The OWASP Foundation
http://www.owasp.org

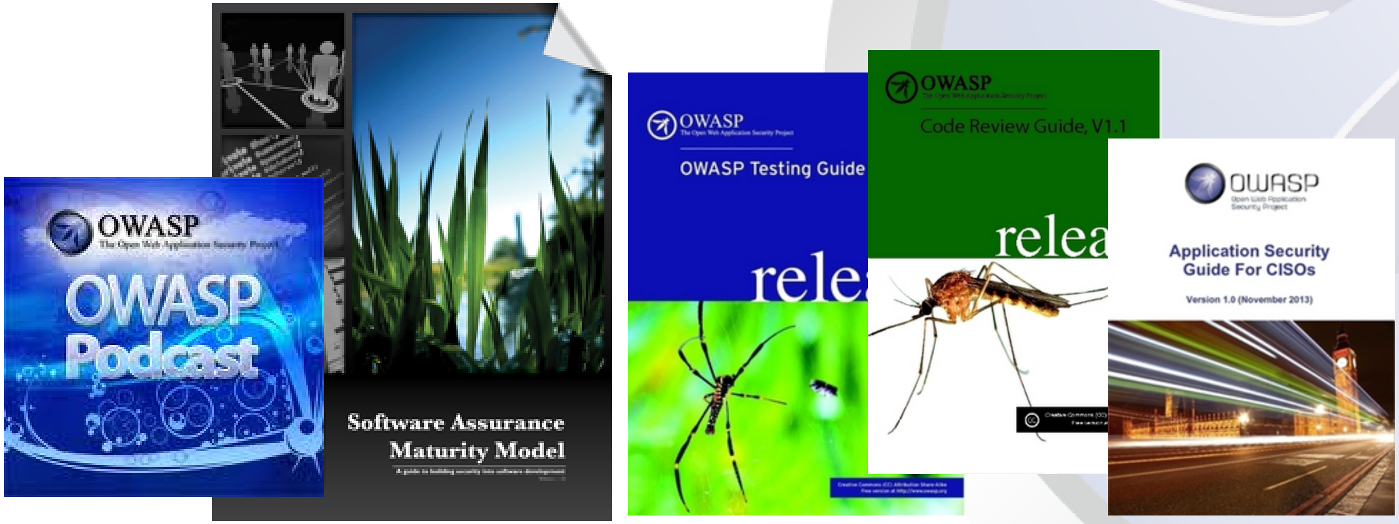
How Do I Approach Application Security?

San Francisco 2014

Jim Manico
OWASP GLOBAL BOARD MEMBER
OWASP Cheat-Sheet Project Lead

Eoin Keary
CTO BCC Risk Advisory / edgescan.com
OWASP GLOBAL BOARD MEMBER

Michael Coates
Director Shape Security
OWASP GLOBAL BOARD MEMBER


OWASP
The Open Web Application Security Project
OWASP Podcast
Software Assurance Maturity Model
A guide to building security into software development
OWASP
OWASP Testing Guide
rele
OWASP
Code Review Guide, V1.1
relea
OWASP
Application Security Guide For CISOs
Version 1.0 (November 2013)

The Numbers

Cyber Crime:
“Second cause of economic crime experienced by the financial services sector” – PwC

“Globally, every second, 18 adults become victims of ***cybercrime” - Norton***

US - $20.7 billion – (direct losses)
Globally 2012 - $110,000,000,000 – direct losses

“556 million adults across the world have first-hand experience of cybercrime -- more than the entire population of the European Union.”

