ASPECT SECURITY
application security specialists

A Positive Approach to Application Security with OWASP ESAPI

Jeff Williams
Aspect Security CEO
Volunteer Chair of OWASP
jeff.williams@aspectsecurity.com
410-707-1487

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

About Aspect Security

Specialists in Application Security

- Exclusive focus on Application Security since 2002
- Key contributors to OWASP and authors of OWASP Top Ten
- Application security champions in FISMA and Java EE

Assurance Services for Critical Applications

- Millions of lines of code verified per month
- Java, JSP, C/C++, C#, ASP, VB.NET, ABAP, PHP, CFMX, Perl…
- Platforms – J2EE, .NET, SAP, Oracle, PeopleSoft, Struts, …

Acceleration Services for Software, Security, and Management Teams

- Proven application security initiatives
- Integrate key security activities into existing software teams
- Framework and tool tailoring for producing secure code

Application Security Education and Training Curriculum

- Over 180 course offerings per year
- Secure coding for developers (hands-on, language-specific)
- Leaders and managers, testers, architects, threat modeling

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Vulnerabilities and Security Controls

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

10/08

OWASP
The Open Web Application Security Project

About the

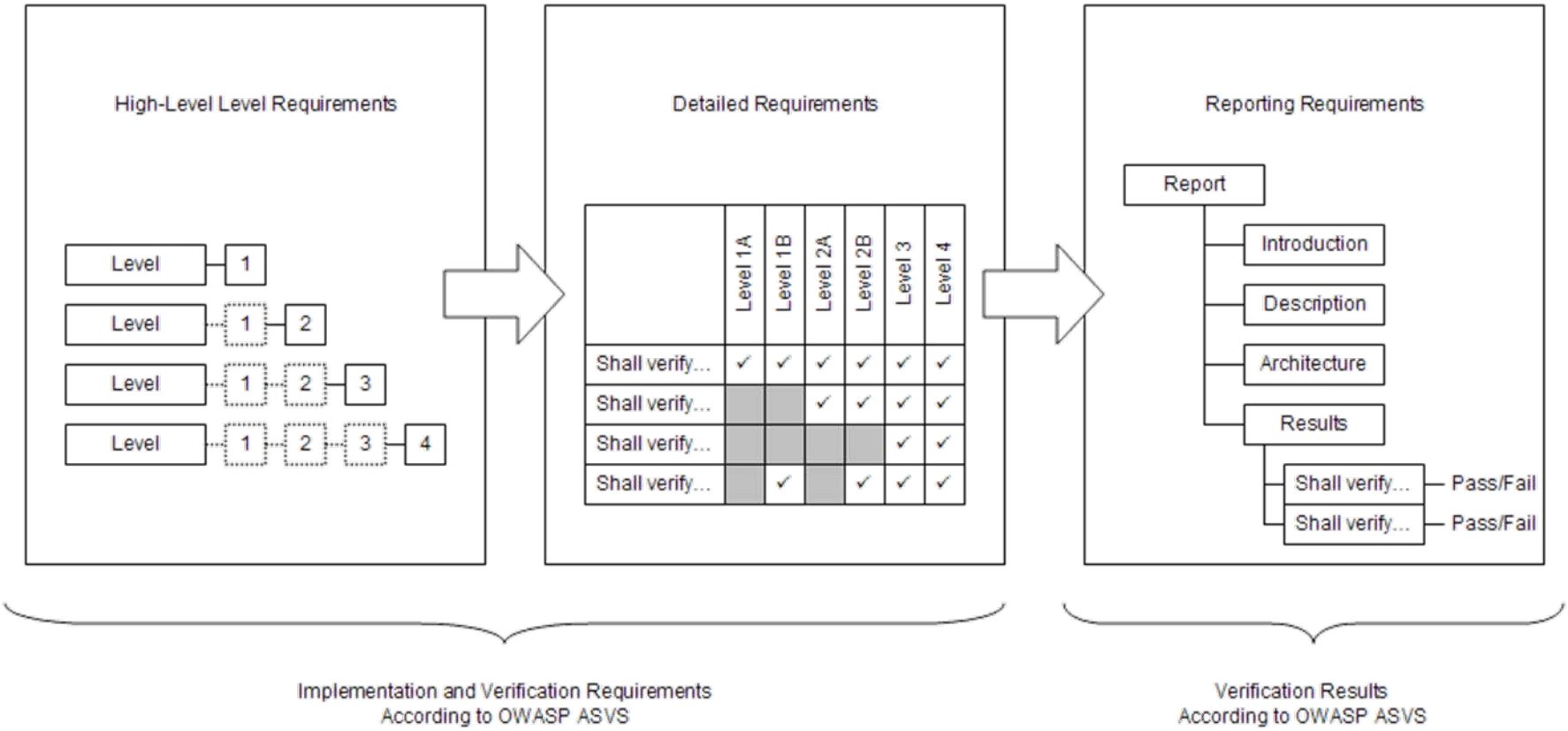
OWASP Application Security Verification Standard 2008

alpha

Creative Commons (CC) Attribution Share-Alike
Free version at http://www.owasp.org

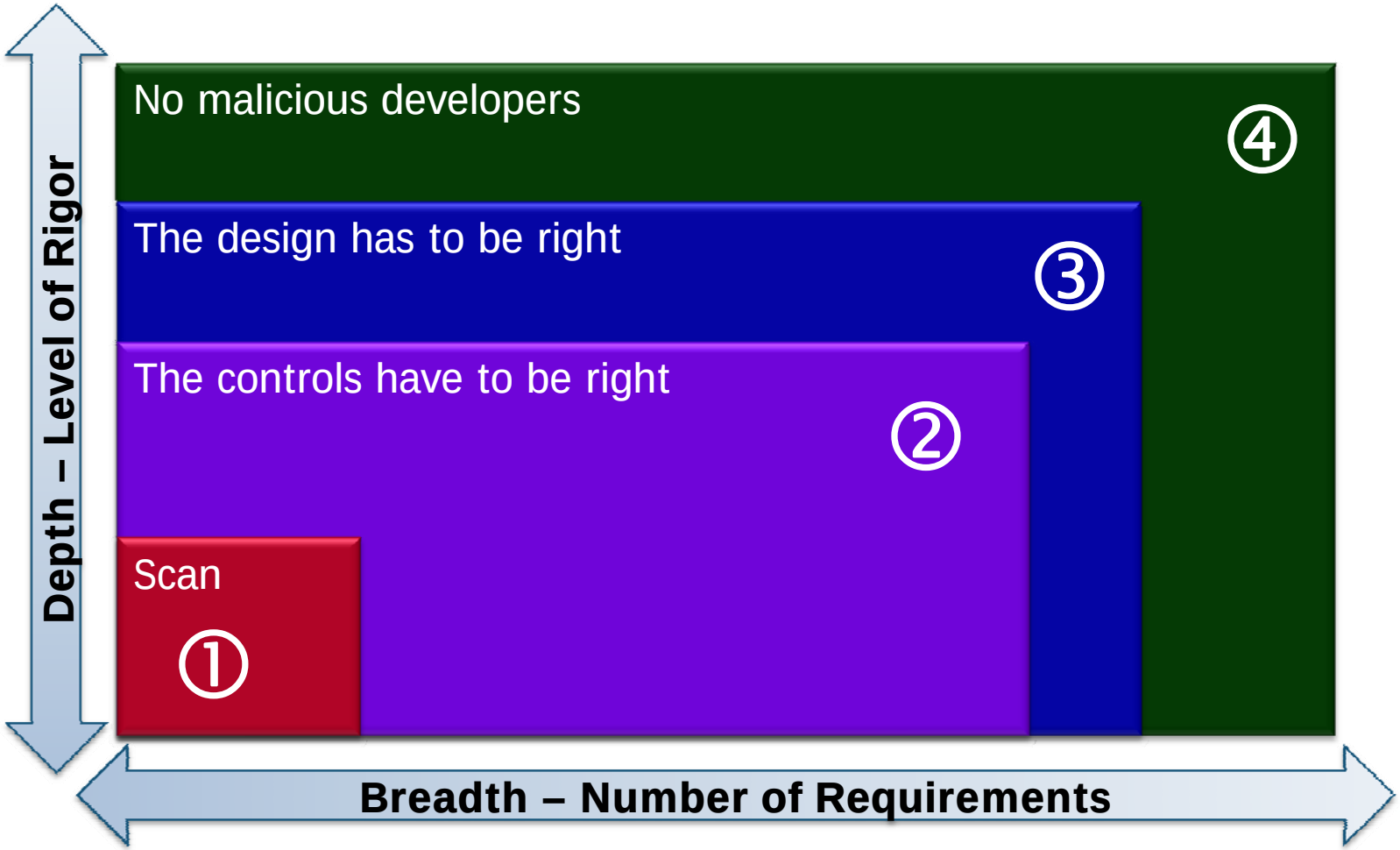
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

What does the ASVS look like?



Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Coverage



Copyright © 2008 – Aspect Security – www.aspectsecurity.com

A positive approach

- **Negative**
  - **The tester shall search for XSS holes**
- **Positive**
  - **<u>The tester shall verify</u> that the application performs input validation and output encoding on all user input**

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

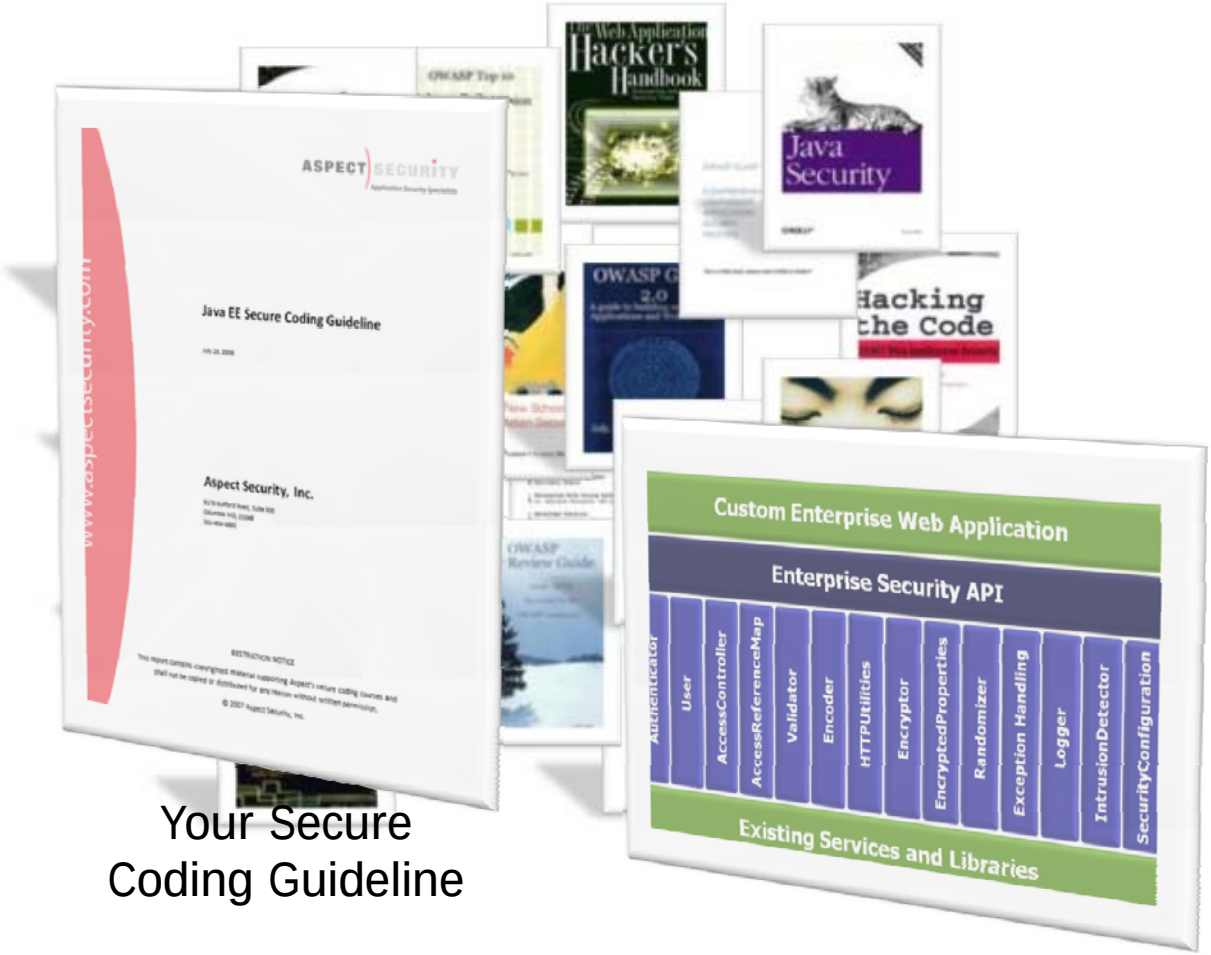
Requirement Summary

Security Area	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V1 – Security Architecture Verification Requirements	2	1	3	3	5	6
V2 – Access Control Verification Requirements	4	3	11	12	12	13
V3 – Authentication Verification Requirements	2	2	11	12	12	13
V4 – Session Management Verification Requirements	5	2	6	6	7	8
V5 – Input Validation Verification Requirements	1		7	8	9	10
V6 – Output Encoding/Escaping Verification Requirements	1	2	8	11	12	13
V7 – Cryptography Verification Requirements			6	9	10	11
V8 – Error Handling and Logging Verification Requirements	1	1	4	6	6	7
V9 – Data Protection Verification Requirements				2	2	2
V10 – Communication Security Verification Requirements	1	1	6	6	7	7
V11 – HTTP Verification Requirements	3	4	5	5	5	5
V12 – Security Configuration Verification Requirements			4	5	6	6
V13 – Malicious Code Search Verification Requirements						4
V14 – Internal Security Verification Requirements					1	3
Total	20	16	71	85	94	108

