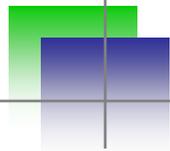


Stop throwing generic security requirement - Who has time?!

Blueinfy



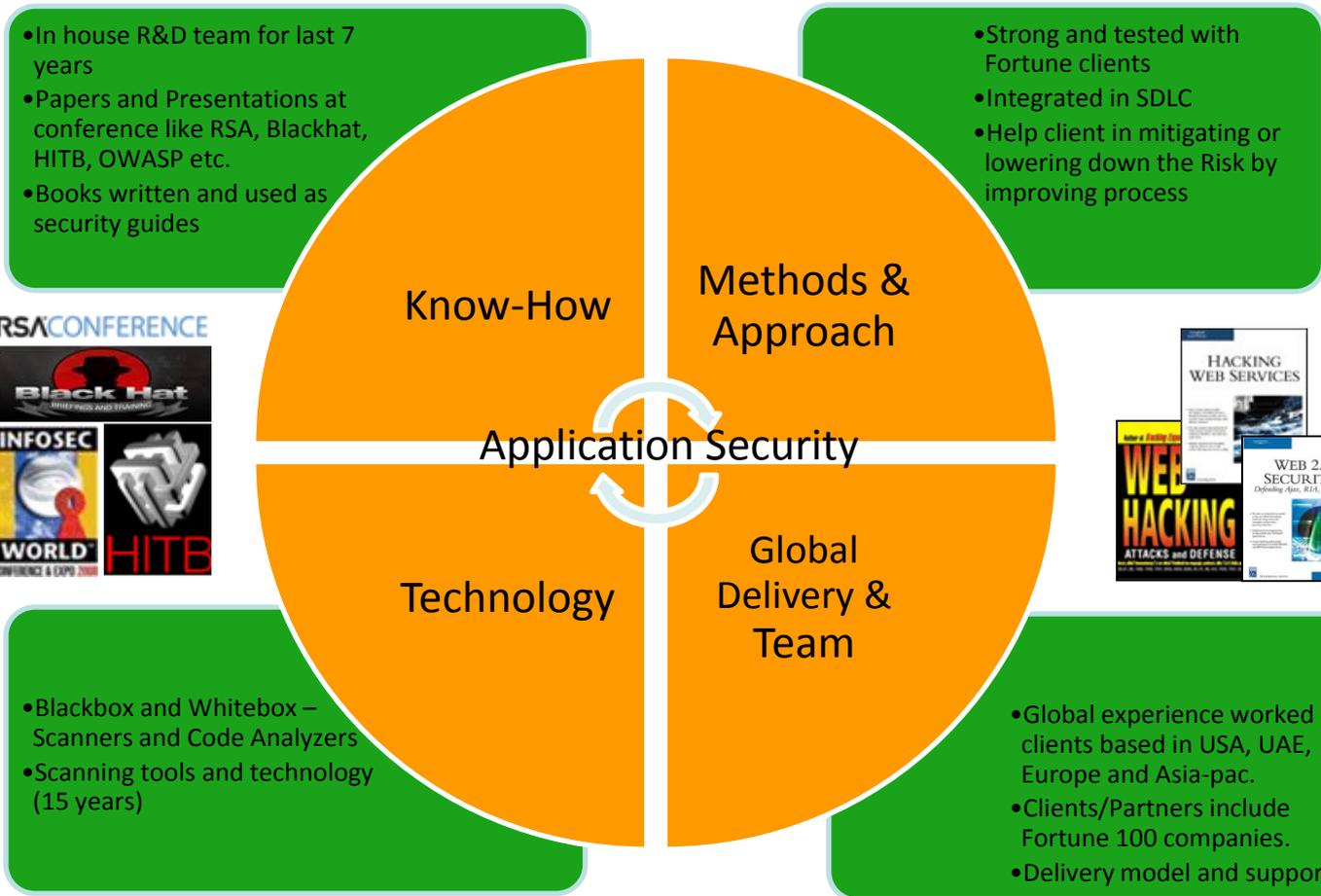
Who Am I?

hemil@blueinfy.com
<http://www.blueinfy.com>
Blog – <http://blog.blueinfy.com/>

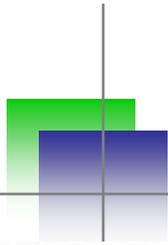
Blueinfy

- **Hemil Shah** – hemil@blueinfy.net
- **Co-CEO & Director, Blueinfy Solutions**
- **Past experience**
 - eSphere Security, HBO, KPMG, IL&FS, Net Square
- **Interest**
 - Web security research
- **Published research**
 - Articles / Papers – Packstroem, etc.
 - Web Tools – wsScanner, scanweb2.0, AppMap, AppCodeScan, AppPrint etc.
 - Mobile Tools – FSDroid, iAppliScan, DumpDroid

