Web Application Report

This report includes important security information about your web application.

Security Report

This report was created by IBM Security AppScan Standard 8.6.0.0, Rules: 1458
Scan started: 7/22/2012 9:12:42 AM
Table of Contents

Introduction

- General Information
- Login Settings

Executive Summary

- Issue Types
- Vulnerable URLs
- Fix Recommendations
- Security Risks
- Causes
- WASC Threat Classification

Issues Sorted by Issue Type

- Authentication Bypass Using SQL Injection 1
- Blind SQL Injection 11
- Cross-Site Scripting 11
- DOM Based Cross-Site Scripting 2
- Poison Null Byte Windows Files Retrieval 2
- Predictable Login Credentials 1
- SQL Injection 12
- Unencrypted Login Request 6
- XPath Injection 1
- Cross-Site Request Forgery 6
- Directory Listing 2
- HTTP Response Splitting 1
- Inadequate Account Lockout 1
- Link Injection (facilitates Cross-Site Request Forgery) 1
- Open Redirect 2
- Phishing Through Frames 2
- Session Identifier Not Updated 1
- Autocomplete HTML Attribute Not Disabled for Password Field 1
- Database Error Pattern Found 1
- Direct Access to Administration Pages 2
- Email Address Pattern Found in Parameter Value 1
- Hidden Directory Detected 2
- Microsoft ASP.NET Debugging Enabled 2
- Missing HttpOnly Attribute in Session Cookie 2
- Permanent Cookie Contains Sensitive Session Information 2
- Unencrypted __VIEWSTATE Parameter 4
- Unsigned __VIEWSTATE Parameter 4
- Application Error 1
- Application Test Script Detected 1
- Email Address Pattern Found 1
- HTML Comments Sensitive Information Disclosure
- Possible Server Path Disclosure Pattern Found
Introduction

This report contains the results of a web application security scan performed by IBM Security AppScan Standard.

High severity issues: 37
Medium severity issues: 25
Low severity issues: 43
Informational severity issues: 25
Total security issues included in the report: 130
Total security issues discovered in the scan: 130

General Information

Scan file name: demo.testfire.net
Scan started: 7/22/2012 9:12:42 AM
Test policy: Default
Host: demo.testfire.net
Operating system: Win32
Web server: IIS
Application server: Any

Login Settings

Login method: Recorded login
Concurrent logins: Enabled
JavaScript execution: Disabled
In-session detection: Enabled
In-session pattern: >Sign Off<
Tracked or session ID cookies:
  ASP.NET_SessionId
  amSessionId
  amUserInfo
  amUserId
  amCreditOffer
Tracked or session ID parameters:
Login sequence:
  http://demo.testfire.net/
  http://demo.testfire.net/bank/login.aspx
  http://demo.testfire.net/bank/login.aspx
  http://demo.testfire.net/bank/main.aspx
Executive Summary