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Enabling Developers

Hands-on Training

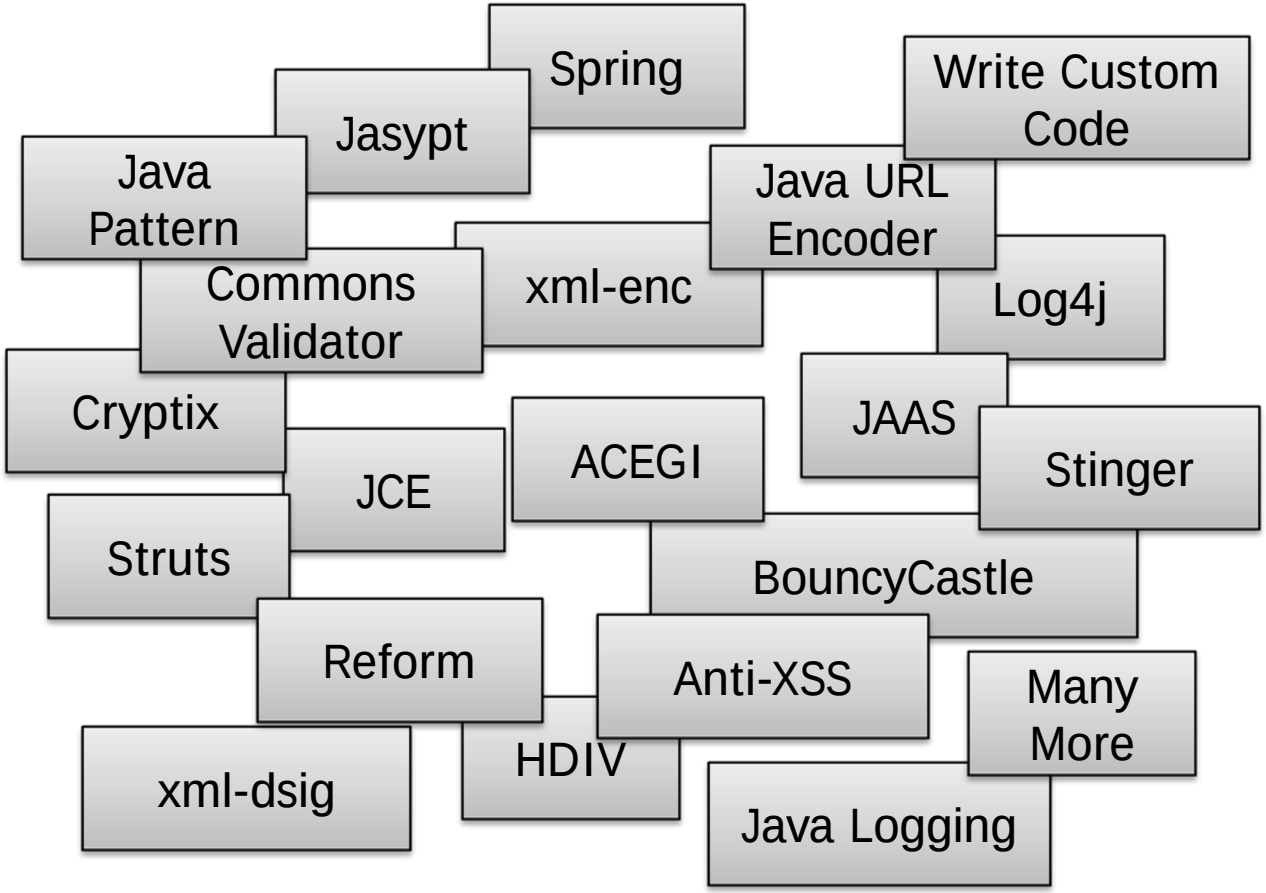


Your Secure Coding Guideline

Your Enterprise Security API

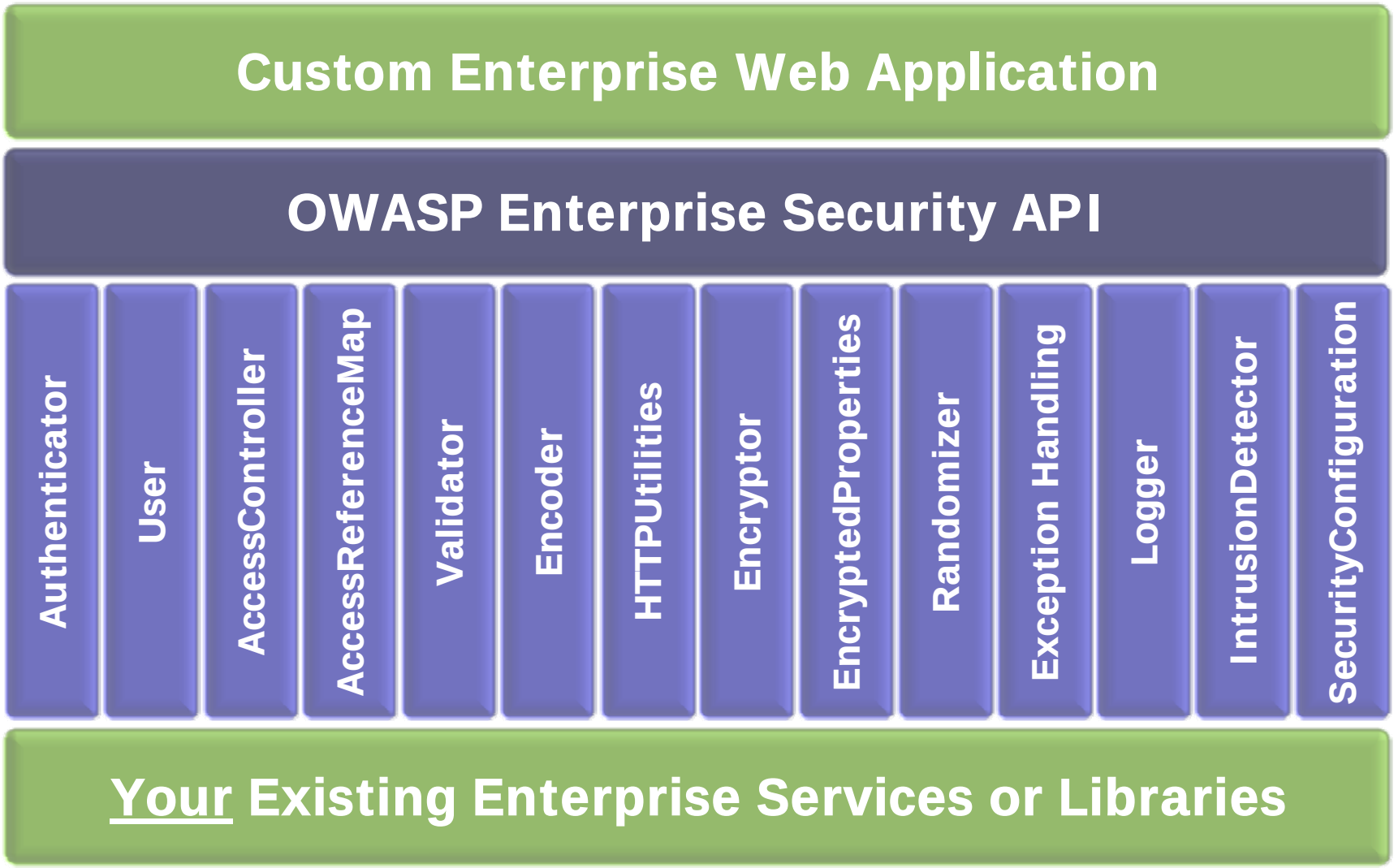
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

The Problem with Security Libraries



Copyright © 2008 – Aspect Security – www.aspectsecurity.com

OWASP ESAPI



Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Some Examples…

1. Input Validation
2. Output Encoding
3. Authentication
4. Access Control
5. Direct Object References
6. Errors, Logging, and Intrusion Detection

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Some Valid Web Encodings for “less-than”

<

// percent encoding
%3c
%3C

// html entity encoding
&#60
&#060
&#0060
&#00060
&#000060
&#0000060
<
<
<
<
<
<
&#x3c
&#x03c
&#x003c
&#x0003c
&#x00003c
&#x000003c
<
<
<
<
<
<
&#X3c
&#X03c
&#X003c
&#X0003c
&#X00003c
&#X000003c
<
<
<
<
<
<
&#x3C
&#x03C
&#x003C
&#x0003C
&#x00003C
&#x000003C
<
<
<
<
<
<
&#X3C
&#X03C
&#X003C
&#X0003C
&#X00003C
&#X000003C
<
<
<
<
<
<
&lt
&lT
&Lt
&LT
<
&lT;
≪
<

// javascript escape syntax
\<
\x3c
\X3c
\u003c
\U003c
\x3C
\X3C
\u003C
\U003C

// css escape syntax
\3c
\03c
\003c
\0003c
\00003c
\3C
\03C
\003C
\0003C
\00003C

Double Encoding

// double encoding examples
< -> < -> &#26;lt&#59 (double entity)
\ -> %5c -> %255c(double percent)
etc...

// double encoding with multiple schemes example
< -> < -> %26lt%3b (first entity, then percent)
< -> %26 -> &#25;26 (first percent, then entity)
etc...

// nested encoding examples
%3c -> %253c (nested encode % with percent)
%3c -> %%33%63 (nested encode percent both nibbles)
%3c -> %%33c (nested encode first nibble with percent)
%3c -> %3%63 (nested encode second nibble with percent)
< -> &&108;t; (nested encode l with entity)
etc...

// nested encoding with multiple schemes examples
< -> &%6ct; (nested encode l with percent)
%3c -> %&x33;c (nested encode 3 with entity)
etc...

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

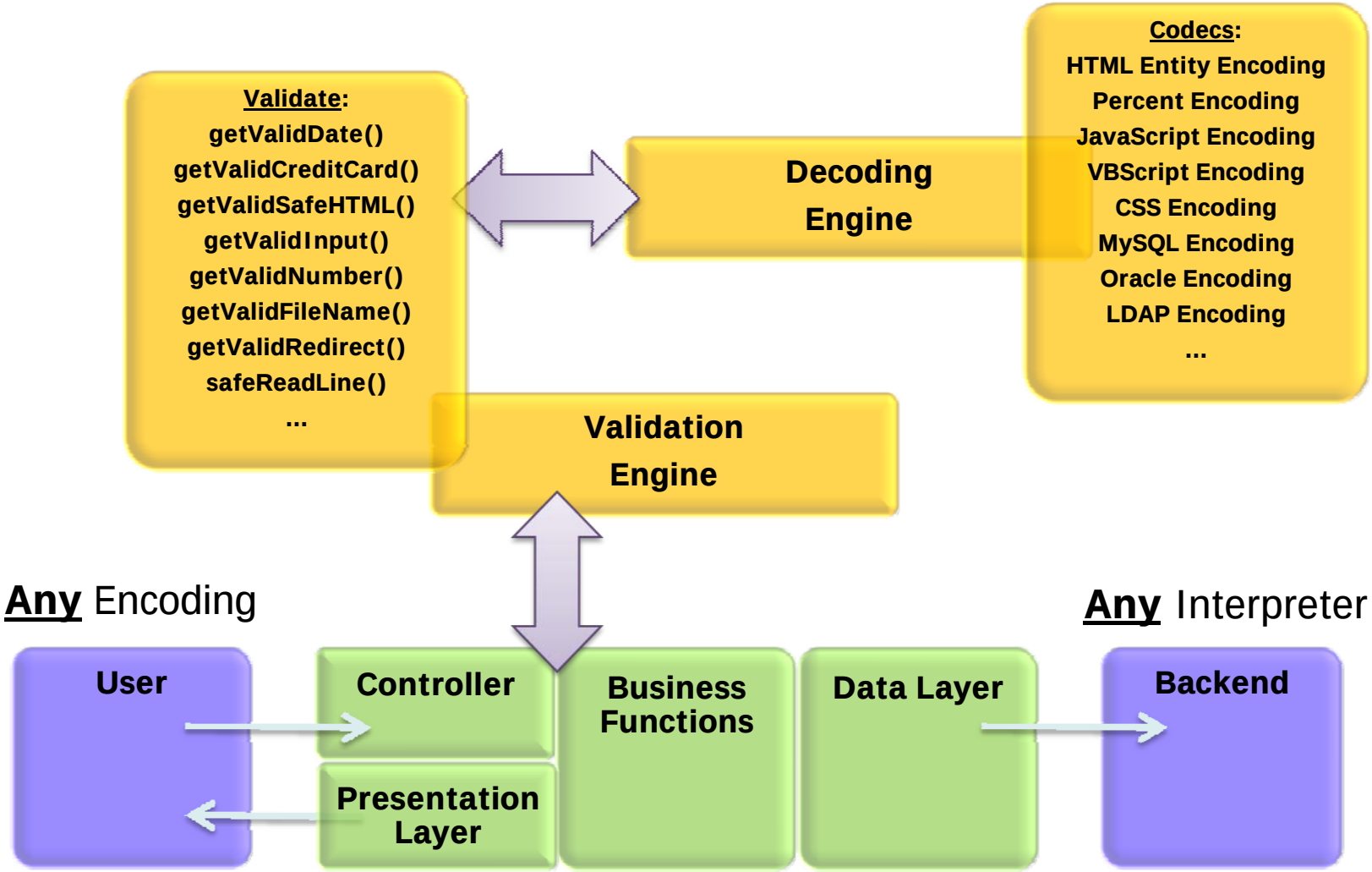
Handling Double Encoding

Original	Decode 1	Partial	Decode 2	Final
%26lt;	HTML Entity	%26lt;	URL Decode	<
%26lt;	URL Decode	<	HTML Entity	<
&#25;3c	HTML Entity	%3c	URL Decode	<
&#25;3c	URL Decode	&#25;3c	HTML Entity	%3c
%2526lt%253B	HTML Entity	%2526lt%253B	URL Decode	%26lt%3B
%2526lt%253B	URL Decode	%26lt%3B	HTML Entity	%26lt%3B
%2526lt%253B	2x URL Decode	<	HTML Entity	<

What components decode what? In what order?