About Blueinfy



- BBC
- Dark Readings
- Bank Technology
- SecurityWeek
- MIT Technology Review



Security Concerns

Hackers access bankers' info on Fed website

Gary Strauss, USA TODAY 5:16 p.m. EST February 6, 2013

One Breach = \$1 Million To \$53 Million In Damages Per Year, Report Says

New Ponemon report studies real attack cases and their financial fallout; new Digital Forensics Association study tallies five-year public breach data

RubyGems.org hacked, interrupting Heroku services and putting sites using Rails at risk

January 30, 2013 8:49 PM

Anonymous Hacks US

Government Site, Threatens

Supreme 'Warheads'

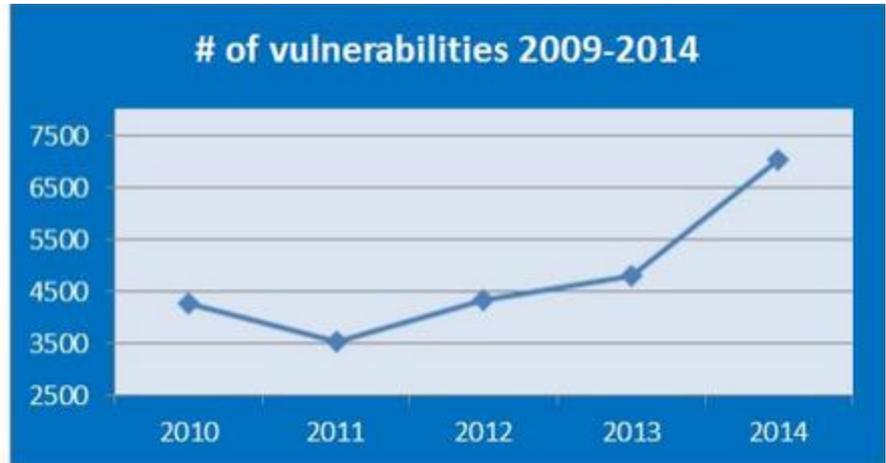
Hackers Disrupt Mexican Defense Ministry's Website

**Yahoo Mail users hit by widespread hacking, XSS exploit seemingly to blame
(Update: Fixed)**

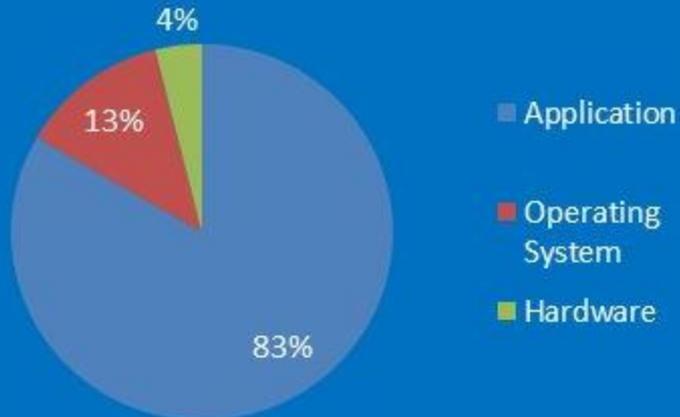
Aussie Travel Cover hack exposes details of 770,000 customers

Where are we standing?

Year	# of vulnerabilities
2010	4,258
2011	3,532
2012	4,347
2013	4,794
2014	7,038

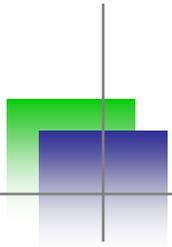


Vulnerability distribution by product type - 2014



Source: Most vulnerable operating systems and applications in 2014 by GFI Languard

