What You Need To Know About Heartbleed, A Really Major Bug That Short-Circuits Web Security

A flaw in OpenSSL may have exposed passwords and encryption keys across the Web. Here's how to protect yourself and your sites.

Lauren Orsini April 08, 2014

(Lauren Orsini is a professional journalist, amateur programmer and full time otaku. She lives in Arlington, VA, and complains about the traffic.)

Heartbleed, a long-undiscovered bug in cryptographic software called OpenSSL that secures Web communications, may have left roughly two-thirds of the Web vulnerable to eavesdropping *for the past two years*. Heartbleed isn't your garden-variety vulnerability, so here's a quick guide to what it is, why it's so serious, and what you can do to keep your data safe.

What's Heartbleed?

The short version is that it's a vulnerability in the way your browser talks to a website over an encrypted channel. An attacker could theoretically take advantage of the bug to unravel the secure channels used by banks, e-commerce sites and other sensitive locations to steal passwords and other sensitive information.

The slightly longer version is that Heartbleed is a flaw in the OpenSSL implementation of the basic cryptographic protocol that secures Web communications, known as SSL.

What's SSL?

It stands for Secure Socket Layer, a cryptographic protocol that puts the S in "https"—the prefix you see on Web addresses when they're using a secure, encrypted connection. SSL basically ensures that no one can eavesdrop while you're banking, shopping or doing anything else.

What's OpenSSL?

OpenSSL is an open-source implementation of SSL and its successor protocol, TLS (which stands for Transport Security Layer). It's the default cryptographic library in the Apache and nginx Web servers, which together powered almost exactly two-thirds of all active websites as of April, according to Netcraft data (h/t Ars Technica). That means OpenSSL is used to protect sensitive Web communications across a vast swathe of the Internet.

Let's Go Back To Heartbleed. What Exactly Does It Do?

Officially known as CVE-2014-0160, Heartbleed is a recently discovered bug in OpenSSL that could allow an attack to read information off a Web server even though it's supposed to be secured against intrusion. The bug affects an OpenSSL extension known as “heartbeat,” which makes it possible to keep a secure communication channel open without re-negotiating security protocols over and over again.

In effect, the bug allows a malicious users to request data from a Web server's memory—data that could include the site's SSL encryption keys, user passwords and other sensitive information. According to Heartbleed, a website established by researchers at Codenomicon who identified the bug (as did a Google engineer):

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

Why Is HeartBleed So Serious?

There are several reasons. The first, and most obvious, is that Web servers may leak sensitive data as a result of the vulnerability. The Heartbleed bug lets an attack force a server to cough up the contents of its active memory (albeit in 64KB chunks). Depending on what the server happens to be doing, its memory may contain usernames, passwords or credit card numbers.

Heartbleed also lets attackers obtain the server’s secret keys—cryptographic measures that are supposed to ensure only an owner can access sensitive data—in order to impersonate servers and decrypt their communications.

What’s more, the site explains, is that it’s disturbingly difficult to tell if somebody exploits it. That means victims may have no way to tell if they've been, well, victimized.

Finally, the Heartbleed bug has been in the wild for roughly two years. That's a lot of potential exploitation, should any bad guys have stumbled across it in that time. Even worse, if anyone recorded encrypted traffic over that time *and* extracted a server's encryption keys, they can go back and decrypt past communications at their leisure.

Sites that use perfect forward secrecy, like Gmail, should be protected from retrospective decryption. Unfortunately, perfect forward secrecy isn't widely used across the Web.

Can I Tell If My Site Has Been Exploited Via Heartbleed?

No. However, you can test your site to see if it's vulnerable.

What Can I Do To Protect My Site?

If your site turns out to be vulnerable, OpenSSL recommends that you upgrade to OpenSSL version 1.0.1g, which patches the Heartbleed vulnerability. If you can't upgrade for some reason, you can disable OpenSSL heartbeat support for a quick fix.

Even if you patch the bug, however, it’s impossible to tell if an attack took place in the past.

What Can I Do To Protect Myself As An Internet User?

Now is a very good time to change passwords on sites that haven't been affected or updated to patch their servers. (You can check here.) If a site has been affected, however, you'll want to wait until it addresses this vulnerability before changing that login.

Even if you don’t think you were affected, it's possible that sites you regularly use—like Yahoo—might have been. Ronald Prins of security firm Fox-IT tweeted that he used the Heartbleed vulnerability to find a Yahoo username and password, and provided proof on his blog. However, Yahoo has apparently patched its servers, as a Heartbleed vulnerability test for Yahoo.com has shown up clean since 4 PM Eastern.

Has Anyone Exploited The Heartbleed Vulnerability Yet?

We don’t know. Security researchers who study the bug have noted that when they exploit it, nothing unusual shows up on the logs. So if bad actors have used it for attacks, we can’t tell.

Logo via Codenomicon

Updated to adjust advice. Users hoping to secure their logins should not change those passwords on an affected site that hasn't yet updated its server. (Thank you, @RichardNixon!)