Target's December 19 disclosure 100+ million payment cards

TARGET

Neiman Marcus

Snapchat: 4.6 million user records

Loyalty build
winning customers
building brands

OpenSSL
Cryptography and SSL/TLS Toolkit

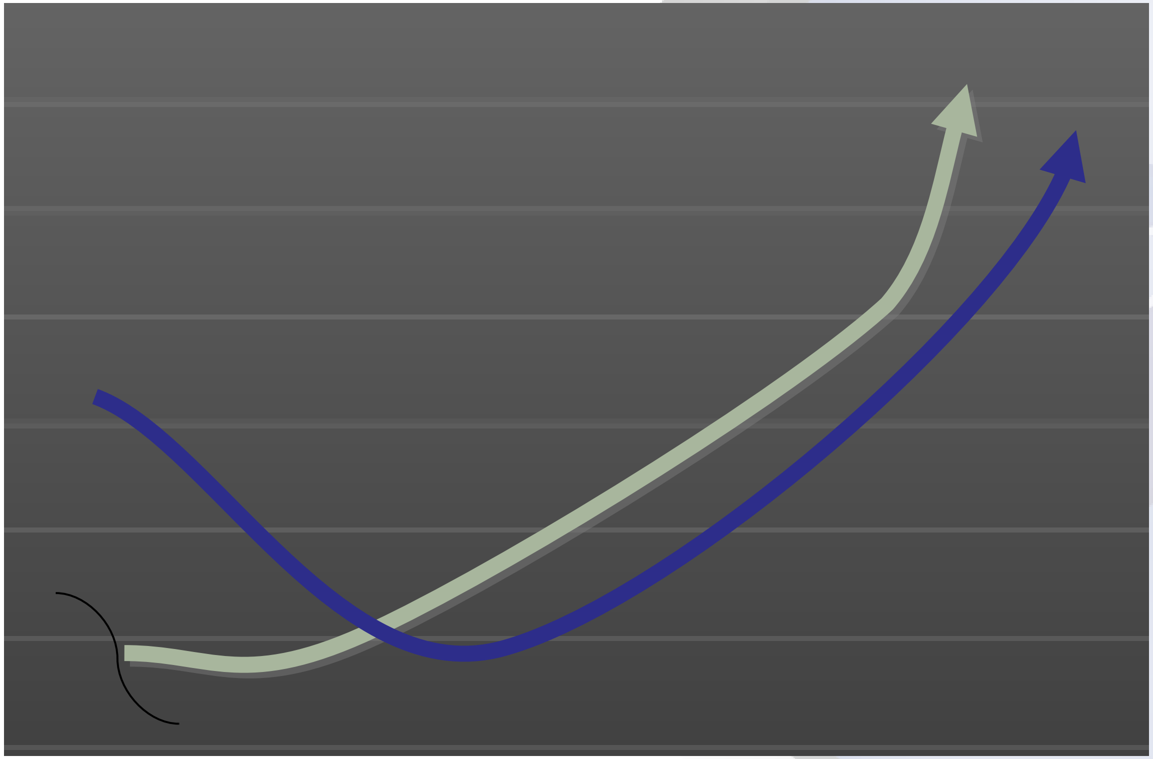
LoyaltyBuild November disclosure 1.5 million + records

Pentesting?

A penetration test is a method of evaluating computer and network security by simulating an attack on a computer system or network from external and internal threats.

This is a **component** of an overall security assessment.