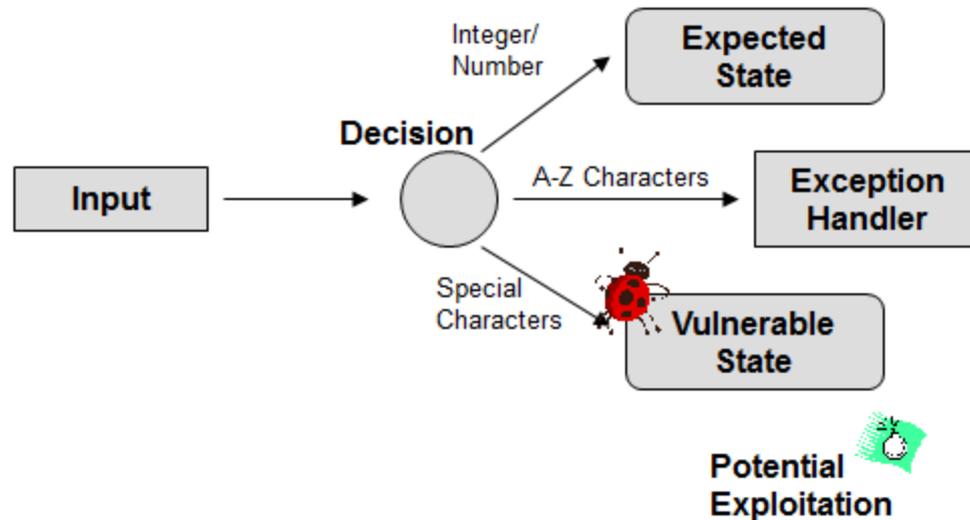
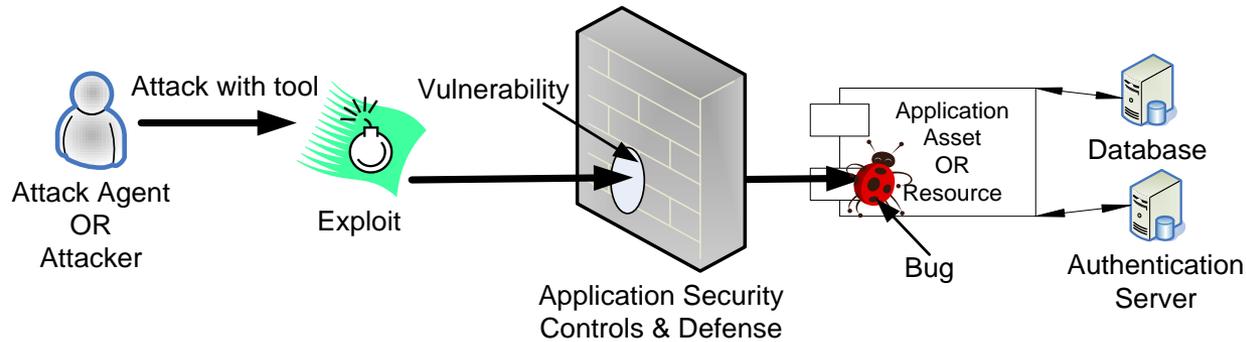
<http://www.gfi.com/blog/most-vulnerable-operating-systems-and-applications-in-2014/>

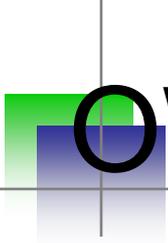


Attacks and Hacks

- 80% applications are having security issues
- Web Application Layer vulnerabilities are growing at higher rate in security space
- Client side hacking and vulnerabilities are on the rise – from 5% to 30% (IBM)
- Web browser vulnerabilities is growing at high rate
- End point exploitation shifting from OS to browser and its plugins

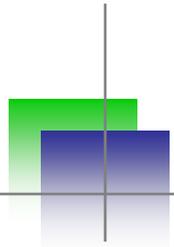
Vulnerabilities





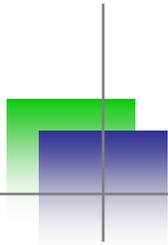
OWASP Top 10 Vulnerabilities

- A1 – Injections
- A2 – Broken authentication & session
- A3 – Cross site scripting
- A4 – Insecure direct object reference
- A5 – Security misconfiguration
- A6 – Sensitive data Exposure
- A7 – Missing functional level access control
- A8 – CSRF
- A9 – Using components with known vulnerabilities
- A10 – Unvalidated redirects and forwards



Category of vulnerability

- Architecture and design level vulnerabilities
- Code level vulnerabilities
 - Input validation
 - Access controls
- Configuration level vulnerabilities
 - Missing patches
 - Server hardening issues
- 9 out of OWASP Top 10 vulnerabilities are due to poor code



