Security Code Review

Sherif Koussa
OWASP Ottawa Chapter Leader
Software Secured - Principal
sherif.koussa@owasp.org

OWASP Education Project

Copyright 2007 © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

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

Thursday, 9 May, 13
About Sherif

2007 2009 2011 2013

Principal Consultant @ SoftwareSecured
✓ Security Code Review
✓ Penetration Testing
✓ Secure SDL Integration
✓ Application Security Training
Take Aways
Take Aways

What is Security Code Review
Take Aways

- What is Security Code Review
Take Aways

- What is Security Code Review
- Effective Security Code Review Process
**Take Aways**

- What is Security Code Review
- Effective Security Code Review Process
Take Aways

• What is Security Code Review

• Effective Security Code Review Process

• Key Tools to Use
Take Aways

- What is Security Code Review
- Effective Security Code Review Process
- Key Tools to Use
Take Aways

- What is Security Code Review
- Effective Security Code Review Process
- Key Tools to Use
- Practice Security Code Review
What is this presentation not going to do?
What is this presentation not going to do?

- Ground Breaking Attack\Hack\Black
What is this presentation not going to do?

- Ground Breaking Attack\Hack\Black
What is this presentation not going to do?

- Ground Breaking Attack\Hack\Black
- New Tool
What is this presentation not going to do?

- Ground Breaking Attack\Hack\Black
- New Tool
What is this presentation not going to do?

- Ground Breaking Attack\Hack\Black
- New Tool
- How to Fix Vulnerabilities
What IS Security Code Review?
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
- Integrated Activity into Software Development Lifecycle
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
- Integrated Activity into Software Development Lifecycle
- Cross-Team Integration
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
- Integrated Activity into Software Development Lifecycle
- Cross-Team Integration
  - Development Teams
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
- Integrated Activity into Software Development Lifecycle
- Cross-Team Integration
  - Development Teams
  - Security Teams
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
- Integrated Activity into Software Development Lifecycle
- Cross-Team Integration
  - Development Teams
  - Security Teams
  - Project\Risk Management
What IS Security Code Review?

- The Inspection of Source Code to Find Security Weakness
- Integrated Activity into Software Development Lifecycle
- Cross-Team Integration
  - Development Teams
  - Security Teams
  - Project/Risk Management
- Security Code Review Process
Why Security Code Reviews
Why Security Code Reviews

- Effectiveness of security controls against known threats
- Exercise all application execution paths
- Find all instances of a certain vulnerability
- The only way to find certain types of vulnerabilities
- Effective remediation instructions
What Are We Looking For?
What Are We Looking For?

- Software Weaknesses
  - SQL Injection
  - Cross-site Scripting
  - Insufficient Authentication
What Are We Looking For?

- Software Weaknesses
  - SQL Injection
  - Cross-site Scripting
  - Insufficient Authentication
- Application Logic Issues
  - Application Logic Bypass
What Are We Looking For?

- Software Weaknesses
  - SQL Injection
  - Cross-site Scripting
  - Insufficient Authentication
- Application Logic Issues
  - Application Logic Bypass
- Dead\Debug Code
What Are We Looking For?

- Software Weaknesses
  - SQL Injection
  - Cross-site Scripting
  - Insufficient Authentication
- Application Logic Issues
  - Application Logic Bypass
- Dead\Debug Code
- Misconfiguration Issues
Important Steps For Effective Process
Important Steps For Effective Process

- Reconnaissance
Important Steps For Effective Process

- Reconnaissance
- Threat Assessment
Important Steps For Effective Process

- Reconnaissance
- Threat Assessment
- Automation
**Important Steps For Effective Process**

- Reconnaissance
- Threat Assessment
- Automation
- Manual Review
Important Steps For Effective Process

- Reconnaissance
- Threat Assessment
- Automation
- Manual Review
- Confirmation & PoC
Important Steps For Effective Process

- Reconnaissance
- Threat Assessment
- Automation
- Manual Review
- Confirmation & PoC
- Reporting
RECONNAISSANCE
Reconnaissance
Reconnaissance