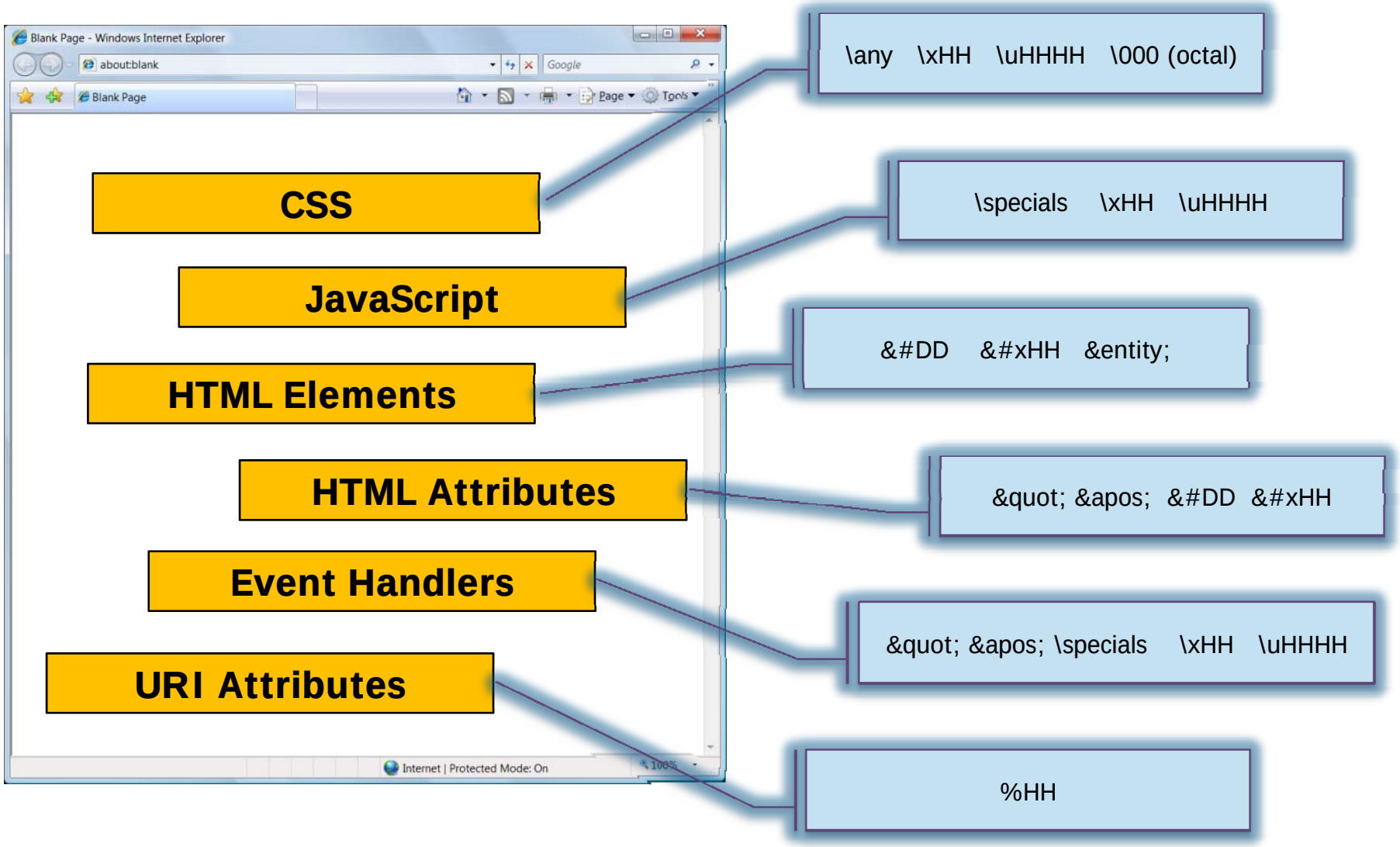
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

1. ESAPI Input Validation



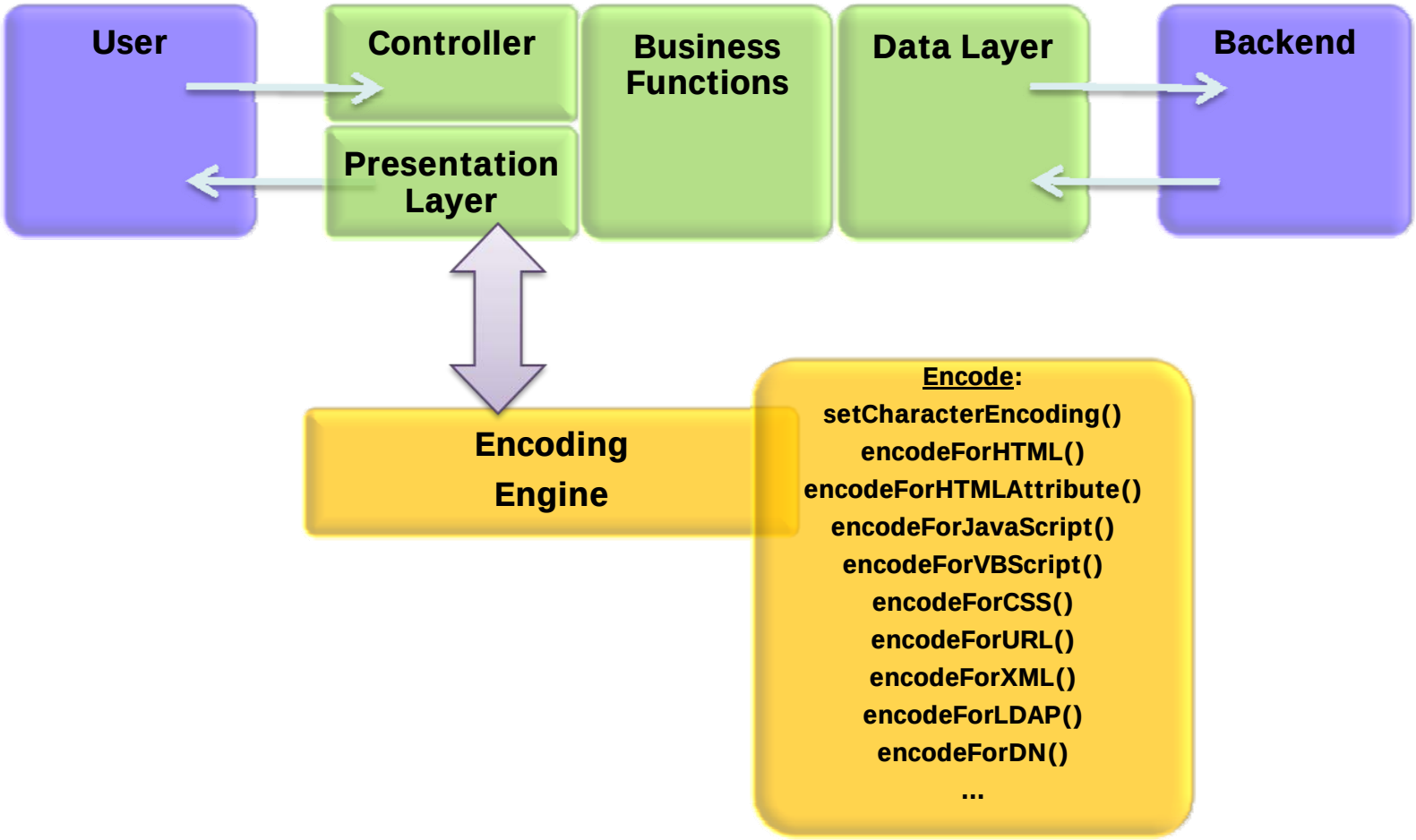
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