Issue Types

<table>
<thead>
<tr>
<th>Issue Type</th>
<th>Number of Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Bypass Using SQL Injection</td>
<td>1</td>
</tr>
<tr>
<td>Blind SQL Injection</td>
<td>1</td>
</tr>
<tr>
<td>Cross-Site Scripting</td>
<td>11</td>
</tr>
<tr>
<td>DOM Based Cross-Site Scripting</td>
<td>3</td>
</tr>
<tr>
<td>Poison Null Byte Windows Files Retrieval</td>
<td>1</td>
</tr>
<tr>
<td>Predictable Login Credentials</td>
<td>1</td>
</tr>
<tr>
<td>SQL Injection</td>
<td>12</td>
</tr>
<tr>
<td>Unencrypted Login Request</td>
<td>6</td>
</tr>
<tr>
<td>XPath Injection</td>
<td>1</td>
</tr>
<tr>
<td>Cross-Site Request Forgery</td>
<td>6</td>
</tr>
<tr>
<td>Directory Listing</td>
<td>2</td>
</tr>
<tr>
<td>HTTP Response Splitting</td>
<td>1</td>
</tr>
<tr>
<td>Inadequate Account Lockout</td>
<td>1</td>
</tr>
<tr>
<td>Link Injection (facilitates Cross-Site Request Forgery)</td>
<td>6</td>
</tr>
<tr>
<td>Open Redirect</td>
<td>2</td>
</tr>
<tr>
<td>Phishing Through Frames</td>
<td>6</td>
</tr>
<tr>
<td>Session Identifier Not Updated</td>
<td>1</td>
</tr>
<tr>
<td>Autocomplete HTML Attribute Not Disabled for Password Field</td>
<td>4</td>
</tr>
<tr>
<td>Database Error Pattern Found</td>
<td>16</td>
</tr>
<tr>
<td>Direct Access to Administration Pages</td>
<td>2</td>
</tr>
<tr>
<td>Email Address Pattern Found in Parameter Value</td>
<td>2</td>
</tr>
<tr>
<td>Hidden Directory Detected</td>
<td>3</td>
</tr>
<tr>
<td>Microsoft ASP.NET Debugging Enabled</td>
<td>3</td>
</tr>
<tr>
<td>Missing HttpOnly Attribute in Session Cookie</td>
<td>4</td>
</tr>
<tr>
<td>Permanent Cookie Contains Sensitive Session Information</td>
<td>1</td>
</tr>
<tr>
<td>Unencrypted __VIEWSTATE Parameter</td>
<td>4</td>
</tr>
<tr>
<td>Unsigned __VIEWSTATE Parameter</td>
<td>4</td>
</tr>
<tr>
<td>Application Error</td>
<td>15</td>
</tr>
<tr>
<td>Application Test Script Detected</td>
<td>1</td>
</tr>
<tr>
<td>Email Address Pattern Found</td>
<td>3</td>
</tr>
<tr>
<td>HTML Comments Sensitive Information Disclosure</td>
<td>5</td>
</tr>
<tr>
<td>Possible Server Path Disclosure Pattern Found</td>
<td>1</td>
</tr>
</tbody>
</table>

Vulnerable URLs

<table>
<thead>
<tr>
<th>URL</th>
<th>Number of Issues</th>
</tr>
</thead>
</table>

10/2/2012
Fix Recommendations 23

<table>
<thead>
<tr>
<th>Remediation Task</th>
<th>Number of Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always use SSL and POST (body) parameters when sending sensitive information.</td>
<td>6</td>
</tr>
<tr>
<td>Analyze client side code and sanitize its input sources</td>
<td>3</td>
</tr>
<tr>
<td>Change the login credentials to a stronger combination</td>
<td>1</td>
</tr>
<tr>
<td>Ensure that accessed files reside in the virtual path and have certain extensions; remove special characters from user input</td>
<td>1</td>
</tr>
<tr>
<td>Review possible solutions for hazardous character injection</td>
<td>55</td>
</tr>
<tr>
<td>Analyze and harden client side (JavaScript) code.</td>
<td>2</td>
</tr>
<tr>
<td>Decline malicious requests</td>
<td>6</td>
</tr>
<tr>
<td>Do not accept externally created session identifiers</td>
<td>1</td>
</tr>
<tr>
<td>Enforce account lockout after several failed login attempts</td>
<td>1</td>
</tr>
<tr>
<td>Modify the server configuration to deny directory listing, and install the latest security patches available</td>
<td>2</td>
</tr>
<tr>
<td>Add the ‘HttpOnly’ attribute to all session cookies</td>
<td>4</td>
</tr>
<tr>
<td>Risk</td>
<td>Number of Issues</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>H It may be possible to bypass the web application's authentication mechanism</td>
<td>5</td>
</tr>
<tr>
<td>H It is possible to view, modify or delete database entries and tables</td>
<td>29</td>
</tr>
<tr>
<td>H It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</td>
<td>32</td>
</tr>
<tr>
<td>H It is possible to view the contents of any file (for example, databases, user information or configuration files) on the web server (under the permission restrictions of the web server user)</td>
<td>1</td>
</tr>
<tr>
<td>H It might be possible to escalate user privileges and gain administrative permissions over the web application</td>
<td>4</td>
</tr>
<tr>
<td>H It may be possible to steal user login information such as usernames and passwords that are sent unencrypted</td>
<td>6</td>
</tr>
<tr>
<td>H It is possible to access information stored in a sensitive data resource</td>
<td>1</td>
</tr>
<tr>
<td>M It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files</td>
<td>2</td>
</tr>
<tr>
<td>M It is possible to deface the site content through web-cache poisoning</td>
<td>1</td>
</tr>
<tr>
<td>M It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</td>
<td>12</td>
</tr>
<tr>
<td>M It is possible to upload, modify or delete web pages, scripts and files on the web server</td>
<td>6</td>
</tr>
<tr>
<td>M It is possible for an attacker to use the web server to attack other sites, which increases his or her anonymity</td>
<td>2</td>
</tr>
<tr>
<td>L It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations</td>
<td>17</td>
</tr>
<tr>
<td>L It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site</td>
<td>3</td>
</tr>
<tr>
<td>L It may be possible to steal session information (cookies) that was kept on disk as permanent cookies</td>
<td>1</td>
</tr>
<tr>
<td>L It might be possible to undermine application logic</td>
<td>4</td>
</tr>
<tr>
<td>I It is possible to gather sensitive debugging information</td>
<td>15</td>
</tr>
<tr>
<td>I It is possible to download temporary script files, which can expose the application logic and other sensitive information such as usernames and passwords</td>
<td>1</td>
</tr>
<tr>
<td>I It is possible to retrieve the absolute path of the web server installation, which might</td>
<td>1</td>
</tr>
</tbody>
</table>
help an attacker to develop further attacks and to gain information about the file system structure of the web application