- Primary Business Goal of the Application
Reconnaissance

- Primary Business Goal of the Application
- Use Cases\Abuse Cases
Reconnaissance

- Primary Business Goal of the Application
- Use Cases/Abuse Cases
- Different User Roles
Reconnaissance

- Primary Business Goal of the Application
- Use Cases/Abuse Cases
- Different User Roles
- Technology Stack of the Application
Reconnaissance

- Primary Business Goal of the Application
- Use Cases\Abuse Cases
- Different User Roles
- Technology Stack of the Application
- Environment Discovery
Reconnaissance

- Primary Business Goal of the Application
- Use Cases/Abuse Cases
- Different User Roles
- Technology Stack of the Application
- Environment Discovery
- Use the Application
THREAT ASSESSMENT

- Reconnaissance
- Threat Assessment
- Checklists
- Tools
- OWASP Top 10
- Automation
- Manual Review
- Confirmation & PoC
- Reporting
Enumerate Assets

Assets = $$\text{Assets}$$
Enumerate Threats
Enumerate Vulnerabilities
OWASP Top 10
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
Enumerate Vulnerabilities
OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
Enumerate Vulnerabilities
OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
Enumerate Vulnerabilities
OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable Components
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable Components
- A10 Unvalidated Redirects and Forwards
Enumerate Vulnerabilities

OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable Components
- A10 Unvalidated Redirects and Forwards
Enumerate Vulnerabilities
OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable Components
- A10 Unvalidated Redirects and Forwards
Enumerate Vulnerabilities
OWASP Top 10

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable Components
- A10 Unvalidated Redirects and Forwards
AUTOMATION

- Reconnaissance
- Threat Assessment
- Checklists
- OWASP Top 10
- Tools
- Confirmation & PoC
- Manual Review
- Automation
- Reporting

OWASP
Automation

- Static Code Analysis Tools
  - Static Analysis Technologies Evaluation Criteria (SATEC)

- Scripts: DependencyCheck (GitHub)
Automation with PMD

- PMD is a source code analyzer which finds common programming flaws.
- Could be extended to find security flaws
- Download from Sourceforge
Automation with PMD

- PMD is a source code analyzer which finds common programming flaws.
- Could be extended to find security flaws
- Download from Sourceforge
PMD Demo...
Automation with .NET
CAT.NET is a binary code analysis tool that helps identify common variants of certain prevailing vulnerabilities that can give rise to common attack vectors - Microsoft
CAT.NET is a binary code analysis tool that helps identify common variants of certain prevailing vulnerabilities that can give rise to common attack vectors - Microsoft

- Comes with built-in rules:
  - Reflected Cross-Site Scripting
  - SQL Injection
  - XPath Injection
  - LDAP Injection
  - File Canonicalization Issues
  - Command Injection
  - Information Disclosure
CAT.NET is a binary code analysis tool that helps identify common variants of certain prevailing vulnerabilities that can give rise to common attack vectors - Microsoft.

Comes with built-in rules:
- Reflected Cross-Site Scripting
- SQL Injection
- XPath Injection
- LDAP Injection
- File Canonicalization Issues
- Command Injection
- Information Disclosure

Download from MSDN
CAT.NET is a binary code analysis tool that helps identify common variants of certain prevailing vulnerabilities that can give rise to common attack vectors - Microsoft

Comes with built-in rules:

- Reflected Cross-Site Scripting
- SQL Injection
- XPath Injection
- LDAP Injection
- File Canonicalization Issues
- Command Injection
- Information Disclosure

Download from MSDN
CAT.NET Demo...
MANUAL REVIEW

- Reconnaissance
- Threat Modeling
- Checklists
- Tools
- OWASP Top 10
- Confirmation & PoC
- Report
- Automation
- Manual Review
A1. Injection

- Start With Automation
- Database Script (*.sql, *.txt, etc)
- Pay Attention to Patterns & Coding Styles
- Second Order Injection