HTML Execution Contexts



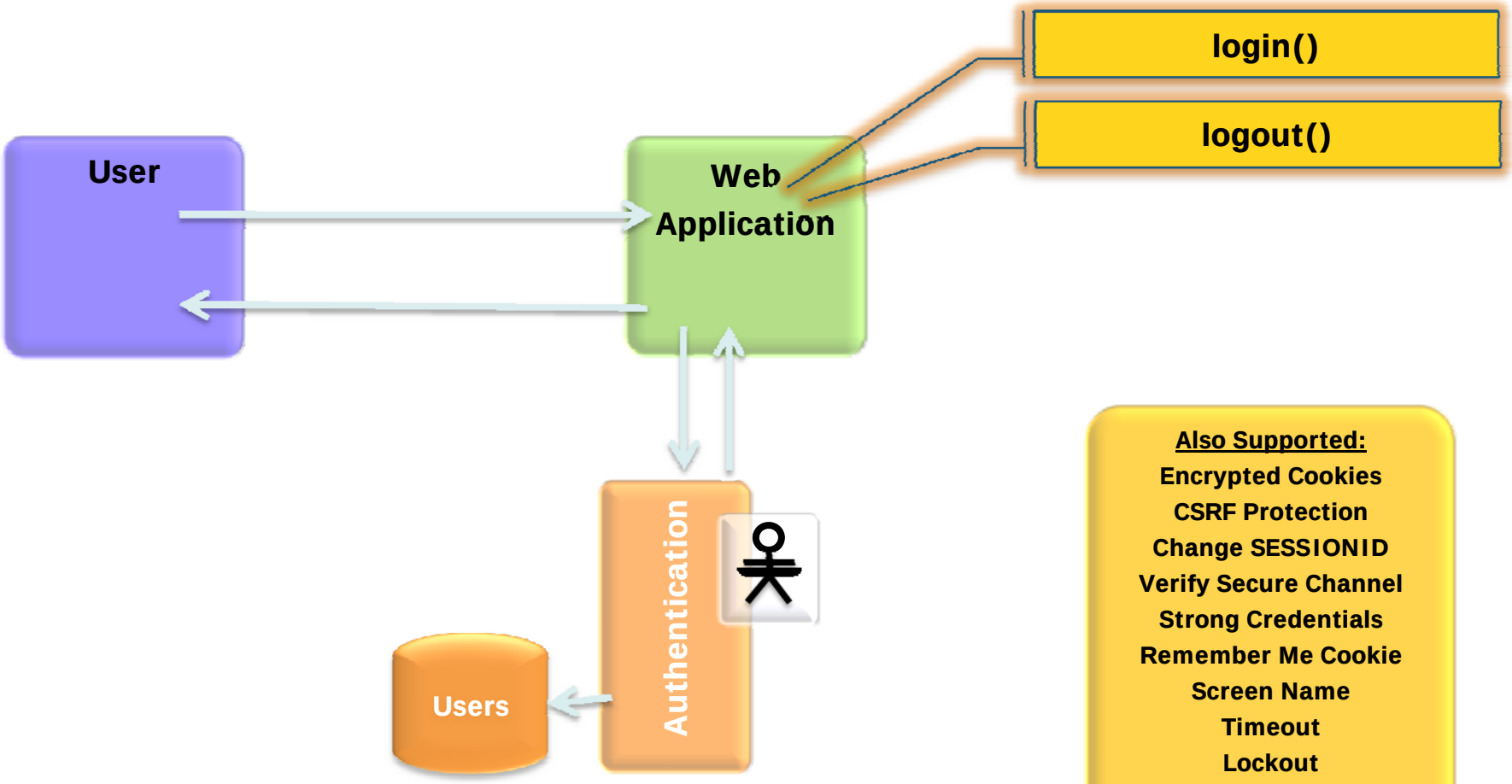
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

2. ESAPI Output Encoding



Copyright © 2008 – Aspect Security – www.aspectsecurity.com

3. ESAPI Authentication

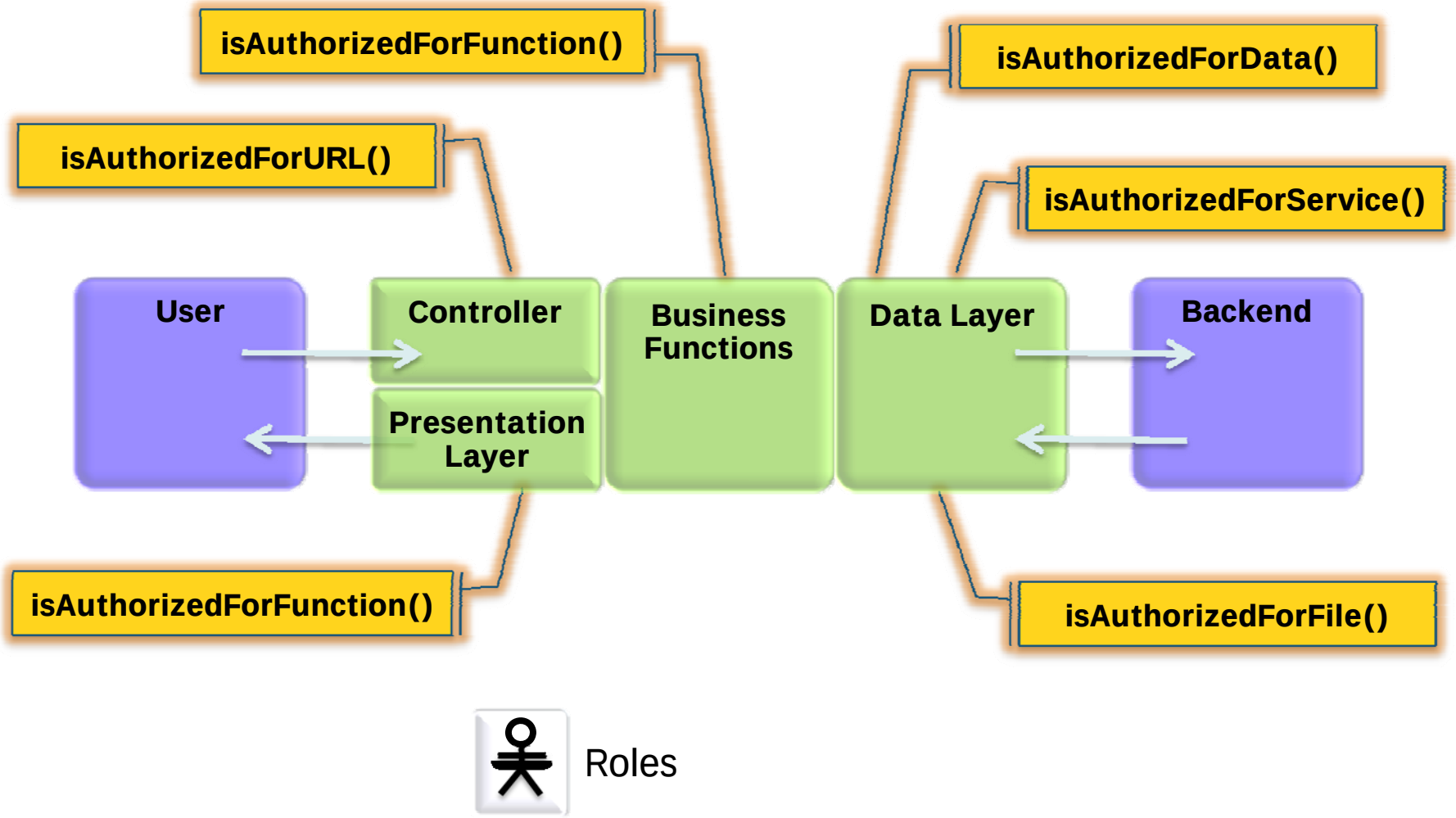


Also Supported:

Encrypted Cookies
CSRF Protection
Change SESSIONID
Verify Secure Channel
Strong Credentials
Remember Me Cookie
Screen Name
Timeout
Lockout
...