Its (not) the $$$$



Information security spend

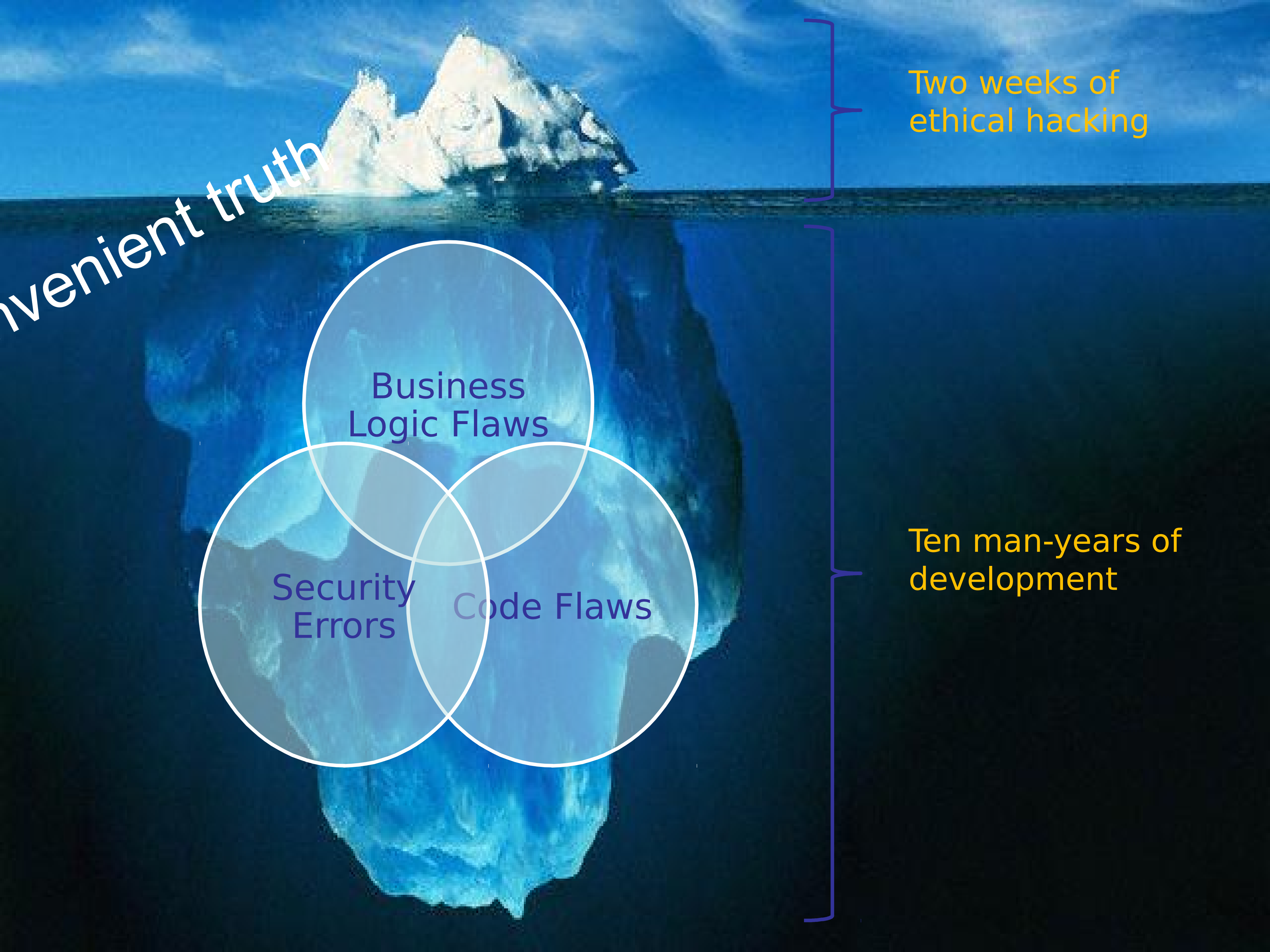
Security incidents (business impact)

But we are approaching this problem completely wrong and have been for years.....

Asymmetric Arms Race

A traditional end of cycle / Annual pentest only gives minimal security…..

There are too many variables and too little time to ensure “real security”.