Causes

<table>
<thead>
<tr>
<th>Cause</th>
<th>Number of Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sanitation of hazardous characters was not performed correctly on user input</td>
<td>56</td>
</tr>
<tr>
<td>The web application uses client-side logic to create web pages</td>
<td>3</td>
</tr>
<tr>
<td>User input is not checked for the '..' (dot dot) string</td>
<td>1</td>
</tr>
<tr>
<td>Insecure web application programming or configuration</td>
<td>23</td>
</tr>
<tr>
<td>Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted</td>
<td>6</td>
</tr>
<tr>
<td>Insufficient authentication method was used by the application</td>
<td>6</td>
</tr>
<tr>
<td>Directory browsing is enabled</td>
<td>2</td>
</tr>
<tr>
<td>The web application performs a redirection to an external site</td>
<td>2</td>
</tr>
<tr>
<td>The web server or application server are configured in an insecure way</td>
<td>5</td>
</tr>
<tr>
<td>The web application sets session cookies without the HttpOnly attribute</td>
<td>4</td>
</tr>
<tr>
<td>The web application stores sensitive session information in a permanent cookie (on disk)</td>
<td>1</td>
</tr>
<tr>
<td>Proper bounds checking were not performed on incoming parameter values</td>
<td>15</td>
</tr>
<tr>
<td>No validation was done in order to make sure that user input matches the data type expected</td>
<td>15</td>
</tr>
<tr>
<td>Temporary files were left in production environment</td>
<td>1</td>
</tr>
<tr>
<td>Debugging information was left by the programmer in web pages</td>
<td>5</td>
</tr>
<tr>
<td>Latest patches or hotfixes for 3rd. party products were not installed</td>
<td>1</td>
</tr>
</tbody>
</table>

