Security Code Review

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

OWASP
Education Project

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

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

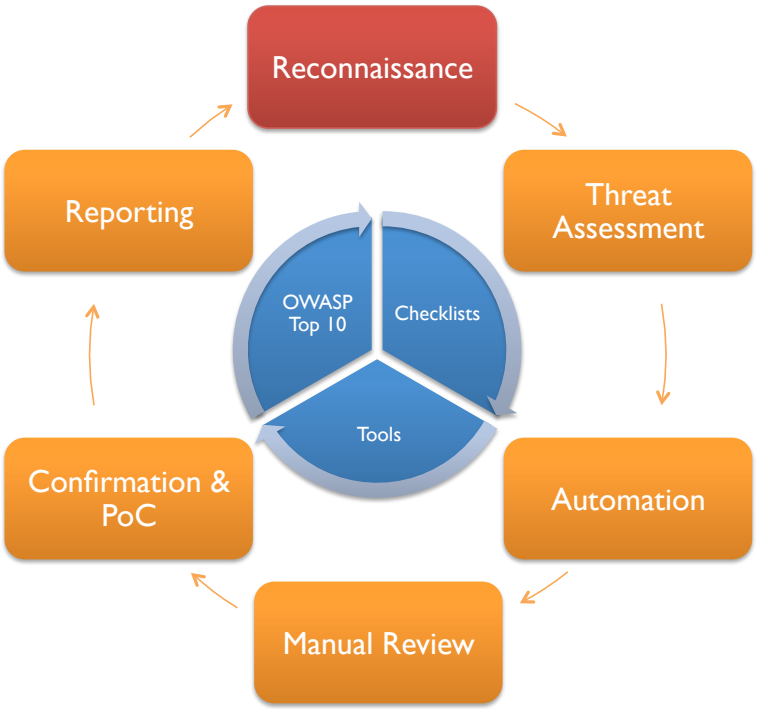
- Reconnaissance
- Threat Assessment
- Automation
- Manual Review
- Confirmation & PoC

Important Steps For Effective Process

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


Reconnaissance
Threat Assessment
Automation
Manual Review
Confirmation & PoC
Reporting
Security Skills
Checklist
Tools

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

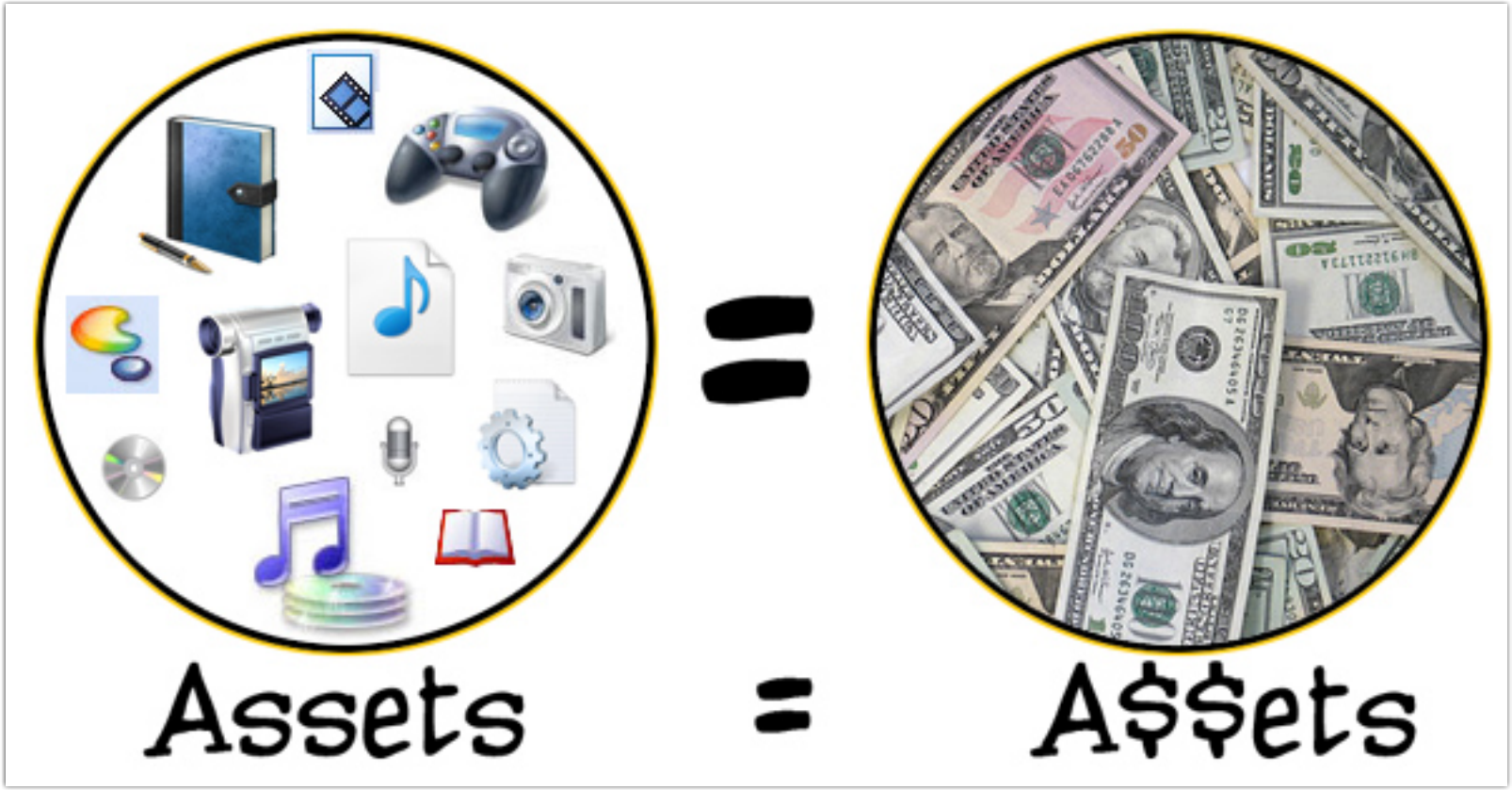
- Primary Business Goal of the Application
- Use Cases\Abuse Cases
- Different User Roles
- Technology Stack of the Application
- Environment Discovery