nvenient truth

Business Logic Flaws

Security Errors

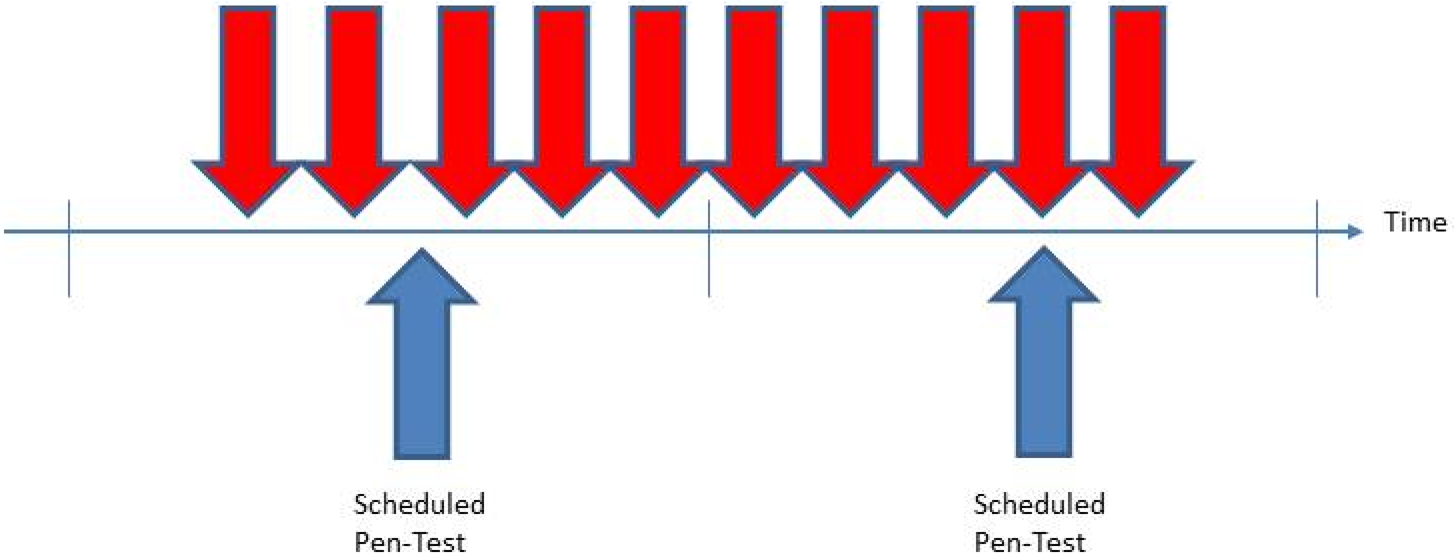
Code Flaws

Two weeks of ethical hacking

Ten man-years of development

An Attacker has 24x7x365 to Attack

Attacker Schedule



The Defender has 20 man days per year to detect and defend

Who has the edge?

HTTP Manipulation – Scanning – Is Not Enough

Problem has moved (back) to the client.
Some “Client Side” vulnerabilities can’t be tested via HTTP parameter testing.

AJAX
Flex/Flash/Air
Native Mobile Web Apps – Data Storage, leakage, malware
DOM XSS – Sinks & Sources in client script -> no HTTP rec

Scanning in not enough anymore.
We need DOM security assessment.
Javascript parsing/Taint analysis/String analysis/Manual Validation

window.location = http://example.com/a/page.ext?par=val#javascript&#x3a;alert(1)
jQuery.globalEval(userContent):

http://code.google.com/p/domxsswiki/

We can’t test what we don’t understand

Business Logic – Finite State Machines

Automated scanners are dumb

No idea of business state or state transitions
No clue about horizontal or vertical authorization / roles
No clue about business context

We test applications for security issues without knowing the business proce
We cant “break” logic (in a meaningful way) we don’t understand

Running a $30,000 scanning tool against your mission critical application?
Will this find flaws in your business logic or state machine?

