EPIDEMIOLOGY OF SOFTWARE VULNERABILITIES: A STUDY OF ATTACK SURFACE SPREAD

Kymberlee Price
@Kym_Possible
Director of Strategic Operations
Synack

Jake Kouns
@jkouns
CISO
Risk Based Security
UNDER PRESSURE
what am I forgetting?

Hint! security
Development Realities

Can only pick two!
Developers using established third-party libraries to
  – Speed up the development process
  – Realize quality improvements over creating an in-house proprietary solution from the ground up.
SO WHAT IS THE PROBLEM?
Third-party programs are responsible for 76% of the vulnerabilities discovered in the 50 most popular programs in 2013, say the results of Secunia's Vulnerability Review 2014.

Where The Vulns Are

“When reviewing this report, you find that it is flawed and not referring to 3rd Party Libraries. But rather from third party – i.e. – non-Microsoft – programs.
<table>
<thead>
<tr>
<th>Year</th>
<th>Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>5,027</td>
</tr>
<tr>
<td>2013</td>
<td>10,868</td>
</tr>
<tr>
<td>2012</td>
<td>10,205</td>
</tr>
<tr>
<td>2011</td>
<td>7,852</td>
</tr>
<tr>
<td>2010</td>
<td>9,110</td>
</tr>
<tr>
<td>2009</td>
<td>8,147</td>
</tr>
<tr>
<td>2008</td>
<td>9,740</td>
</tr>
<tr>
<td>2007</td>
<td>9,559</td>
</tr>
<tr>
<td>2006</td>
<td>11,029</td>
</tr>
</tbody>
</table>

Source: OSVDB.org  
*YTD June 2014
Vulnerabilities by Type

Vulnerabilities in OSVDB by Quarter by Type

<table>
<thead>
<tr>
<th>Year</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>11,029</td>
</tr>
<tr>
<td>2007</td>
<td></td>
</tr>
<tr>
<td>2008</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>7,852</td>
</tr>
<tr>
<td>2012</td>
<td>10,205</td>
</tr>
<tr>
<td>2013</td>
<td>10,868</td>
</tr>
<tr>
<td>2014</td>
<td>5,027</td>
</tr>
</tbody>
</table>

Source: OSVDB.org
*YTD June 2014
Estimated 100+ Vendors impacted by Heartbleed
Vendor’s Impacted By HeartBleed
The Heartbleed bug may be a devastating flaw still affecting thousands of websites, but efforts to patch any remaining systems are effectively over.

Two months after Heartbleed surfaced in April, more than 300,000 unpatched servers remain vulnerable to Heartbleed. The figure comes from Errata Security's Robert Graham who recently scanned the Internet for a third time to get a count of Heartbleed-vulnerable sites.
Efficiency At What Cost?

• Not just one library impacting many organizations
• A single application may have as many as 100 different third party libraries implemented
  – That is a whole lot of patching to keep up on for both devs and customers
LETS TALK DATA
What libraries are the biggest offenders for spreading pestilence?

- Volume of vulnerabilities 2007-2014
- Frequency of release 2007-2014
- Average vulnerability severity
- Pervasiveness of library usage
OpenSSL

Vulnerabilities
Advisories

2007 2008 2009 2010 2011 2012 2013 2014

2 2 1 4 5 3 2 1

11 7 3 11 6 3 5 3

45 Vulns

1-6 releases per year

Average CVSS 5.39
Vuln Spread:

...And multiple products by HP, Oracle (including Java), F-Secure, IBM, MySQL, Novell, OpenBSD, Intel, Juniper, Rapid7, nginx, Huawei, Trend Micro, Linux, Tableau, McAfee, F5, Cisco, Fortinet, Sophos, Python, Citrix, SUSE, Ubuntu, Debian, FreeBSD, RedHat...
FreeType

- Vulnerabilities
- Advisories

2007: 3, 3
2008: 3, 1
2009: 2, 1
2010: 3, 3
2011: 3, 3
2012: 22
2013: 0, 0
2014: 2, 1

Average CVSS: 7.89

50 Vulns

2 releases per year
Vuln Spread:

And also... OSX, Webkit, Firefox, OpenJDK, OpenOffice, StarOffice, Ubuntu, Gentoo, Oracle Solaris, SUSE, Slackware, BlackBerry products, Fedora, RedHat, Debian, Avaya products, PlayStation 3/4/Vita, Opera for Wii, multiple video games...
LibPNG