Reconnaissance

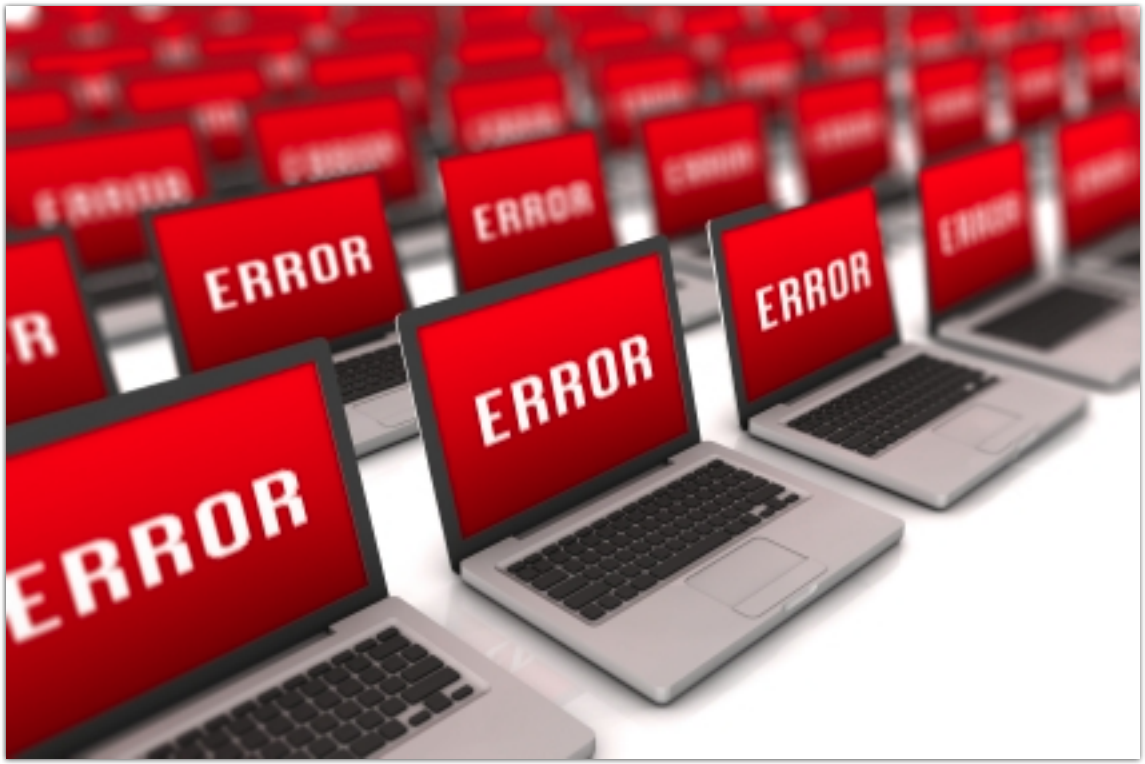
- Primary Business Goal of the Application
- Use Cases\Abuse Cases
- Different User Roles
- Technology Stack of the Application
- Environment Discovery
- Use the Application