WASC Threat Classification

<table>
<thead>
<tr>
<th>Threat</th>
<th>Number of Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abuse of Functionality</td>
<td>4</td>
</tr>
<tr>
<td>Application Privacy Tests</td>
<td>14</td>
</tr>
<tr>
<td>Application Quality Tests</td>
<td>15</td>
</tr>
<tr>
<td>Brute Force</td>
<td>2</td>
</tr>
<tr>
<td>Content Spoofing</td>
<td>12</td>
</tr>
<tr>
<td>Cross-site Request Forgery</td>
<td>6</td>
</tr>
<tr>
<td>Cross-site Scripting</td>
<td>14</td>
</tr>
<tr>
<td>Directory Indexing</td>
<td>2</td>
</tr>
<tr>
<td>HTTP Response Splitting</td>
<td>1</td>
</tr>
<tr>
<td>Information Leakage</td>
<td>21</td>
</tr>
<tr>
<td>Insufficient Authentication</td>
<td>1</td>
</tr>
<tr>
<td>Insufficient Session Expiration</td>
<td>1</td>
</tr>
<tr>
<td>Null Byte Injection</td>
<td>3</td>
</tr>
<tr>
<td>Predictable Resource Location</td>
<td>1</td>
</tr>
<tr>
<td>Session Fixation</td>
<td>1</td>
</tr>
<tr>
<td>Threat Type</td>
<td>Count</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>SQL Injection</td>
<td>29</td>
</tr>
<tr>
<td>URL Redirector Abuse</td>
<td>2</td>
</tr>
<tr>
<td>XPath Injection</td>
<td>1</td>
</tr>
</tbody>
</table>
Issues Sorted by Issue Type

Authentication Bypass Using SQL Injection

**Severity:** High

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** uid (Parameter)

**Risk:** It may be possible to bypass the web application's authentication mechanism

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because when four types of request were sent - a valid login, an invalid login, an SQL attack, and another invalid login - the responses to the two invalid logins were the same, while the response to the SQL attack seems similar to the response to the valid login.

---

Blind SQL Injection

---

10/2/2012
**Blind SQL Injection**

**Severity:** High

**URL:** http://demo.testfire.net/bank/account.aspx

**Entity:** listAccounts (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because it shows that values can be appended to parameter values, indicating that they were embedded in an SQL query. In this test, three (or sometimes four) requests are sent. The last is logically equal to the original, and the next-to-last is different. Any others are for control purposes. A comparison of the last two responses with the first (the last is similar to it, and the next-to-last is different) indicates that the application is vulnerable.

---

**Cross-Site Scripting**

![Original Response](image1)

![Test Response (last)](image2)

![Original Response](image3)

![Test Response (next-to-last)](image4)

10/2/2012
### Cross-Site Scripting

**Severity:** High  
**URL:** http://demo.testfire.net/search.aspx  
**Entity:** txtSearch (Parameter)  
**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user  
**Causes:** Sanitation of hazardous characters was not performed correctly on user input  
**Fix:** Review possible solutions for hazardous character injection  

**Reasoning:** The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

**Test Response**

![Simulation of the pop-up that appears when this page is opened in a browser](image)

**Raw Test Response:**

```
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 2.0.50727.2)  
```

10/2/2012
Cross-Site Scripting

Severity: **High**

**URL:**  http://demo.testfire.net/survey_complete.aspx

**Entity:**  txtEmail (Parameter)

**Risk:**  It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:**  Sanitation of hazardous characters was not performed correctly on user input

**Fix:**  Review possible solutions for hazardous character injection

**Reasoning:**  The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user’s browser.

**Test Response**
Raw Test Response:

...<li><a id="_ctl0__ctl0_Content_MenuHyperLink18" href="default.aspx?content=inside_careers.htm">Careers</a></li>
</ul>
</td>
<td valign="top" colspan="3" class="bb">
<div style="width: 99%;">
<h1><span id="_ctl0__ctl0_Content_Main_lblTitle">Thanks</span></h1>
<span id="_ctl0__ctl0_Content_Main_lblContent"><p>Thanks for your entry. We will contact you shortly at:<br /><br />
jsmith@demo.testfire.net</p></span>
</div>
</td>
</tr>
</table>
...

