HEARTBLEED

Everything was secure until, suddenly, it wasn't.

WHAT HAPPENED?

Without leaving a trace, adversaries can use vulnerable versions of OpenSSL software to steal usernames and passwords, instant messages, emails, business critical documents and communications.

The vulnerability was introduced December, 2011.

The vulnerability was discovered April, 2014.

Some instances of OpenSSL have been repaired, but not all.

SURPRISED? DON'T BE.

Fact is, organizations unknowingly use risky components all of the time. According to a study conducted by Sonatype, Inc. and Aspect Security, there is widespread use of vulnerable components—long after alerts have been issued.

In one year, there were **46 million** downloads of insecure versions of the 31 most popular open source security libraries.[1] Other examples include:[2]

STRUTS2
WEB APPLICATION FRAMEWORK

CVE -2013-2251
Release Date: July 20, 2013
CVSS v2 Base Score: **9.3 HIGH**
Impact Subscore: **10.0**
Exploitability Subscore: **8.6**

Since then,
4,076 organizations
have downloaded it
179,050 times

HTTP CLIENT
HTTP IMPLEMENTATION FOR JAVA

CVE -2012-5783
Release Date: November 4, 2012
CVSS v2 Base Score: **5.8 MEDIUM**
Impact Subscore: **4.9**
Exploitability Subscore: **8.6**

Since then,
29,468 organizations
have downloaded it
3,749,193 times

BOUNCY CASTLE
CRYPTOGRAPHY API

CVE -2007-6721
Release Date: March 30, 2009
CVSS v2 Base Score: **10.0 HIGH**
Impact Subscore: **10.0**
Exploitability Subscore: **10.0**

Since then,
11,236 organizations
have downloaded it
214,484 times

JETTY
WEB APPLICATION SERVER

CVE -2009-4611
Release Date: January 13, 2010
CVSS v2 Base Score: **5.0 MEDIUM**
Impact Subscore: **2.9**
Exploitability Subscore: **10.0**

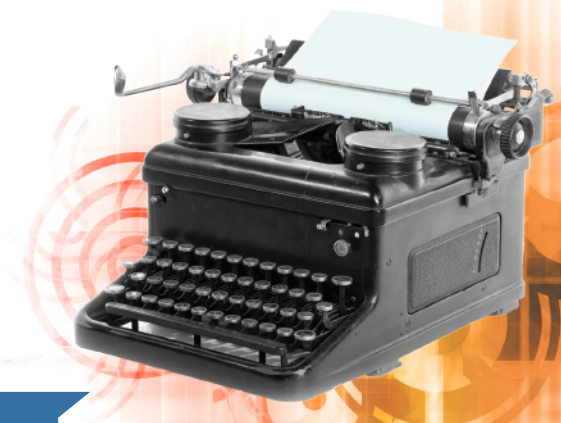
Since then,
36,181 organizations
have downloaded it
5,174,913 times

WHAT CAN BE DONE?

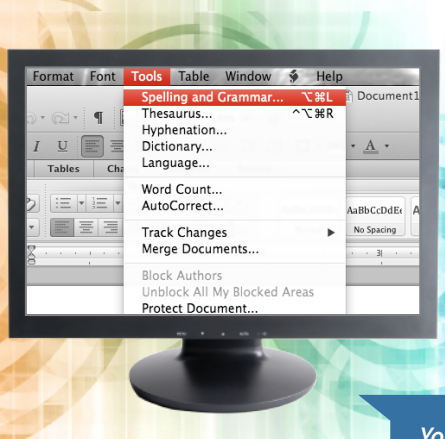
Three steps to prevent future open source component attacks:

1 USE AUTOMATION TO CONTROL WHAT'S IN YOUR SOFTWARE

27,000 components are downloaded every hour of every day[3]. Automated inventory and monitoring of risky components must be a mandatory part of modern software development.



The process of manually reviewing components for approval is as outdated as the Underwood typewriter. Get automated, people!



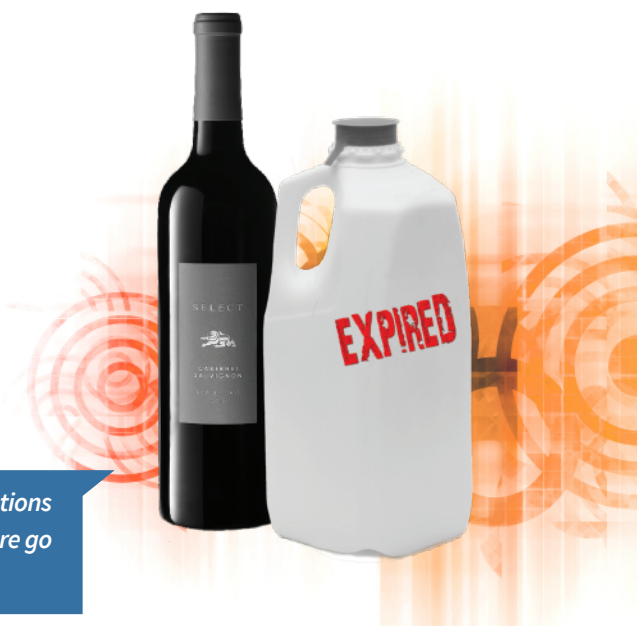
2 STOP USING VULNERABLE COMPONENTS

Make it easy for developers to choose the safest, highest quality open source components. Component security must be integrated into the tools developers use every day.

You don't wait until a book is published before you check for typos. New tools enable developers to replace flawed components as easily as a modern spellchecker. Make it easy for developers!

3 KNOW WHAT & WHERE NEW VULNERABILITIES AFFECT YOU

With continuous monitoring, you can immediately know what applications are affected when new vulnerabilities are reported and learn which component versions are safest.



Since software ages like milk and not wine, applications become less secure over time. Don't let your software go sour! It's an avoidable problem!

SUMMARY

The Heartbleed bug is a single instance of a vulnerability that had world-wide impact. We have the responsibility—and the ability—to build more secure software. Join us at:

WWW.SONATYPE.COM/HEARTBLEED

[1]2012 Executive Brief: Addressing Security Concerns in Open Source Components by Sonatype, Inc. and Aspect Security
[2, 3,4]Sonatype, Inc. analysis of activity in (Maven) Central Repository