THREAT ASSESSMENT

Enumerate 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)

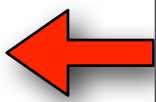
Enumerate Vulnerabilities OWASP Top 10

- 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

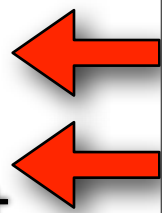
- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Insecure Direct Object References
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Missing Function Level Access Control
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Known Vulnerable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

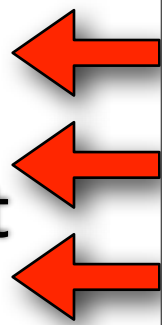
Enumerate Vulnerabilities OWASP Top 10



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

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 analyze which finds common programming flaws.
- Could be extended to find security flaws
- Download from Sourceforge

PMD Demo...

Automation with .NET

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

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

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
- 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

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
- 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

A1. Injection

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

Quiz-O-Code

```
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.");
      }
    }
```

Quiz-O-Code

```
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.");
      }
    }
```

Will it catch "UNI/**/ON", "SEL/**?ECT"?

A2. Broken Authentication and Session Management

- Authentication Process
- Password Storage
- Password Reset\Changes
- Session Generation
- Session Timeout
- Cookie Domain\Path

Quiz-o-code

```
2.  public String SHAEncrypt(String inString)
3.  {
4.          StringBuffer sb = new StringBuffer();
5.          try
6.          {
7.                  MessageDigest algorithm = null;
8.                  try
9.                  {
10.                         algorithm = MessageDigest.getInstance("SHA-1");
11.                 }
12.                 catch (NoSuchAlgorithmException e)
13.                 {
14.                         logger.exception( e.getMessage(), e );
15.                 }
16.                 algorithm.reset();
17.                 byte[] buf = new byte[inString.length()];
18.                 buf = inString.getBytes();
19.                 algorithm.update(buf);
20.                 byte[] digest = algorithm.digest();
21.
22.                 for (int i = 0; i < digest.length; i++)
23.                 {
24.                         sb.append(digest[i]);
25.                 }
26.         }
27.         catch (Exception e)
28.         {
29.                 logger.exception( e.getMessage() , e );
30.         }
31.
32.         return sb.toString();
33. }
```

```
2.  public String SHAEncrypt(String inString)
3.  {
4.          StringBuffer sb = new StringBuffer();
5.          try
6.          {
7.                  MessageDigest algorithm = null;
8.                  try
9.                  {
10.                         algorithm = MessageDigest.getInstance("SHA-1");
11.                 }
12.                 catch (NoSuchAlgorithmException e)
13.                 {
14.                         logger.exception( e.getMessage(), e );
15.                 }
16.                 algorithm.reset();
17.                 byte[] buf = new byte[inString.length()];
18.                 buf = inString.getBytes();
19.                 algorithm.update(buf);
20.                 byte[] digest = algorithm.digest();
21.
22.                 for (int i = 0; i < digest.length; i++)
23.                 {
24.                         sb.append(digest[i]);
25.                 }
26.         }
27.         catch (Exception e)
28.         {
29.                 logger.exception( e.getMessage() , e );
30.         }
31.
32.         return sb.toString();
```

```
2.  public String SHAEncrypt(String inString)
3.  {
4.          StringBuffer sb = new StringBuffer();
5.          try
6.          {
7.                  MessageDigest algorithm = null;
8.                  try
9.                  {
10.                         algorithm = MessageDigest.getInstance("SHA-1");
11.                 }
12.                 catch (NoSuchAlgorithmException e)
13.                 {
14.                         logger.exception( e.getMessage(), e );
15.                 }
16.                 algorithm.reset();
17.                 byte[] buf = new byte[inString.length()];
18.                 buf = inString.getBytes();
19.                 algorithm.update(buf);
20.                 byte[] digest = algorithm.digest();
21.
22.                 for (int i = 0; i < digest.length; i++)
23.                 {
24.                         sb.append(digest[i]);
25.                 }
26.         }
27.         catch (Exception e)
28.         {
29.                 logger.exception( e.getMessage() , e );
30.         }
31.
32.         return sb.toString();
```