Simulation of the pop-up that appears when this page is opened in a browser

The Alloro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express

Raw Test Response:

...<div style="width: 99%;">
<h1><span id="_ctl0__ctl0_Content_Main_lblTitle">Thanks</span></h1>
<span id="_ctl0__ctl0_Content_Main_lblContent"><p>Thanks for your entry. We will contact you shortly at:<br />
jsmith@demo.testfire.net</p></span>
</div>
...
## Cross-Site Scripting

**Severity:** High  
**URL:** http://demo.testfire.net/comment.aspx  
**Entity:** name (Parameter)  
**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.  
**Causes:** Sanitation of hazardous characters was not performed correctly on user input  
**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

**Raw Test Response:**

```html
HTTP/1.1 200 OK  
Date: Sun, 22 Jul 2012 08:04:10 GMT  
Server: Microsoft-IIS/6.0  
X-Powered-By: ASP.NET  
X-AspNet-Version: 2.0.50727  
Cache-Control: no-cache  
Pragma: no-cache  
Expires: -1  
Content-Type: text/html; charset=iso-8859-1  
Content-Length: 7229

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="_ctl0__ctl0_head">
<title>Altoro Mutual: Thank You</title>

Alt<br>
oro Mutual: Thank-You

...  
...  
</td>
</tr>

</table>
```

---

### Issue 4 of 11

10/2/2012
Cross-Site Scripting

Severity: High

URL: http://demo.testfire.net/comment.aspx

Entity: comment.aspx (Page)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response

Raw Test Response:

...
Cross-Site Scripting

Severity: **High**

**URL:** http://demo.testfire.net/subscribe.aspx

**Entity:** txtEmail (Parameter)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

**Test Response**
We recognize that things are always evolving and changing here at Altoro Mutual. Please enter your email below and we will automatically notify you of noteworthy events.

<form action="subscribe.aspx" method="post" name="subscribe" id="subscribe" onsubmit="return confirmEmail(txtEmail.value);">
	<table>
		<tr>
			<td colspan="2">
				<span id="_ctl0__ctl0_Content_Main_message" style="color:Red;font-size:12pt;font-weight:bold;">Thank you. Your email test@altoromutual.com has been accepted.</span>
		
		</td>
		</tr>
		<tr>
			<td>Email: </td>
			<td>
				<input type="text" id="txtEmail" name="txtEmail" value="" style="width: 150px;"/>
			</td>
		</tr>
	</table>

<br/>

Raw Test Response:

...
## Cross-Site Scripting

<table>
<thead>
<tr>
<th><strong>Severity:</strong></th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL:</strong></td>
<td><a href="http://demo.testfire.net/bank/apply.aspx">http://demo.testfire.net/bank/apply.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>amCreditOffer (Cookie)</td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
<td>It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.</td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
<td>Sanitation of hazardous characters was not performed correctly on user input</td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
<td>Review possible solutions for hazardous character injection</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

**Test Response**

```
...<h1>Visa Application</h1>...
...
userid = userCookie.Values["UserID"].ToString();
```
**Cross-Site Scripting**

**Severity:** High

**URL:** http://demo.testfire.net/bank/customize.aspx

**Entity:** customize.aspx (Page)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

---

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

**Test Response**
Cross-Site Scripting

Severity: High

URL: http://demo.testfire.net/bank/login.aspx

Entity: uid (Parameter)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Test Response
Login Failed: We're sorry, but this username was not found in our system. Please try again.

<form action="login.aspx" method="post" name="login" id="login" onsubmit="return (confirminput(login));">
<table>
<tr>
<td>Username:</td>
<td><input type="text" id="uid" name="uid" value="jsmith" onmouseover="alert(144)" style="width: 150px;"> </td>
</tr>
<tr>
<td>Password:</td>
<td><input type="text" id="pwd" name="pwd" value="" style="width: 150px;"> </td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="Login" name="login"></input>
</td>
</tr>
</table>
</form>

The Altoro Mutual website is published by Watchfire, Inc. for the sole purpose of demonstrating the effectiveness of Watchfire products in detecting web application vulnerabilities andwebsite defects. This site is not a real banking site. Similarities, if any, to third-party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied.
Cross-Site Scripting

Severity: High

URL: http://demo.testfire.net/bank/transfer.aspx

Entity: debitAccount (Parameter)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

Raw Test Response:

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:39:53 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1
Content-Length: 9466

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="_ctl0__ctl0_head"><title>Altoro Mutual: Transfer Funds</title></head>
<body>
...
## Cross-Site Scripting

**Severity:** High

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** creditAccount (Parameter)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** Sanitation of hazardous characters was not performed correctly on user input.