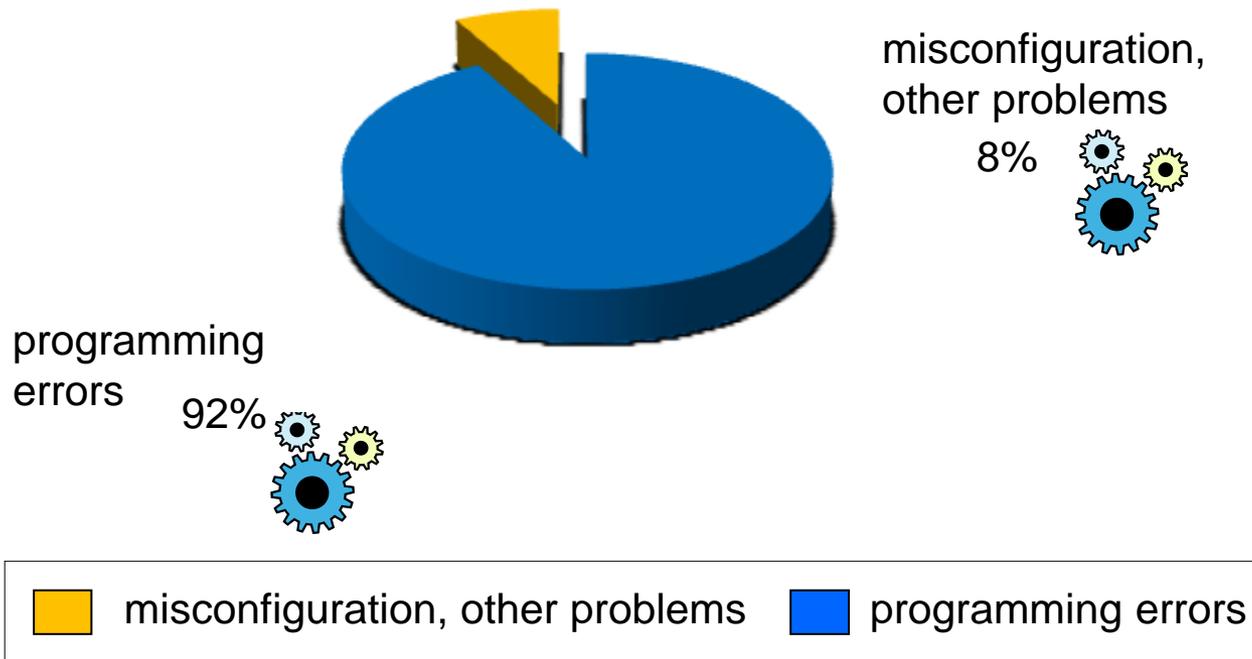
Level of vulnerabilities

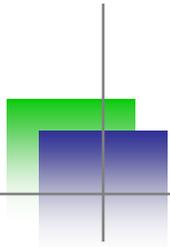
High ← Vulnerabilities → Low

Architect and Design Bugs	<ul style="list-style-type: none">• Session Management<ul style="list-style-type: none">• Improper Error Handling• Insecure Cryptographic Storage
Validation Bugs	<ul style="list-style-type: none">• Injection Flaws<ul style="list-style-type: none">• XSS• Malicious File Execution• CSRF
Logical Bugs	<ul style="list-style-type: none">• Indirect Object Reference• Failure to Restrict URL<ul style="list-style-type: none">• Broken Authentication