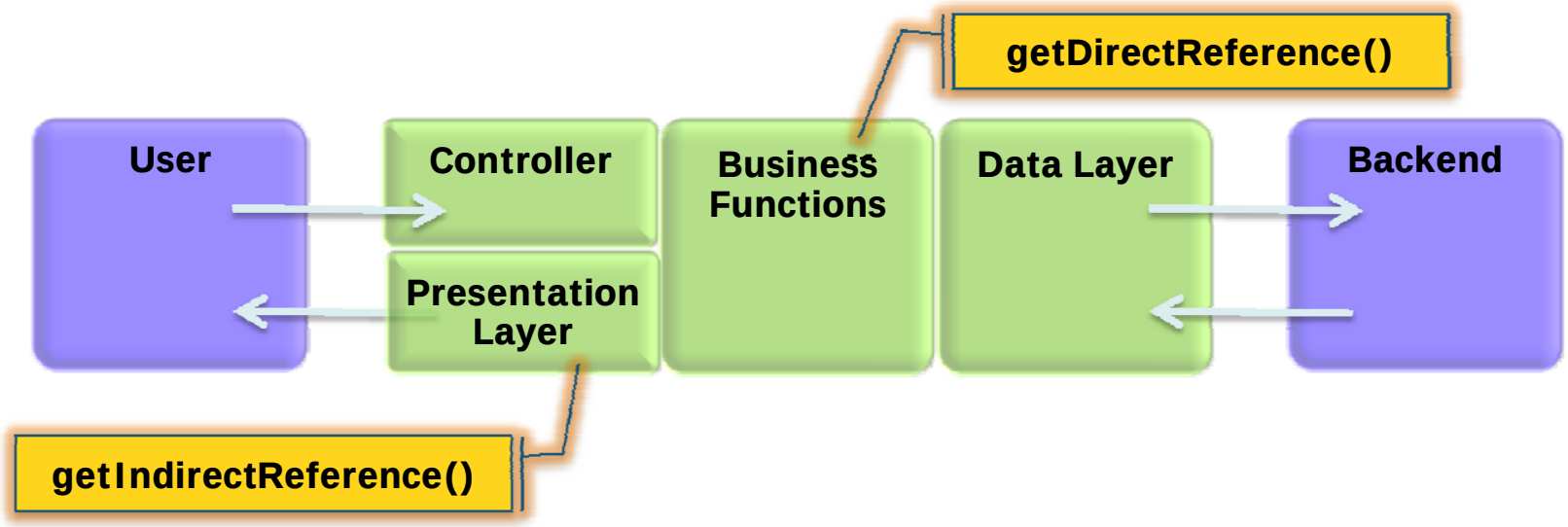
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

4. ESAPI Access Control



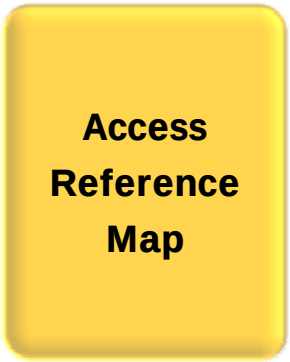
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

5. ESAPI Direct Object Reference Protection



http://app?file=Report123.xls
http://app?file=1

http://app?id=9182374
http://app?id=7d3J93

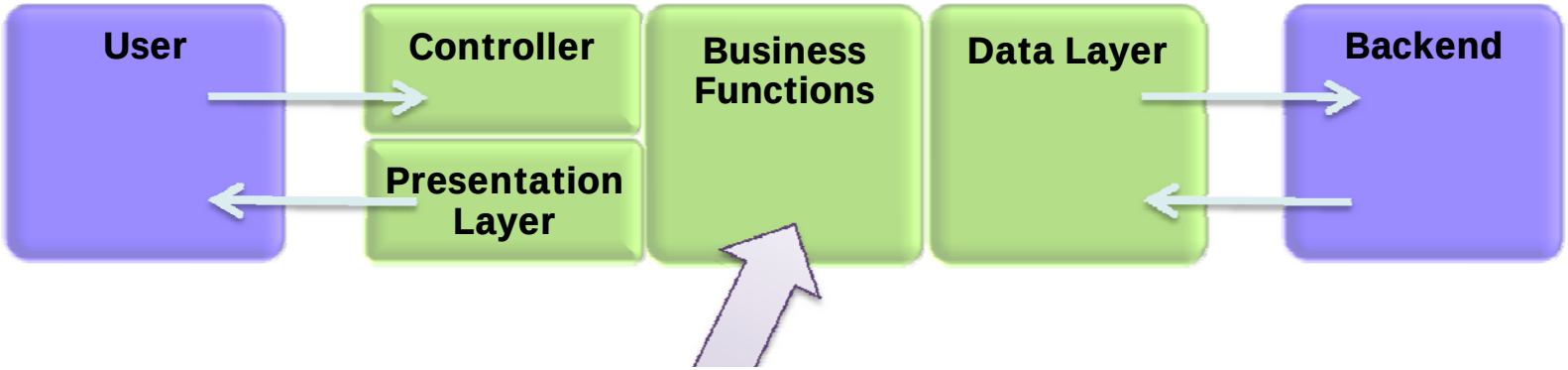


Report123.xls

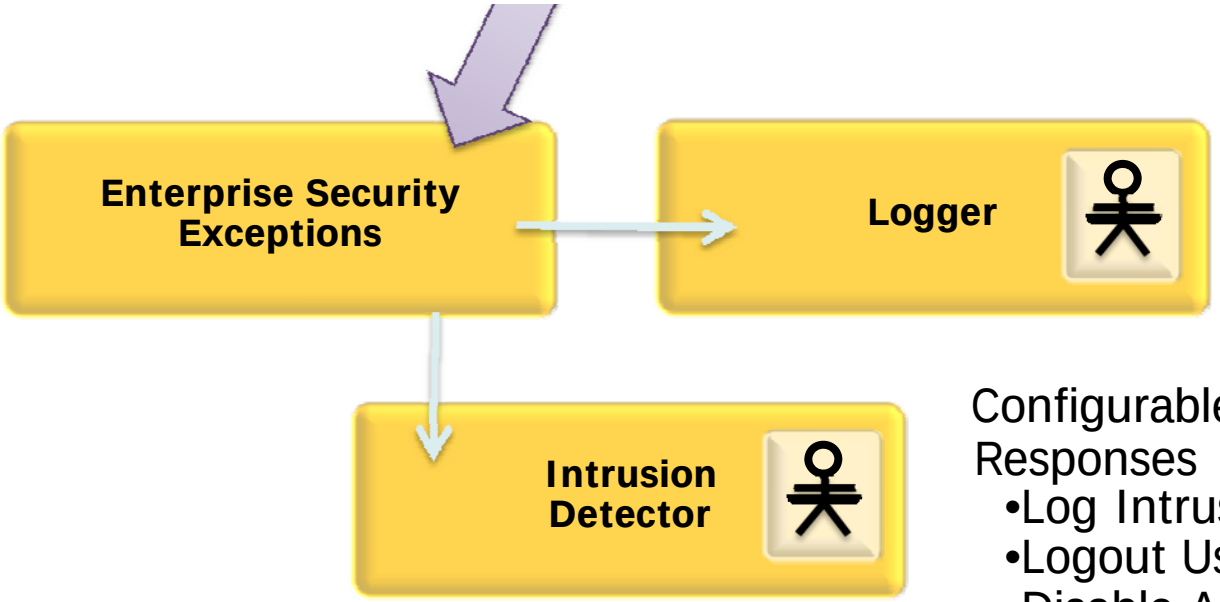
Acct:9182374

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

6. Errors, Logging, and Detection



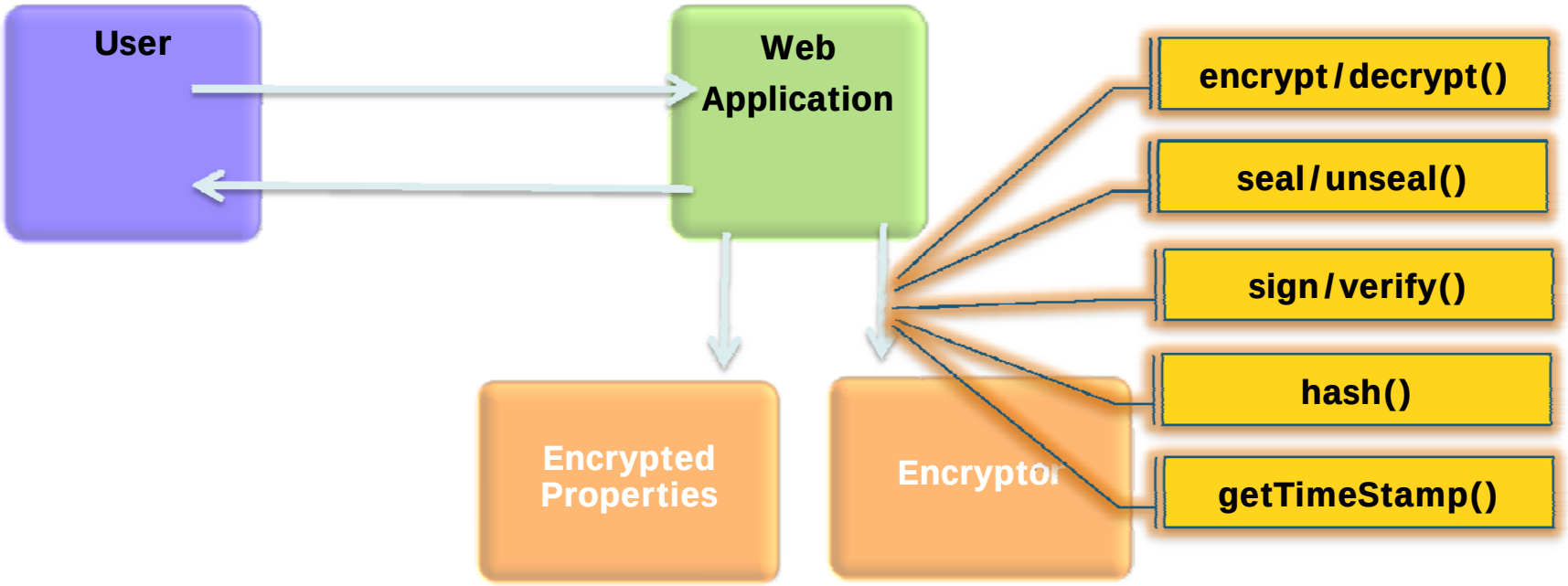
throw new ValidationException("User message", "Log message");