Fail-Open Scenario

```
2.  public String SHAEncrypt(String inString)
3.  {
4.          StringBuffer sb = new StringBuffer();
5.          try
6.          {
7.                  MessageDigest algorithm = null;
8.                  try
9.                  {
10.                         algorithm = MessageDigest.getInstance("SHA-1");
11.                 }
12.                 catch (NoSuchAlgorithmException e)
13.                 {
14.                         logger.exception( e.getMessage(), e );
15.                 }
16.                 algorithm.reset();
17.                 byte[] buf = new byte[inString.length()];
18.                 buf = inString.getBytes();
19.                 algorithm.update(buf);
20.                 byte[] digest = algorithm.digest();
21.
22.                 for (int i = 0; i < digest.length; i++)
23.                 {
24.                         sb.append(digest[i]);
25.                 }
26.         }
27.         catch (Exception e)
28.         {
29.                 logger.exception( e.getMessage() , e );
30.         }
31.
32.         return sb.toString();
```

Fail-Open Scenario

```
2.  public String SHAEncrypt(String inString)
3.  {
4.          StringBuffer sb = new StringBuffer();
5.          try
6.          {
7.                  MessageDigest algorithm = null;
8.                  try
9.                  {
10.                         algorithm = MessageDigest.getInstance("SHA-1");
11.                 }
12.                 catch (NoSuchAlgorithmException e)
13.                 {
14.                         logger.exception( e.getMessage(), e );
15.                 }
16.                 algorithm.reset();
17.                 byte[] buf = new byte[inString.length()];
18.                 buf = inString.getBytes();
19.                 algorithm.update(buf);
20.                 byte[] digest = algorithm.digest();
21.
22.                 for (int i = 0; i < digest.length; i++)
23.                 {
24.                         sb.append(digest[i]);
25.                 }
26.         }
27.         catch (Exception e)
28.         {
29.                 logger.exception( e.getMessage() , e );
30.         }
31.
32.         return sb.toString();
```

Fail-Open Scenario

```
2.  public String SHAEncrypt(String inString)
3.  {
4.          StringBuffer sb = new StringBuffer();
5.          try
6.          {
7.                  MessageDigest algorithm = null;
8.                  try
9.                  {
10.                         algorithm = MessageDigest.getInstance("SHA-1");
11.                 }
12.                 catch (NoSuchAlgorithmException e)
13.                 {
14.                         logger.exception( e.getMessage(), e );
15.                 }
16.                 algorithm.reset();
17.                 byte[] buf = new byte[inString.length()];
18.                 buf = inString.getBytes();
19.                 algorithm.update(buf);
20.                 byte[] digest = algorithm.digest();
21.
22.                 for (int i = 0; i < digest.length; i++)
23.                 {
24.                         sb.append(digest[i]);
25.                 }
26.         }
27.         catch (Exception e)
28.         {
29.                 logger.exception( e.getMessage() , e );
30.         }
31.
32.         return sb.toString();
```

Fail-Open Scenario

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

```
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

```
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

```
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

```
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

```
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 + "'/>");
}
```

CONFIRMATION & POC

Confirmation & PoC

Confirmation & PoC

```
public static string LookupUsername(String userID)
{
    string userName = "";
    try
    {
        string getUserName = "SELECT userName FROM Users WHERE userID = {0}";

        logTranstionOperation("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);
            }
        }
    }
```

Confirmation & PoC

```
public static string LookupUsername(String userID)
{
    string userName = "";
    try
    {
        string getUserName = "SELECT userName FROM Users WHERE userID = {0}";

        logTranstionOperation("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
            {
                us
            }
        }
    }
```

```
public static void logTranstionOperation(String name, String userId)
{
    Int64 userIdentifier = Int64.Parse(userId);

    //do something with the user identifier.

}
```

REPORTING


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

Reporting

- Weakness Metadata
- Thorough Description
- Recommendation
- Assign Appropriate Priority

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

Recommendation: Use paramaterized SQL instead of dynamic concatenation, refer to http://msdn.microsoft.com/en-us/library/ff648339.aspx for details.

Owner: John Smith

CHECKLISTS


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

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)
- SQL Injection Attacks and Defenses (http://www.amazon.com/SQL-Injection-Attacks-Defense-Second/dp/1597499633)
- MSDN Blogs (http://dlbmodigital.microsoft.com/ppt/DN-100225-ARevuru-1032438061-FINAL.pdf)