Root cause of Vulnerabilities

Security Survey : Vulnerability Distribution

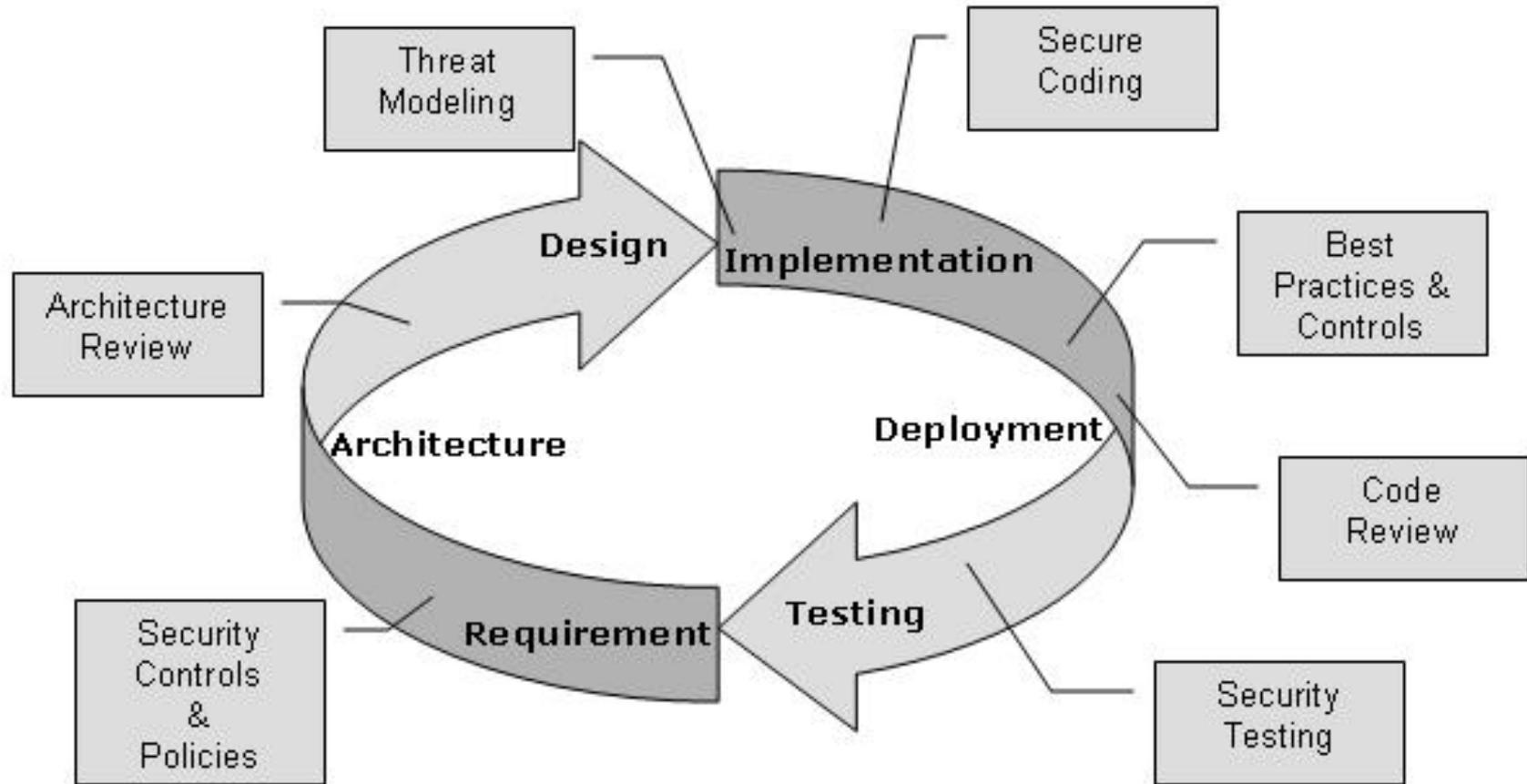


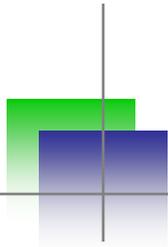


Root cause analysis

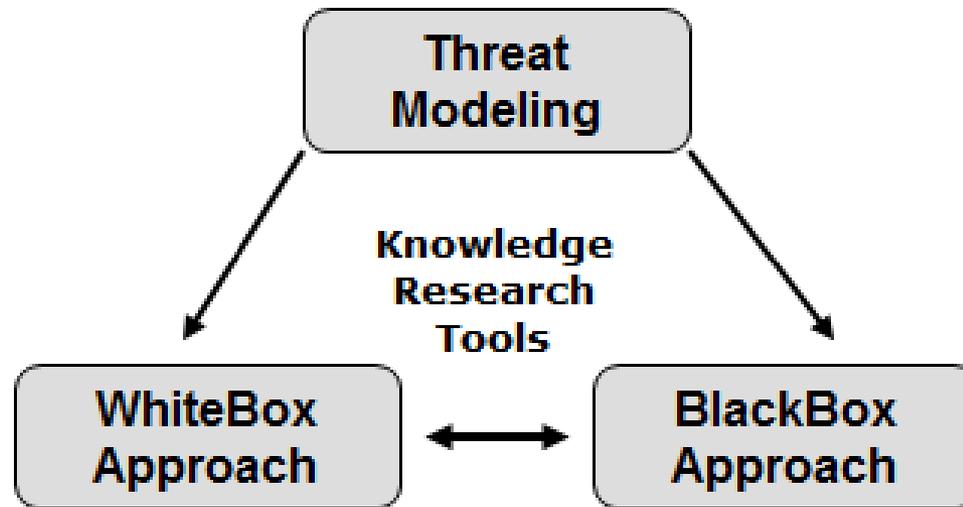
- 9 out of OWASP Top 10 vulnerabilities are due to poor code implementation
- Does this mean???
 - Developers are at the fault
 - Project managers are at the fault
 - Missing training to the developers – Is it really possible???
 - Issue in designing ADLC processes???

Secure ADLC – Is it enough???