- 26 Vulns
- 3 releases per year
- Average CVSS 6.58

<table>
<thead>
<tr>
<th>Year</th>
<th>Vulnerabilities</th>
<th>Advisories</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2008</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2009</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2010</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2011</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>2012</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2013</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2014</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Vuln Spread:

Visio, PowerPoint, Adobe Photoshop/Flash/Illustrator, Webkit, iOS, OSX, Android, GIMP, Fedora, Debian, Ubuntu, Slackware, Red Hat, SUSE, Gentoo, Oracle Solaris, VMWare Server, and countless applications.
FFmpeg

64 Vulns

3 releases per year

Average CVSS 8.35
Vuln Spread: YouTube, Chromium, ChromeOS, QuickTime, JavaCV, DirectShow, DropCam, Gstreamer, Mplayer, xine, PlayStation, Gentoo, Ubuntu, Debian, FreeBSD, Mandriva Linux, LaunchPad, Libav..
WHAT'S BEING DONE CURRENTLY?
ACCOUNTABILITY
Open source is secure because everyone can review it - more eyes makes all bugs shallow.

Code Quality

Everyone *could* look at it, but they don’t. Accountability for quality is deferred.
That means closed-source is more secure because no one can review it and supported by enterprises, right?

Code Quality

Bad code is just that, bad code and exists in Closed Source as well.
Many projects are not well funded or resourced

Bug bounty programs
  – Crowd source more eye on critical software or websites

Code audit initiatives
  – TrueCrypt
  – OpenSSL
Core Infrastructure Initiative first round funding, announced 29May14: Network Time Protocol, OpenSSH and OpenSSL.

OpenSSL will receive funds for two, fulltime core developers. The OpenSSL project is accepting additional donations, which can be coordinated directly with the OpenSSL Foundation (contact at info@opensslfoundation.com).

The Open Crypto Audit Project (OCAP) will also receive funding in order to conduct a security audit of the OpenSSL code base. Other projects are under consideration and will be funded as assessments are completed and budget allows.
• Premise being that it is the only way to ensure the code is secure is to own responsibility for the code.

• Companies and groups are forking libraries
  – Google Blink (Webkit)
  – OpenBSD LibreSSL (OpenSSL)
“catastrophic failure”...

“unsafe for Linux”
Know What You’re Committing To

- Evaluate vuln trends in libraries as part of selection criteria
- Do you have enough people resources to handle the expected vulnerability load?
- What is your prioritization model?
- What early monitoring processes can you put in place to minimize surprises?
- Can you identify low friction areas to diminish risk?
- Communicate, communicate, communicate
Network Security Incident Response Procedures

Start

We've Been Hacked
- I discovered it
- Someone else discovered it

Cover It Up?
- Yes
- No

Hope it doesn't happen again

Any idea what happened?
- Yes
- No

Close the Hole?
- Can't
- Can

Throw Money at the Problem and Hope it Helps

Despair
Incident Response

1. Identify Issue
2. Assess Impact
3. Dev & Test Fix
4. Public Release w/ CVE
5. Post Release

So you’re a software vendor...

But wait! The vulnerability was in a third party library!

Enterprise admin? Your patch lifecycle starts HERE
Vendors, Monitor Your Libraries!

- Source code scanning tools
- Vulnerability Database providers
- Your Legal Team is your friend
What Else Can Be Done? (dev edition)

- Vendor security testing
  - Active security testing of third party and OSS libraries using in house and/or outsourced security researchers
- Proactive plan for routine patching as part of dev lifecycle
- Robust Incident Response Plans for critical vulnerability disclosures that include Dev Team resources for product sustainment
What Else Can Be Done? (IT edition)

- Network scanning - know what software is in use where.
- Know where risk is in your environment
  - Monitor OSS advisory releases for software used in your apps/products.
  - ASK your SW Vendors if they are affected by third party vulns
  - Code and Network scanning for un-patched vulnerabilities.
- Plot the vulnerability trends for your environment
- Plan time for sustainment
DISCUSSION!
EPIDEMIOLOGY OF SOFTWARE VULNERABILITIES: A STUDY OF ATTACK SURFACE SPREAD

Kymberlee Price
@Kym_Possible
Director of Strategic Operations
Synack

Jake Kouns
@jkouns
CISO
Risk Based Security