Configurable Thresholds
Responses
- Log Intrusion
- Logout User
- Disable Account

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Data Protection Challenges



Random Tokens and GUIDs

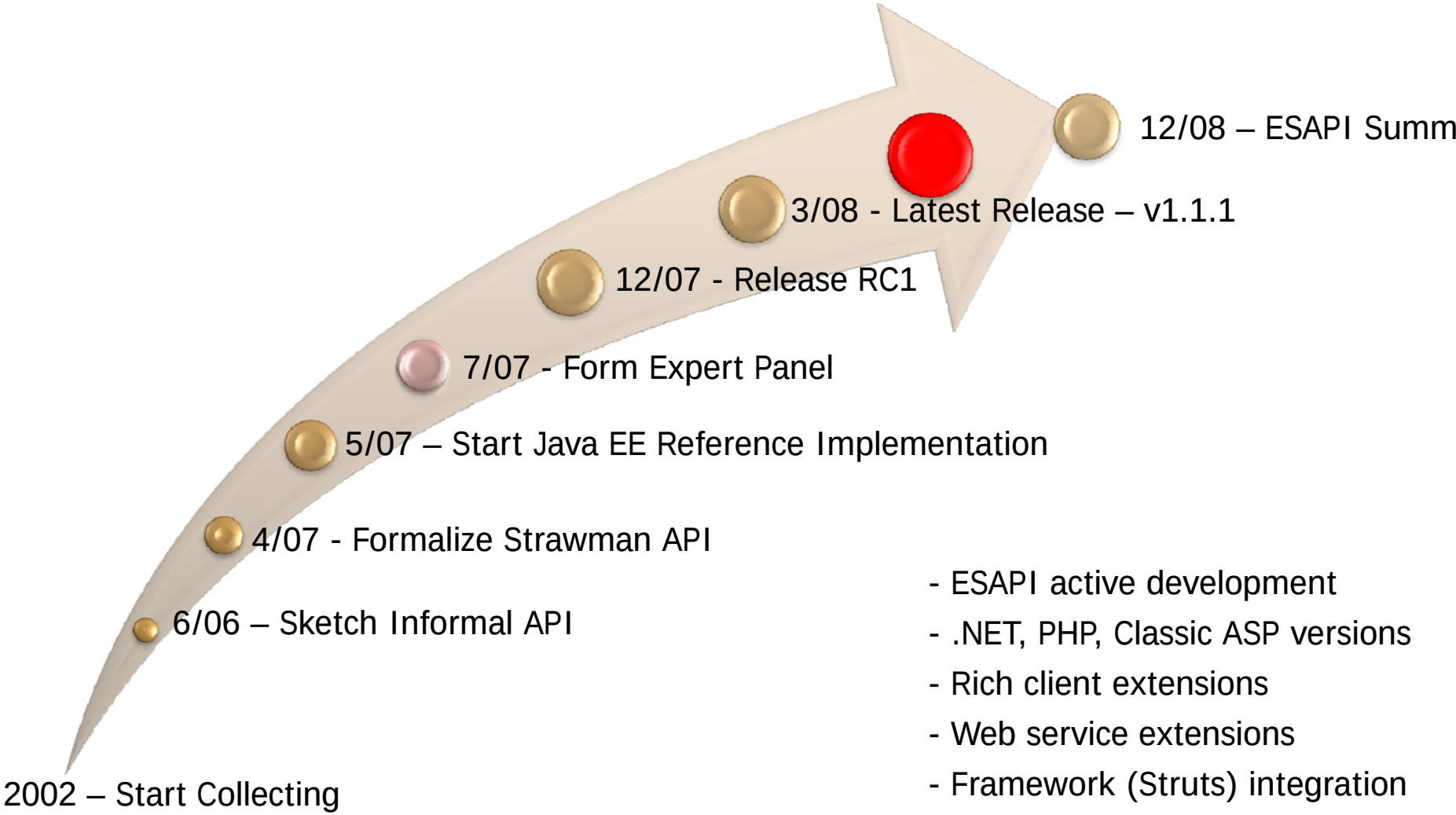
Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Coverage

OWASP Top Ten	OWASP ESAPI
A1. Cross Site Scripting (XSS)	Validator, Encoder
A2. Injection Flaws	Encoder
A3. Malicious File Execution	HTTPUtilities (Safe Upload)
A4. Insecure Direct Object Reference	AccessReferenceMap, AccessController
A5. Cross Site Request Forgery (CSRF)	User (CSRF Token)
A6. Leakage and Improper Error Handling	EnterpriseSecurityException, HTTPUtils
A7. Broken Authentication and Sessions	Authenticator, User, HTTPUtils
A8. Insecure Cryptographic Storage	Encryptor
A9. Insecure Communications	HTTPUtilities (Secure Cookie, Channel)
A10. Failure to Restrict URL Access	AccessController

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Project Plan and Status



- ESAPI active development
- .NET, PHP, Classic ASP versions
- Rich client extensions
- Web service extensions
- Framework (Struts) integration

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Source Code and Javadoc Online Now!

Project Home Page: http://www.owasp.org/index.php/ESAPI
Code Repository: http://code.google.com/p/owasp-esapi-java

owasp-esapi-java - Google Code - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://code.google.com/p/owasp-esapi-java/source/browse/trunk/src/org/owasp/

```
243     }
244
245     private String generateStrongPassword(String oldPassword) {
246         IRandomizer r = ESAPI.randomizer();
247         String newPassword = "";
248         int limit = 10;
249         for (int i=0; i<limit; i++) {
250             try {
251                 newPassword = r.getRandomString(8, Encoder.CHAR_PASSWORD);
252                 verifyPasswordStrength(newPassword, oldPassword);
253                 return newPassword;
254             } catch (AuthenticationException e) {
255                 logger.logDebug(Logger.SECURITY, "Password generator created weak password: " + newPassword +
256             }
257         }
258         logger.logCritical(Logger.SECURITY, "Strong password generation failed after  " + limit + " attempts")
259         return null;
260     }
261
262     /*
263      * (non-Javadoc)
264      *
265      * @see org.owasp.esapi.interfaces.IAuthenticator#generateStrongPassword(int, char[])
266      */
267     public String generateStrongPassword(String oldPassword, IUser user) {
268         String newPassword = generateStrongPassword(oldPassword);
269         if (newPassword != null)
270             logger.logCritical(Logger.SECURITY, "Generated strong password for " + user.getAccountName());
271         return newPassword;
272     }
273
274     /*
275      * Returns the currently logged user as set by the setCurrentUser() methods. Must not log in this method b
276      * logger calls getCurrentUser() and this could cause a loop.
277      *
278      * @see org.owasp.esapi.interfaces.IAuthenticator#getCurrentUser()
279      */
280     public User getCurrentUser() {
281         if ( currentUser == null ) {
282                 return anonymous;
283         }
284         User user = (User)currentUser.get();
285         if (user == null)
286             user = anonymous;
287         return user;
288     }
289
290     /*
291      * Returns the current HttpServletRequest.
292      *
293      * @see org.owasp.esapi.interfaces.IAuthenticator#getCurrentRequest()
294      */
295     public HttpServletRequest getCurrentRequest() {
296         return (HttpServletRequest)currentRequest.get();
297     }
298
299     public HttpServletResponse getCurrentResponse() {
```

Done Profile1

IEncryptor - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://owasp-esapi-java.googlecode.com/svn/trunk/javadoc/index.html

esapi googlecode

All Classes

Packages
- org.owasp.esapi
- org.owasp.esapi.errors
- org.owasp.esapi.filters
- org.owasp.esapi.interfaces

org.owasp.esapi.interfaces

Interfaces
- IAccessController
- IAccessReferenceMap
- IAuthenticator
- IEncoder
- IEncryptedProperties
- IEncryptor
- IExecutor
- IHTTPUtilities
- IIntrusionDetector
- ILogger
- IRandomizer
- ISecurityConfiguration
- IUser
- IValidator

Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

org.owasp.esapi.interfaces

Interface IEncryptor

All Known Implementing Classes:
Encryptor

public interface **IEncryptor**

The IEncryptor interface provides a set of methods for performing common encryption, random number, and hashing operations. Implementations should rely on a strong cryptographic implementation, such as JCE or BouncyCastle. Implementors should take care to ensure that they initialize their implementation with a strong "master key", and that they protect this secret as much as possible.

Possible future enhancements (depending on feedback) might include:

- encryptFile

Since:
June 1, 2007

Author:
Jeff Williams (jeff.williams .at. aspectsecurity.com) Aspect Security

Method Summary

java.lang.String	decrypt(java.lang.String ciphertext) Decrypts the provided ciphertext string (encrypted with the encrypt method) and returns a plaintext string.
java.lang.String	encrypt(java.lang.String plaintext) Encrypts the provided plaintext and returns a ciphertext string.
long	getTimeStamp() Gets a timestamp representing the current date and time to be used by other functions in the library.
java.lang.String	hash(java.lang.String plaintext, java.lang.String salt) Returns a string representation of the hash of the provided plaintext and salt.
java.lang.String	seal(java.lang.String data, long timestamp) Creates a seal that binds a set of data and an expiration timestamp.
java.lang.String	sign(java.lang.String data) Create a digital signature for the provided data and return it in a string.