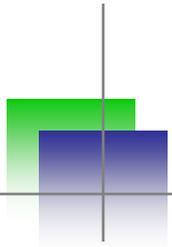




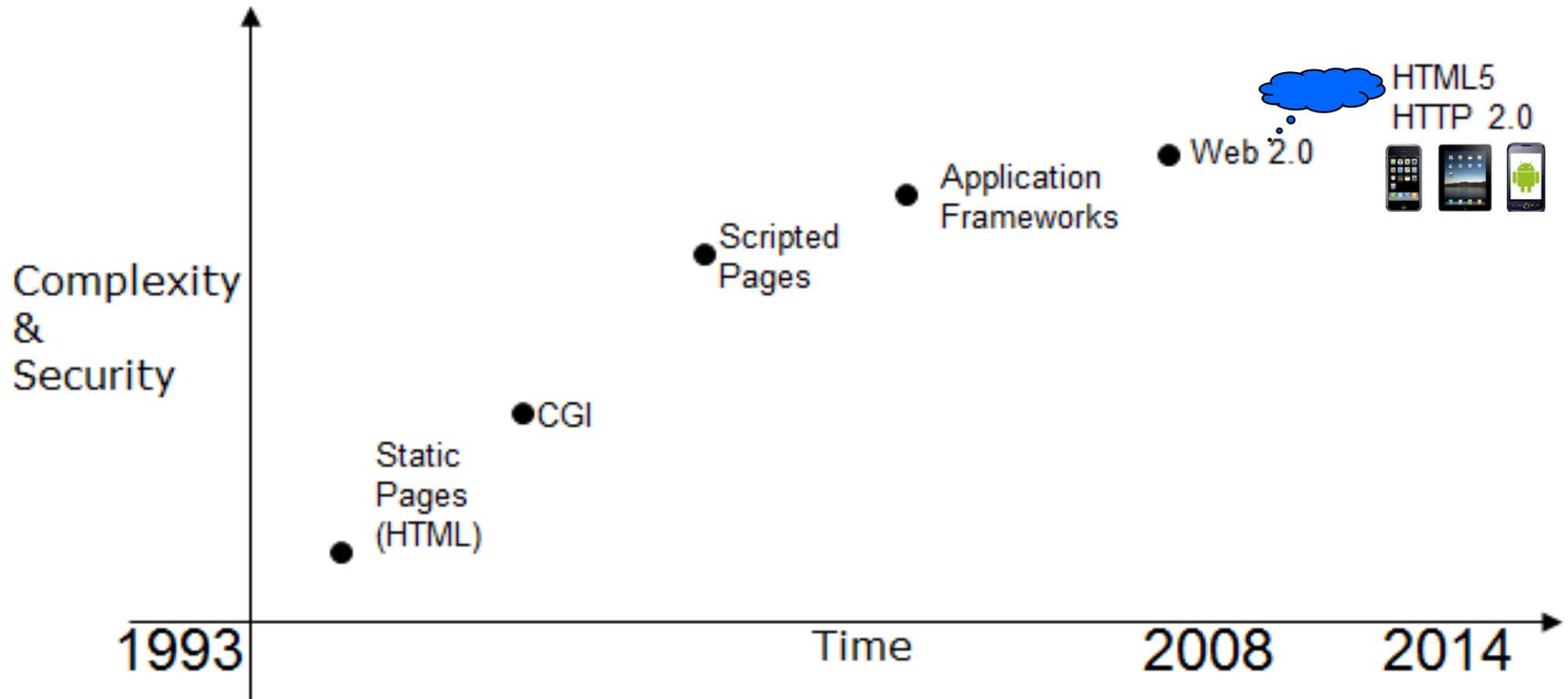
Ideal world



Knowledge = Secure Coding Guideline

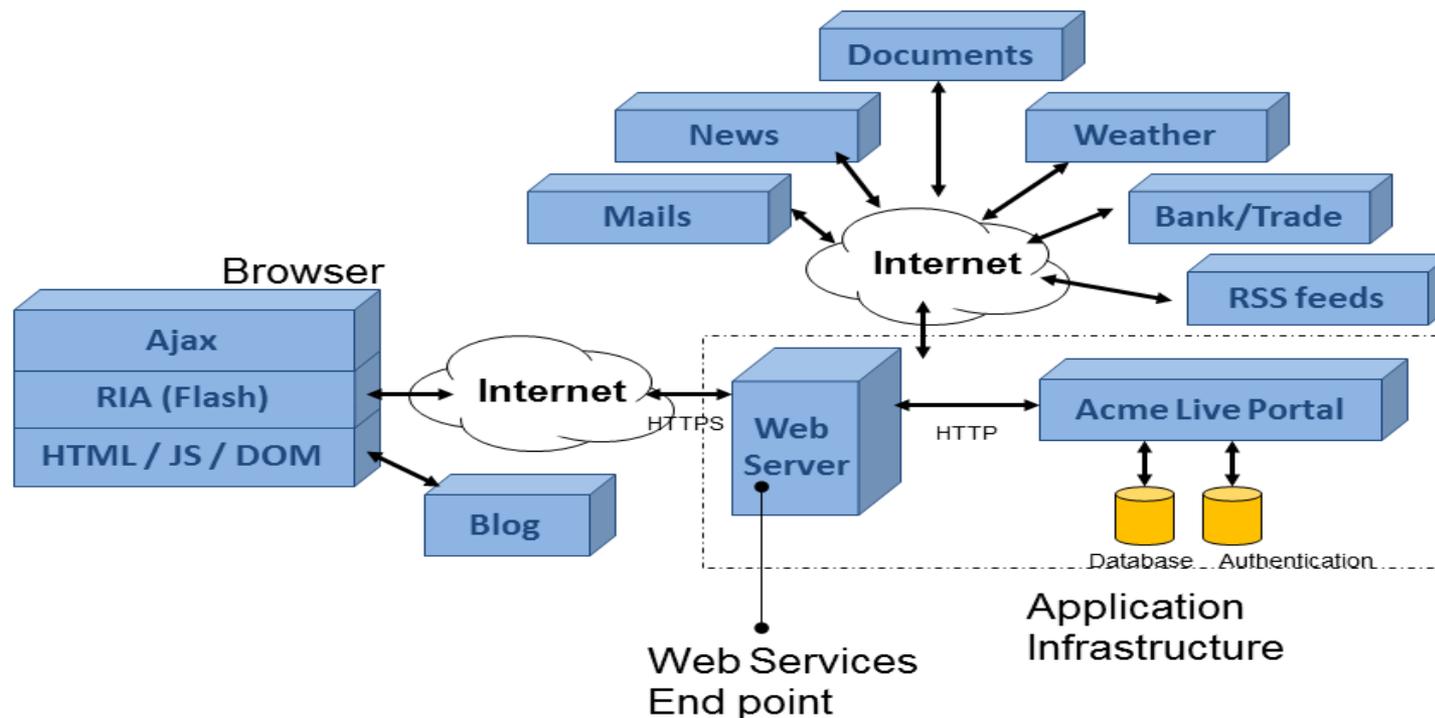


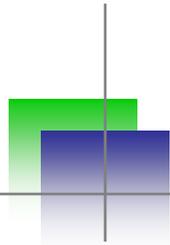
Technology Trend



Modern application architecture

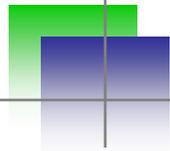
Acme Live Portal Architecture





Developer's challenge

- Rapid development
- Catching up with new technologies – Web 2.0, HTML5, HTTP 2.0, Mobile, Cloud and on on on on on...
- Tight deadlines
- Frequent release cycles
- Business requirement
 - Reading more than 500 pages of generic security requirement

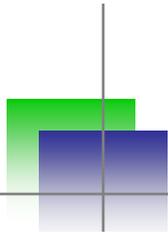


Secure Coding Guideline

- If you google it, you will find “75,900,000” (7.6 million) results
- Why would I need to custom – *I was told in class V that one should not re-invent the wheel.*
- Does SECURE CODING GUIDELINE bring any value??? Is it required??? Or should I have one as I have mention it in my ADLC???
- **Compliance does not need it.**

Typical Secure coding guideline

- Typical secure coding guideline would have –
 - Control Name
 - Control Description
 - Recommendation
 - Link to standards – OWASP & SANS



Sample control

Control Name – Cross Site Scripting

Control Description

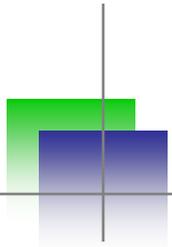
A Cross Site Scripting (XSS) attack occurs when an attacker uses a web application to inject malicious code in the form of a client side script – arbitrary JavaScript – to an end user. As a result of the attack being successful, this attacker-controlled content is executed in the context of the current user. A potential session theft may occur if the user is logged in. If the user is not logged in, the attacker can retain the session cookie until it can be reasonably assumed that the user has logged in. The attacker gains full control over the victim's account. Because the application assumes the script originated from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the client and used with that site. These scripts can even rewrite the content of the HTML page.

Recommendation –

Consider performing a combination of input filtering and context-appropriate output encoding, such as HTML entity encoding for all the data that could be provided by external data sources like databases, users or web services if that data is included in the application's output. Sanitize all user input. Enforce input validation checks on the server side. Include data type checks, data size and range checks. Check user input against whitelist values, check against other data values for logical accuracy and data integrity based on business rules. An example of a blacklist would be where any input that contains characters that are considered "invalid" for web applications, such as "< > & ' `"; Despite efforts to authoritatively identify all such characters, some characters are likely to be left out. A better way of validating input would be to define a list of allowed characters, also referred to as a whitelist.

OWASP Top 10 Link – https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_%28XSS%29

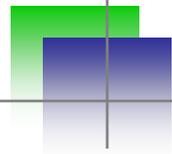
SANS Top 25 - <https://cwe.mitre.org/data/definitions/79.html>



What does this mean?

