The Game of Hide and Seek, Hidden Risks in Modern Software Development

Ryan Berg
CSO
Sonatype
@ryanberg00
Agenda

- The changing dynamics surrounding application security
- Why this is a supply chain problem
- What should you be doing, but likely aren’t
- Q&A
The Language of Security is Risk

Security Product Management

http://xkcd.com/795/
What is Risk?
What does the snail tell us?

“…WE OWE A DUTY OF REASONABLE CARE TO OUR NEIGHBOR”

Lord Atkin: Donoghue v. Stevenson (1932)
“...a manufacturer of products, which he sells in such a form as to show that he intends them to reach the ultimate consumer in the form in which they left him with no reasonable possibility of intermediate examination, and with knowledge that the absence of reasonable care in the preparation or putting up of products will result in an injury to the consumer's life or property, owes a duty to the consumer to take that reasonable care.”
What is Risk?

*United States v. Carroll Towing Co.*
159 F.2d 169 (2d. Cir. 1947)
The cost of Doing nothing can’t be ignored

“...IF THE PROBABILITY BE CALLED P; THE INJURY, L; AND THE BURDEN, B; LIABILITY DEPENDS UPON WHETHER B IS LESS THAN L MULTIPLIED BY P: I.E., WHETHER B < PL”.

Translation: If the Cost of Protecting Against Harm is less than the Cost of the Damage Multiplied by the Likelihood of the Damage, then there is negligence.

Risk = probability x impact
Modern software development HAS CHANGED

Application security HASN’T CHANGED ENOUGH
Modern software development
HAS CHANGED

Application security
HASN’T CHANGED ENOUGH
Modern software development HAS CHANGED

Application security HASN’T CHANGED ENOUGH
APPLICATIONS are assembled using third party “components,” most of which are open source.

In fact, 90% of a typical application is open source.
Open source usage is EXPLODING

Yesterday’s source code is today’s OPEN SOURCE
Creating today’s software

SUPPLY CHAIN

COMPONENT SELECTION
DEVELOPMENT
BUILD AND DEPLOY
PRODUCTION
Do you know who your SUPPLIERS ARE?
OPEN SOURCE:
QUALITY
INNOVATION
EFFICIENCY
NO CONTROLS.
OPEN ACCESS.
HACKER TARGETS.
Security spans the Enterprise

Security concerns are across the Enterprise

<table>
<thead>
<tr>
<th>Development</th>
<th>Operations</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>Performance</td>
<td>Security</td>
</tr>
<tr>
<td>Usability</td>
<td>Reliability/Scalability</td>
<td>Compliance</td>
</tr>
<tr>
<td>Performance</td>
<td>Compliance</td>
<td>Everything Else</td>
</tr>
<tr>
<td>Reliability/Scalability</td>
<td>Security</td>
<td></td>
</tr>
<tr>
<td>Maintainability</td>
<td>Maintainability</td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Features/Usability</td>
<td></td>
</tr>
<tr>
<td>Compliance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Haven’t We Learned Compliant Does Not Mean Secure, Often the Opposite

That pesky hacker won’t get to our data now

Our automated source code scanner will find all the holes he could ever use

Phew...a PCI compliant “green light” status, case closed my friend!
## Evolution of Spend

<table>
<thead>
<tr>
<th>Prevention</th>
<th>Detection</th>
<th>Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall</td>
<td>IDS</td>
<td>SIEM</td>
</tr>
<tr>
<td>Encryption</td>
<td>SAST</td>
<td>DAM</td>
</tr>
<tr>
<td>IPS</td>
<td>DAST</td>
<td>RAST</td>
</tr>
<tr>
<td>WebApp Firewall (WAF)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A disproportionate spend against the risk
Assembled 90%

Source: 2012 / 2013 Sonatype analysis of more than 1,000 enterprise applications
What me worry, tis’ just a bit of floating ice
Components are a hidden risk
What are you doing to **ADDRESS THIS RISK?**
What are you doing to **ADDRESS THIS RISK?**
FOSS Review Board
Scans post development
Golden repository
Approval workflow
FOSS Review Board
Scans post development

NOTHING?

Golden repository
Approval workflow
If you’re not using secure COMPONENTS you’re not securing your APPS.
Today’s approaches AREN’T WORKING

- 46m vulnerable components downloaded!
- 90% of repositories have 1+ critical vulnerability!
- 71% of apps have 1+ critical or severe vulnerability!

Component Selection | Development | Build and Deploy | Production
WHY?

The problem is too complex to manage manually.

**Complexity**
One component may rely on 100s of others

**Diversity**
- 40,000 Projects
- 200M Classes
- 400K Components

**Volume**
Typical enterprise consumes 1,000s of components monthly

**Change**
Typical component is updated 4X per year
Manual processes DON’T WORK
Automation should ENFORCE POLICIES
Humans should manage EXCEPTIONS
To reduce cost per defect!

To achieve compliance!

To manage risk!

WHY is this important?
For Example: CVE-2013-2251

- Network exploitable
- Medium access complexity
- No authentication required for exploit
- Allows unauthorized disclosure of information; allows unauthorized modification; allows disruption of service
Hackers have first mover advantage
THE ANTI-PATTERNs
TURN OFF THE LIGHTS
LOCK THE DOORS
POINT FINGERS
THESE ARE NOT MY DROIDS
EVERYTHING IS A NAIL

To a man with a hammer, everything looks like a nail.
Success Requires Discipline
The Problem is Not Problem Discovery

• When our software development ecosystem looks like this it is easy to find problems

• The real challenge is to develop at scale and deliver continuous value continuously when everything else is a mess
The problem is no longer like this
It’s Starting to Look More Like This
Visibility and control. Automated and integrated policy enforcement throughout the software lifecycle.

Developer friendly – makes it easy to find and fix problems early.

Proactive and ongoing for continued trust.

Time for a **FRESH APPROACH?**
Got questions?

Get the ANSWERS.

- What production applications are at risk?
- What problems are most critical?
- What components are being used?
  Where are they?
- Which components have known security vulnerabilities?
- What are our license obligations?
- Do our applications comply with our policies?
- How can we choose the best components from the start?
Building A Better Bridge Between Dev, Ops and Security

- Need to recognize that the priorities are different
- Tooling needs to adopt the practice of the practitioner not the other way around
- A Tool is not a process and a process is not a tool learn to leverage both.
Building A Better Bridge Between Dev, Ops and Security

- Need to recognize that the priorities are different
- Tooling needs to adopt the practice of the practitioner not the other way around
- A Tool is not a process and a process is not a tool learn to leverage both.
Go Fast. Be Secure.

Build security in from the start

Enforce policy in the tools you already use

Reduce risk by automating governance throughout the lifecycle

Reduce cost by fixing early in the process

React to new threats by knowing what they are and where to fix them

Go fast by using tools your developers already know
Thank You
@ryanberg00

Announcing a **NEW BREED** of On-Demand Application Security

LEARN MORE
Sonatype:
South Hall #2327

HP Fortify on Demand:
North Hall #3401

www.sonatype.com/fortify