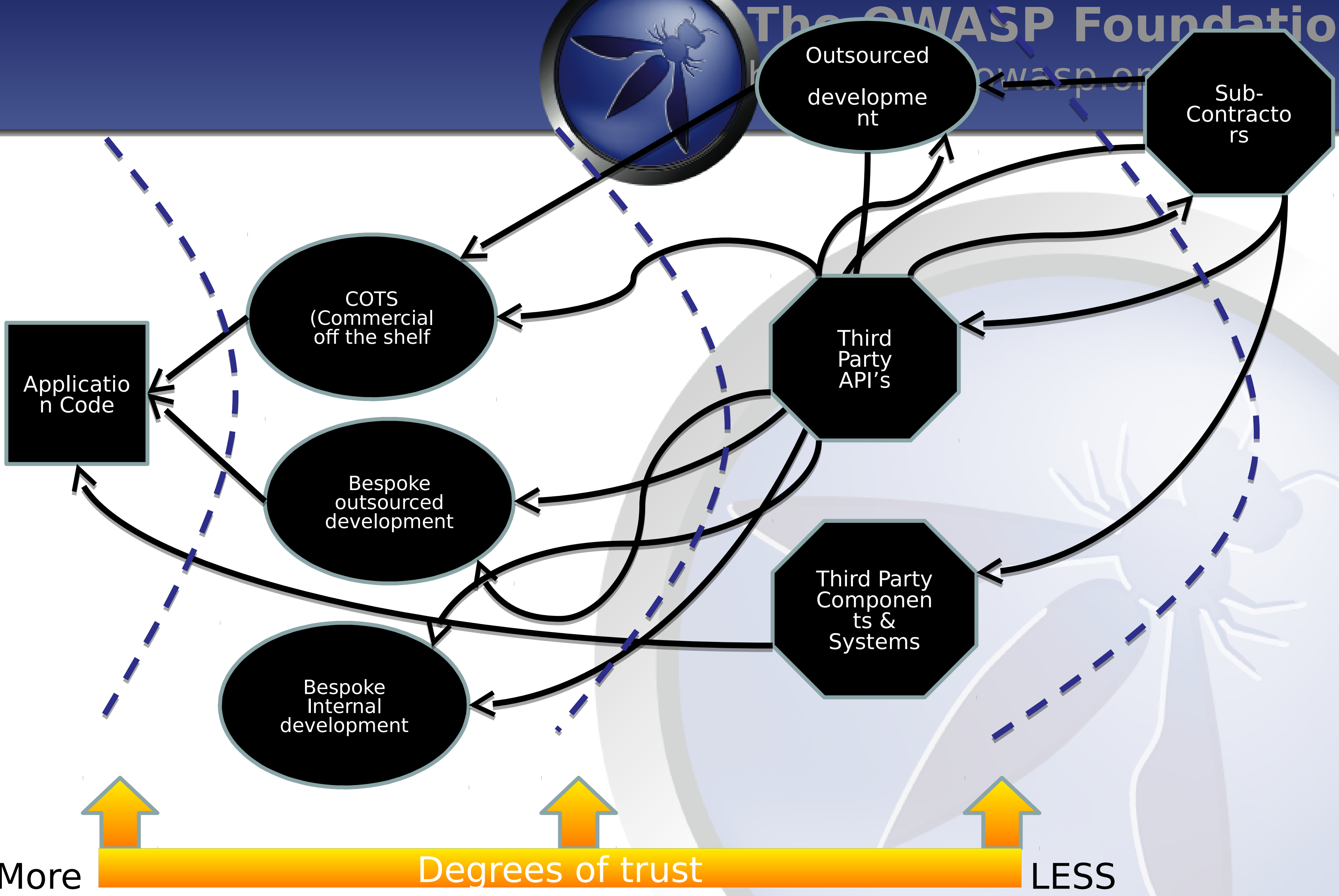
We need human intelligence & verification

“Onions”

SDL — 🧍 *Design review*
Threat Modeling 🧍
Code review/SAST/CI 🧍🤖
🧍🤖 *Negative use/abuse cases/Fuzzing/DAST*

Live/ Ongoing
🧍🤖 *Continuous/Frequent monitoring / Testing*
🧍 *Manual Validation*
Vulnerability management & Priority 🧍🤖
Dependency Management …. 🧍🤖

“Robots are good at detecting known unknowns
“Humans are good at detecting unknown unkno



Outsourced development

Sub-Contractors

COTS (Commercial off the shelf

Third Party API's

Application Code

Bespoke outsourced development

Third Party Components & Systems

Bespoke Internal development

More

Degrees of trust

LESS

You may not let some of the people who have developed your code into your offices!!