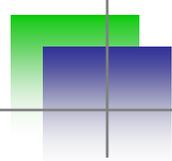


- Giving same pill for all the medical issues
- Can it cure it???



What is missing - Mapping

- Authentication type
- Technology stack
- Libraries/Framework used
- Application functionality
- Web, app and db servers
- Communication method
- Protocol streams

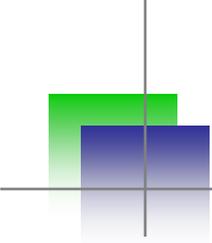


Why generic guideline fails

- We are not helping developers
- Providing generic guideline is **ADDITIONAL** work for developers which no one likes in this world.
- The idea behind generic guideline is to have reference for developers once vulnerabilities are discovered
- Secure coding guideline is **NOT** a reference guide once vulnerabilities are discovered

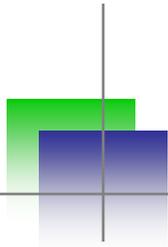
Generating security requirement

- Risk based approach
- Needs to consider lot of parameters –
 - Type of application
 - Authentication type
 - Number and type of users and roles
 - Communication method and protocol stream
 - Application Hosting
 - Data classification
 - Technology, Library and Frameworks used
 - Functionalities



How to generate security requirement

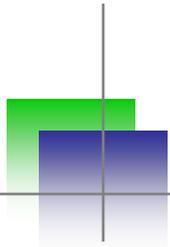
- Based on pre-defined rules
 - Fetch controls which are relevant to the application
 - Provide controls before developer writes a single line of code
 - If we can have business requirement, can we have security requirement
- This seems good idea, let one of my team member do it
 - Consider consistency
 - Tool/Utility can save lot of time
 - Everyone can do it why should we have one designated person to do it???



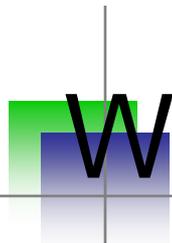
Sample control

Control Name – Cross Site Scripting
Type of application – Web application, Mobile application, web services
Functionality – All
Type of vulnerability – Code Level
Web Server – All
Application server - All
Control Description
A Cross Site Scripting (XSS) attack occurs when an attacker uses a web application to inject malicious code in the form of a client side script – arbitrary JavaScript – to an end user. As a result of the attack being successful, this attacker-controlled content is executed in the context of the current user. A potential session theft may occur if the user is logged in. If the user is not logged in, the attacker can retain the session cookie until it can be reasonably assumed that the user has logged in. The attacker gains full control over the victim’s account. Because the application assumes the script originated from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the client and used with that site. These scripts can even rewrite the content of the HTML page.
Technology – C#
Library – All
Framework - All
Sample Code –
<pre>string encodedinput = Server.HtmlEncode(rawInput);</pre>
How to import library – Security library is named “SecurityLibrary”, following code helps you import class.
Using SecurityLibrary

Technology – Java
Sample Code –
Import commons-lang
<pre>String encodedinput = StringEscapeUtils.escapeHtml4(rawInput);</pre>
How to import library – Security library is named “SecurityLibrary”, Add library (SecurityLibrary.jar) in classpath when you compile your code.
Import SecurityLibrary
Recommendation –
Consider performing a combination of input filtering and context-appropriate output encoding, such as HTML entity encoding for all the data that could be provided by external data sources like databases, users or web services if that data is included in the application’s output. Sanitize all user input. Enforce input validation checks on the server side. Include data type checks, data size and range checks. Check user input against whitelist values, check against other data values for logical accuracy and data integrity based on business rules. An example of a blacklist would be where any input that contains characters that are considered “invalid” for web applications, such as “<> & ’ ”. Despite efforts to authoritatively identify all such characters, some characters are likely to be left out. A better way of validating input would be to define a list of allowed characters, also referred to as a whitelist.
OWASP Top 10 Link – https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_%28XSS%29
SANS Top 25 - https://cwe.mitre.org/data/definitions/79.html

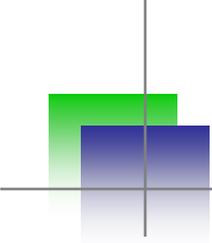


Demo



Why this can get me promotion

- Developers
 - Has list of controls before they even write a single line of a code – **BIGGEST BUYING**
 - Needs to review only 20 pages rather than 500 long pages with all the controls
- Security team
 - Can become input for threat and architecture review, source code review and penetration testing (In a perfect world security as part of ADLC)



Thanks!

Hemil Shah

Co-CEO & Director

+91 99790 55100 or +1 201 203 7008

hemil@blueinfy.net