Manual | Automatic

Thursday, 9 May, 13
private void filter(HttpServletRequest request, HttpServletResponse response)
{
    String input = request.getParameter("input");
    try {
        if (input.toUpperCase().contains("FROM") ||
            input.toUpperCase().contains("SELECT") ||
            input.toUpperCase().contains("UPDATE") ||
            input.toUpperCase().contains("INSERT") ||
            input.toUpperCase().contains("INTO") ||
            input.toUpperCase().contains("WHERE") ||
            input.toUpperCase().contains("ALTER") ||
            input.toUpperCase().contains("SHUTDOWN") ||
            input.toUpperCase().contains("UNION") ||
            input.toUpperCase().contains("DELETE") ||
            input.toUpperCase().contains("CREATE") )
        { response.getOutputStream().println("Please provide a permitted value.");
        }
    }
}
private void filter(HttpServletRequest request, HttpServletResponse response) {
    String input = request.getParameter("input");
    try {
        if (input.toUpperCase().contains("FROM ")) ||
            input.toUpperCase().contains("SELECT ") ||
            input.toUpperCase().contains("UPDATE ") ||
            input.toUpperCase().contains("INSERT ") ||
            input.toUpperCase().contains("INTO ") ||
            input.toUpperCase().contains("WHERE ") ||
            input.toUpperCase().contains("ALTER ") ||
            input.toUpperCase().contains("SHUTDOWN ") ||
            input.toUpperCase().contains("UNION ") ||
            input.toUpperCase().contains("DELETE ") ||
            input.toUpperCase().contains("CREATE ")
        {
            response.getOutputStream().println("Please provide a permitted value.");
        }
    }
}
A2. Broken Authentication and Session Management

- Authentication Process
- Password Storage
- Password Reset/Changes
- Session Generation
- Session Timeout
- Cookie Domain/Path
public String SHAEncrypt(String inString) {
    StringBuffer sb = new StringBuffer();
    try {
        MessageDigest algorithm = null;
        try {
            algorithm = MessageDigest.getInstance("SHA-1");
        }
        catch (NoSuchAlgorithmException e) {
            logger.exception(e.getMessage(), e);
        }
        algorithm.reset();
        byte[] buf = new byte[inString.length()];
        buf = inString.getBytes();
        algorithm.update(buf);
        byte[] digest = algorithm.digest();
        for (int i = 0; i < digest.length; i++) {
            sb.append(digest[i]);
        }
        catch (Exception e) {
            logger.exception(e.getMessage(), e);
        }
    }
    return sb.toString();
}
public String SHAEncrypt(String inString) {
    StringBuffer sb = new StringBuffer();
    try {
        MessageDigest algorithm = null;
        try {
            algorithm = MessageDigest.getInstance("SHA-1");
        }
        catch (NoSuchAlgorithmException e) {
            logger.exception(e.getMessage(), e);
        }
        algorithm.reset();
        byte[] buf = new byte[inString.length()];
        buf = inString.getBytes();
        algorithm.update(buf);
        byte[] digest = algorithm.digest();
        for (int i = 0; i < digest.length; i++) {
            sb.append(digest[i]);
        }
    }
    catch (Exception e) {
        logger.exception(e.getMessage(), e);
    }
    return sb.toString();
}
```java
public String SHAEncrypt(String inString) {
    StringBuffer sb = new StringBuffer();
    try {
        MessageDigest algorithm = null;
        try {
            algorithm = MessageDigest.getInstance("SHA-1");
        } catch (NoSuchAlgorithmException e) {
            logger.exception( e.getMessage(), e );
        }
        algorithm.reset();
        byte[] buf = new byte[inString.length()];
        buf = inString.getBytes();
        algorithm.update(buf);
        byte[] digest = algorithm.digest();
        for (int i = 0; i < digest.length; i++) {
            sb.append(digest[i]);
        }
    } catch (Exception e) {
        logger.exception( e.getMessage() , e );
    }
    return sb.toString();
}
```
```java
public String SHAEncrypt(String inString)
{
    StringBuffer sb = new StringBuffer();
    try
    {
        MessageDigest algorithm = null;
        try
        {
            algorithm = MessageDigest.getInstance("SHA-1");
        }
        catch (NoSuchAlgorithmException e)
        {
            logger.exception( e.getMessage(), e );
        }
        algorithm.reset();
        byte[] buf = new byte[inString.length()];
        buf = inString.getBytes();
        algorithm.update(buf);
        byte[] digest = algorithm.digest();

        for (int i = 0; i < digest.length; i++)
        {
            sb.append(digest[i]);
        }
    }
    catch (Exception e)
    {
        logger.exception( e.getMessage(), e );
    }
    return sb.toString();
}
```
```java
public String SHAEncrypt(String inString)
{
    StringBuffer sb = new StringBuffer();
    try
    {
        MessageDigest algorithm = null;
        try
        {
            algorithm = MessageDigest.getInstance("SHA-1");
        }
        catch (NoSuchAlgorithmException e)
        {
            logger.exception( e.getMessage(), e );
        }
        algorithm.reset();
        byte[] buf = new byte[inString.length()];
        buf = inString.getBytes();
        algorithm.update(buf);
        byte[] digest = algorithm.digest();

        for (int i = 0; i < digest.length; i++)
        {
            sb.append(digest[i]);
        }
    }
    catch (Exception e)
    {
        logger.exception( e.getMessage(), e );
    }
    return sb.toString();
}
```
public String SHAEncrypt(String inString)
{
    StringBuffer sb = new StringBuffer();
    try
    {
        MessageDigest algorithm = null;
        try
        {
            algorithm = MessageDigest.getInstance("SHA-1");
        }
        catch (NoSuchAlgorithmException e)
        {
            logger.exception( e.getMessage(), e );
        }
        algorithm.reset();
        byte[] buf = new byte[inString.length()];
        buf = inString.getBytes();
        algorithm.update(buf);
        byte[] digest = algorithm.digest();
        for (int i = 0; i < digest.length; i++)
        {
            sb.append(digest[i]);
        }
    }
    catch (Exception e)
    {
        logger.exception( e.getMessage(), e );
    }
    return sb.toString();
}
A3. Cross-Site Scripting

