Tavares, CC BY-NC

Computer system administrators around the world are groaning again as six new security problems have been found in the OpenSSL security library.

OpenSSL is a security tool that provides facilities to other computer programs to communicate securely over the public internet.

For example, if you see "https://" at the start of a web address rather than "http:", the "s" part indicates that the connection is secure. More often than not, at least the server computer on the other end of the connection will be using OpenSSL to provide that security.

OpenSSL provides two main forms of security:

1. it scrambles information so it is unreadable to anyone other than the intended recipient
2. it authenticates the source of information, ensuring the sender is who they say they are.

OpenSSL is also used in some common consumer applications, such as software in Google's Android smartphones.

So when the Heartbleed vulnerability in OpenSSL was discovered and widely publicised in April this year, system administrators had to rush to update their systems to protect against it.

The latest bugs

The OpenSSL developers, a loosely-connected group of volunteers who primarily collaborate online, announced this week an updated version of their tool with fixes to the six new vulnerabilities, each reported independently by security researchers around the world.

Of the six, four appear to only be exploitable for "denial-of-service" purposes. An attacker could cause a server running the vulnerable software to stop functioning.

But the other two bugs are more serious.

The first (explained in technical detail here) might, theoretically, allow an attacker to gain full control of a vulnerable server. At that point all data on that server becomes available to the attacker.

But this can only occur if a particular facility called Datagram Transport Layer Security (DTLS) of OpenSSL is in use. Fortunately this facility isn't used by the vast majority of applications using OpenSSL.

The second serious bug (technical explanation here and additional technical analysis here), has been present in OpenSSL for at least 15 years.

In essence, by sending certain messages through OpenSSL in the wrong order, a supposedly "secure" connection can be initiated with a known password. This can be used to establish a "man-in-the-middle" attack, where an attacker with access to the communication channel between a server and client can read and/ or modify any messages between them.

Unlike the DTLS vulnerability, clients and servers using OpenSSL in typical ways are vulnerable.

The impact of this final bug is far less serious than Heartbleed. For an attacker, setting up to mount a man-in-the-middle attack is a considerable task in itself.

For this attack to work, both the server and the client application must use a vulnerable version of OpenSSL. Most popular web browsers, the most common client applications, use an alternative to OpenSSL and are therefore not vulnerable.

What should regular internet users do?

It appears that most regular users won't have to take any action in response to this attack.

Some non-browser client applications (such as music players and chat programs) may need to be updated. This process will generally not require anything proactive on a regular user's part if, as is standard, automatic updates are enabled. At most, you'll be prompted to install an updated version of the affected software which will then be downloaded and installed. Various distributors of Linux, which makes wide use of OpenSSL, have already issued updates.

Any internet accounts which are linked to such applications may require a password reset but again, users should be informed by the service provider of this requirement.

Websites may be unavailable for very short periods as fixed versions of OpenSSL are installed by their system administrators.

OpenSSL under the microscope (again)

When the Heartbleed bug was discovered, the programming community was both shocked and wryly amused at the elementary nature of the mistake.

Unfortunately, it seems to be something of a pattern. The DTLS vulnerability is caused by a mistake by the same German researcher and OpenSSL contributor and programmer, Robin Seggelmann, whose mistake caused the Heartbleed error. The new mistake is very similar in nature.

The fact that such a blatant and obvious mistake as the Heartbleed bug could make it into OpenSSL is, to borrow a phrase, the equivalent of "putting blood in the water" to security researchers.

For these individuals – professionals, university students and hobbyists – being the first to find and report a serious flaw in a high-profile application is a major achievement, in the same way that a paper published in *Nature* would be to a scientist.

The entire OpenSSL project, and particularly Seggelmann's code, has therefore been under renewed scrutiny. We're seeing some of the results with the collection of new vulnerabilities revealed this week.

In time, this process is likely to result in a much more secure OpenSSL. Work has also begun on a "fork" of OpenSSL by another development team. That means they have taken the existing OpenSSL code and begun working on it independently.

This new fork, LibreSSL, is managed by a development team renowned for their obsession with security; it will be interesting to see whether their modified version is widely adopted.

But in the short and medium term it seems likely that more flaws in OpenSSL will be discovered. Therefore, there is every chance that this process will be repeated at least once more.

This story is published courtesy of The Conversation *(under Creative Commons-Attribution/No derivatives).*

Source: The Conversation

Citation: Six more bugs found in popular OpenSSL security tool (2014, June 6) retrieved 23 April 2024 from https://phys.org/news/2014-06-bugs-popular-openssl-tool.html