Done Profile1

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Quality

Java - Eclipse SDK

File Edit Navigate Search Project Run Window Help

PMD Java

Problems Javadoc Declaration Search Console Violation... Coverage JUnit Tasks Call Hiera...

Finished after 1.386 seconds

Runs: 124/124 Errors: 0 Failures: 0

- org.owasp.esapi.LoggerTest
- org.owasp.esapi.AuthenticatorTest
- org.owasp.esapi.UserTest
- org.owasp.esapi.RandomizerTest
- org.owasp.esapi.AccessControllerT
- org.owasp.esapi.HTTPUtilitiesTest
- org.owasp.esapi.ValidatorTest
- org.owasp.esapi.EncryptorTest
  - testHash
  - testEncrypt
  - testDecrypt
  - testSign
  - testVerifySignature
  - testSeal
  - testVerifySeal
  - testGetCertificate
  - testAddCertificate
  - testValidateCertificate

Java EE - C:\Program Files\eclipse\plugins\com.semmle.resources_0.4.0\config\default....

File Edit Navigate Search Project Run Window Help

Java EE

Proble Tasks JUnit Consol Covera Search Quick q Tree re

AllTests (Nov 7, 2007 12:49:36 PM)

Element	Coverage	Covered Instru...	Total Instructio...
ESAPI	87.1 %	14670	16840
test	82.6 %	5044	6106
src	89.7 %	9626	10734
org.owasp.esapi	89.4 %	9341	10449
Threshold.java	52.9 %	27	51
HTTPUtilities.java	56.9 %	325	571
AccessController.java	74.4 %	386	519
SecurityConfiguration.java	77.6 %	350	451
Authenticator.java	78.1 %	882	1130
Encryptor.java	80.5 %	359	446
Randomizer.java	87.3 %	289	331
User.java	93.3 %	1584	1698
IntrusionDetector.java	93.8 %	183	195
Validator.java	95.8 %	835	872
Executor.java	97.7 %	213	218
Encoder.java	98.2 %	3279	3338
AccessReferenceMap.java	100.0 %	121	121
EncryptedProperties.java	100.0 %	205	205
Logger.java	100.0 %	303	303
org.owasp.esapi.errors	100.0 %	285	285

FORTIFY SOFTWARE

JUnit

O'REILLY

pmd DON'T SHOOT THE MESSENGER

UNIVERSITY OF MARYLAND 1856

FindBugs™ because it's easy

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Assurance

ASPECT SECURITY
Application Security Specialists

OWASP
The Open Web Application Security Project
http://www.owasp.org

- **Expert advisory/design/implementation team**
  - Includes security consultants, product vendors, software developers
  - Collectively reviewed over 100 million lines of code
  - Given guidance to static analysis tool vendors
  - Taught over 500 application security classes
  - Minimal and modular design/implementation

- **Tools and Testing**
  - ~600 JUnit test cases (89% coverage)
  - FindBugs, PMD, Ounce, Fortify clean
  - Code review by several Java security experts
  - Penetration test of sample applications
  - Full Javadoc for all functions

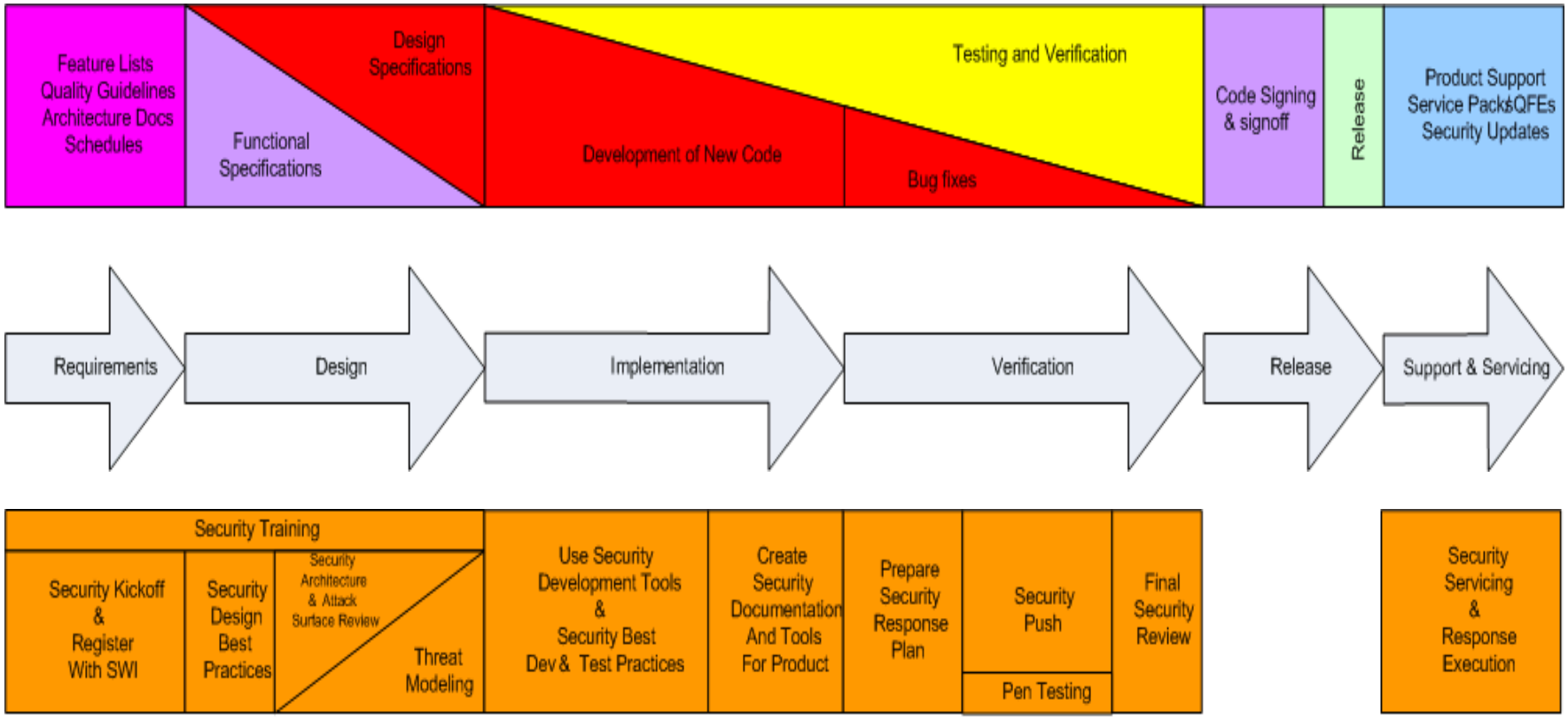
- **Working closely with the Java Servlet Spec team at Sun**
  - They're adopting six new changes to Java EE based on ESAPI

Several major enterprises are using and evaluating ESAPI:

- Sun
- Oracle
- Dept. of Census
- Several Financials

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Microsoft Secure Development Lifecycle (SDL)



- Microsoft achieved a 60% reduction in security flaws

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Small Project Costs to Handle XSS

Cost Area	Typical	With Standard XSS Control
XSS Training	1 days	2 hours
XSS Requirements	2 days	1 hour
XSS Design (Threat Model, Arch Review)	2.5 days	1 hour
XSS Implementation (Build and Use Controls)	7 days	16 hours
XSS Verification (Scan, Code Review, Pen Test)	3 days	12 hours
XSS Remediation	3 days	4.5 hours
Totals	18.5 days	4.5 days

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Potential Enterprise ESAPI Cost Savings

Cost Area	Typical	With ESAPI
AppSec Training (semiannual)	$270K	$135K
AppSec Requirements	250 days ($150K)	50 days ($30K)
AppSec Design (Threat Model, Arch Review)	500 days ($300K)	250 days ($150K)
AppSec Implementation (Build and Use Controls)	1500 days ($900K)	500 days ($300K)
AppSec Verification (Scan, Code Review, Pen Test)	500 days ($300K)	250 days ($150K)
AppSec Remediation	500 days ($300K)	150 days ($90K)
AppSec Standards and Guidelines	100 days ($60K)	20 days ($12K)
AppSec Inventory, Metrics, and Management	250 days ($150K)	200 days ($120K)
Totals	$2.43M	$1.00M

Copyright © 2008 – Aspect Security – www.aspectsecurity.com

Questions and Answers

- **Existing security libraries?**
- **Would this be valuable to your enterprise?**
- **Technical questions?**

Contact Information:
Jeff Williams
Jeff.williams@aspectsecurity.com
Work: 410-707-1487
Main: 301-604-4882

Copyright © 2008 – Aspect Security – www.aspectsecurity.com