- Inspect application’s defenses
- Contextual HTML output encoding
- Tags with no output encoding
- DOM-Based Cross-site Scripting
- HttpOnly Flag on Cookies.
Quiz-O-Code

```javascript
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for (var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

// get the feedly app version cookie
var feedlyAppVersion = readCookie("feedlyAppVersion");
var startPath = readCookie("startPath");

document.location.href = "http://www.feedly.com/index.html#required"

else if (startPath != null) {
    eraseCookie("startPath");
    document.location.href = "http://www.feedly.com/home#" + startPath;
}
else {
    var baseURL = "http://s3.feedly.com/production" + feedlyAppVersion + "/";
    document.write("<base href='" + baseURL + "/'>");
}
```
Quiz-O-Code

```javascript
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for (var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

//get the feedly app version cookie
var feedlyAppVersion = readCookie( "feedlyAppVersion" );
var startPath = readCookie( "startPath" );

document.location.href = "http://www.feedly.com/index.html#required"

else if (startPath !== null) {
    eraseCookie( "startPath" );
    document.location.href = "http://www.feedly.com/home#" + startPath;
} else {
    var baseURL = "http://s3.feedly.com/production" + feedlyAppVersion + "/";
    document.write("<base href='" + baseURL + "/'>");
```
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';;');
    for (var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