2012/13 Study of 31 popular open source libraries

- 19.8 million (26%) of the library downloads have known vulnerabilities
- Today's applications may use up to 30 or more libraries - 80% of the codebase

Dependencies

Spring application development framework : Downloaded 18 million times by over 43,000 organizations in the last year

– Vulnerability: Information leakage CVE-2011-2730

http://support.springsource.com/security/cve-2011-2730

In Apache CXF application framework:

4.2 million downloads.

- Vulnerability: Auth bypass CVE-2010-2076 & CVE 2012-0803

http://svn.apache.org/repos/asf/cxf/trunk/security/CVE-2010-2076.pdf
http://cxf.apache.org/cve-2012-0803.html

Do we test for "dependency" issues?

NO

Does your patch management policy cover application dependencies?

Check out:
https://github.com/jeremylong/DependencyCheck

Information flooding

(Melting a developers brain, white noise and "compliance")

Where do we go now?

Doing things right != Doing the right things

"Not all bugs/vulnerabilities are equal"
(is HttpOnly important if there is no XSS?)

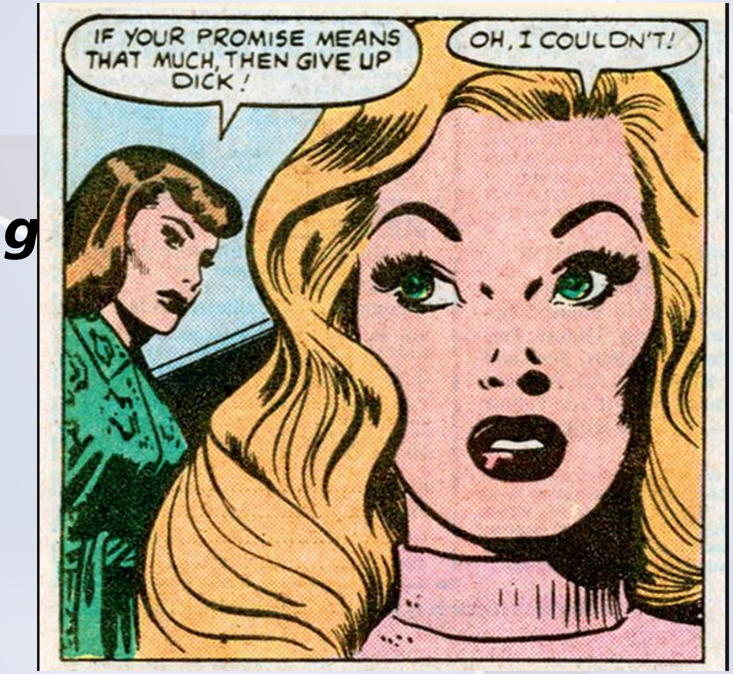
Contextualize Risk
(is XSS /SQLi always High Risk?)

Do developers need to fix everything

- ***Limited time***
- ***Finite Resources***
- ***Task Priority***
- ***Pass internal audit?***

White Noise

Context is important!



Dick Tracy

Problem

Explain issues in “Developer speak” (AKA English)

Is Cross-Site Scripting the same as SQL injection?

Both are injection attacks code and data being confused by system

Cross Site Scripting is primarily JavaScript injection

LDAP Injection, Command Injection, Log Injection, XSS, SQLI etc etc

Think old phone systems, Captain Crunch (John Draper)

Signaling data and voice data on same logical connection – Phone Phreaking

Out of context

XSS causes the browser to execute user supplied input as code. The input breaks out of the [data context] and becomes [execution context].

SQLI causes the database or source code calling the database to confuse [data context] and ANSI SQL [execution context].

Command injection mixes up [data context] and the [execution context].

So….

Building secure applications

.