// get the feedly app version cookie
var feedlyAppVersion = readCookie( "feedlyAppVersion" );
var startPath = readCookie( "startPath" );

document.location.href = "http://www.feedly.com/index.html#required"
else if (startPath != null)
{
    eraseCookie( "startPath" );
document.location.href = "http://www.feedly.com/home#" + startPath;
}
else
{
    var baseURL = "http://s3.feedly.com/production" + feedlyAppVersion + "/";
document.write("<base href='" + baseURL + "/'>");
Quiz-O-Code

```javascript
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';;');
    for (var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

//get the feedly app version cookie
var feedlyAppVersion = readCookie( "feedlyAppVersion" );
var startPath = readCookie( "startPath" );

document.location.href = "http://www.feedly.com/index.html#required"
else if (startPath != null) {
    eraseCookie( "startPath" );
document.location.href = "http://www.feedly.com/home#" + startPath;
}
else {
    var baseURL = "http://s3.feedly.com/production" + feedlyAppVersion + "/";
document.write("<base href='" + baseURL + "'/");
```
```javascript
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';;');
    for (var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

// get the feedly app version cookie
var feedlyAppVersion = readCookie( "feedlyAppVersion" );
var startDate = readCookie( "startDate" );

document.location.href = "http://www.feedly.com/index.html#required"
else if (startDate != null) {
    eraseCookie( "startDate" );
    document.location.href = "http://www.feedly.com/home#" + startDate;
}
else {
    var baseURL = "http://s3.feedly.com/production" + feedlyAppVersion + "/";
    document.write("<base href='" + baseURL + "/'>");
}
```
CONFIRMATION & POC

Reconnaissance

Reporting

Threat Modeling

Checklists

OWASP Top 10

Tools

Confirmation & PoC

Automation

Manual Review

Thursday, 9 May, 13
Confirmation & PoC
public static string LookupUsername(String userID)
{
    string userName = "";
    try
    {
        string getUserName = "SELECT userName FROM Users WHERE userID = {0}";

        logTransitionOperation("LookupUsername", userID);
        using (SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ssbcon"].ConnectionString))
        {
            conn.Open();
            getUserName = String.Format(getUserName, userID);
            SqlCommand command = new SqlCommand(getUserName, conn);
            SqlDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                userName = reader.GetString(0);
            }
        }
    }
}
```csharp
public static string LookupUsername(String userID)
{
    string userName = "";
    try
    {
        string getUserName = "SELECT userName FROM Users WHERE userID = {0}";
        logTransitionOperation("LookupUserName", userID);
        using (SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ssqlcon"].ConnectionStrings))
        {
            conn.Open();
            getUserName = String.Format(getUserName, userID);
            SqlCommand command = new SqlCommand(getUserName, conn);
            SqlDataReader reader = command.ExecuteReader();
            while (reader.Read())
            {
                userName = (String)reader[0];
            }
        }
    }
    catch
    {
        userName = "Error";
    }
    return userName;
}
```
SQL Injection:

Location: \source\ACMEPortal\updateinfo.aspx.cs:

Description: The code below is build dynamic sql statement using unvalidated data (i.e. name) which can lead to SQL Injection

```
51 SqlDataAdapter myCommand = new SqlDataAdapter(
    52 "SELECT au_lname, au_fname FROM author WHERE 
    au_id = " + 
    53 SSN.Text + ", myConnection);
```

Priority: High


Owner: John Smith
Checklists
A bit of history

Aviation: led the modern airplanes evolution after Major Hill’s famous 1934 incident

ICU: usage of checklists brought down infection rates in Michigan by 66%
What Does a Checklist Should Cover?

- Data Validation and Encoding Controls
- Encryption Controls
- Authentication and Authorization Controls
- Session Management
- Exception Handling
- Auditing and Logging
- Security Configurations
Resources to Conduct Your Checklist

- NIST Checklist Project
  http://checklists.nist.gov/

- Mozilla’s Secure Coding QA Checklist
  https://wiki.mozilla.org/WebAppSec/Secure_Coding_QA_Checklist

- Oracle’s Secure Coding Checklist
  http://www.oracle.com/technetwork/java/seccodeguide-139067.html
Full Application Security Code Review
QUESTIONS?

sherif.koussa@owasp.org
sherif@softwaresecured.com
References

- OWASP (www.owasp.org)
- Gotham Digital Science Blog (http://blog.gdssecurity.com/labs/tag/pmd)
- Milad’s Blog (http://miladbr.blogspot.de/2013/04/exploiting-unexploitable-dom-based-xss.html)