**Fix:** Review possible solutions for hazardous character injection.

### Reasoning:
The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user's browser.

### Raw Test Response:

```
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:39:53 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1
Content-Length: 946

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="_ctl0__ctl0_head"><title>Altoro Mutual: Transfer Funds</title></head>
<body>

...<br />

<td colspan="2" align="center">Input type="button" name="transfer" value="Transfer Money" onclick="doTransfer();"
ID="transfer"></td>
</tr>
</table>
</body>
</html>
```

In the response, there is a Syntax error in the query expression 'accountid=1001160141'. This indicates a vulnerability where the application did not properly sanitize user input, allowing for potential malicious script execution.
## Cross-Site Scripting

**Severity:** High

**URL:** http://demo.testfire.net/bank/customize.aspx

**Entity:** lang (Parameter)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** Sanitation of hazardous characters was not performed correctly on user input.

**Fix:** Review possible solutions for hazardous character injection.

**Reasoning:** The test result seems to indicate a vulnerability because Appscan successfully embedded a script in the response, which will be executed when the page loads in the user’s browser.

**Test Response**

![Simulation of the pop-up that appears when this page is opened in a browser](image)

**Raw Test Response:**

```html
...<div class="fl" style="width: 99%;">...
```
Issue 1 of 3

DOM Based Cross-Site Scripting

Severity: High

URL: http://demo.testfire.net/high_yield_investments.htm

Entity: high_yield_investments.htm:101 (Page)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

Causes: The web application uses client-side logic to create web pages.

Fix: Analyze client side code and sanitize its input sources.

Reasoning: Reasoning is not available for this issue.

```javascript
var h = document.location.hash.substring(1); if (h & h != "") { var re = new RegExp (".+@.+"); if (h.match(re)) { document.getElementById("email").innerHTML += " (+h+)"); } }
```

Issue 2 of 3

<h1>Customize Site Language</h1>
### DOM Based Cross-Site Scripting

**Severity:** High

**URL:** http://demo.testfire.net/disclaimer.htm

**Entity:** disclaimer.htm:16 (Page)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** The web application uses client-side logic to create web pages

**Fix:** Analyze client side code and sanitize its input sources

**Reasoning:** Reasoning is not available for this issue.

```javascript
function go() {
  var iPos = document.URL.indexOf("url=")+4;
  var sDst = document.URL.substring(iPos,document.URL.length);
  if (window.opener) {
    window.opener.location.href = sDst;
    cl();
  } else {
    window.location.href = sDst;
  }
}
```

**Issue 3 of 3**

### DOM Based Cross-Site Scripting

**Severity:** High

**URL:** http://demo.testfire.net/disclaimer.htm

**Entity:** disclaimer.htm:19 (Page)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** The web application uses client-side logic to create web pages

**Fix:** Analyze client side code and sanitize its input sources

**Reasoning:** Reasoning is not available for this issue.

```javascript
function go() {
  var iPos = document.URL.indexOf("url=")+4;
  var sDst = document.URL.substring(iPos,document.URL.length);
  if (window.opener) {
    window.opener.location.href = sDst;
    cl();
  } else {
    window.location.href = sDst;
  }
}
```

### Poison Null Byte Windows Files Retrieval

**Issue 1 of 1**

10/2/2012
Poison Null Byte Windows Files Retrieval

Severity: High

URL: http://demo.testfire.net/default.aspx

Entity: content (Parameter)

Risk: It is possible to view the contents of any file (for example, databases, user information or configuration files) on the web server (under the permission restrictions of the web server user)

Causes: Sanitation of hazardous characters was not performed correctly on user input
User input is not checked for the '..' (dot dot) string

Fix: Ensure that accessed files reside in the virtual path and have certain extensions; remove special characters from user input

Reasoning: The test result seems to indicate a vulnerability because the response contained the contents of the "boot.ini" file, proving that the server allows remote users to download the contents of system files.

Raw Test Response:

...<li><a id="_ctl0__ctl0_Content_MenuHyperLink17" href="default.aspx?content=inside_press.htm">Press Room</a></li><li><a id="_ctl0__ctl0_Content_MenuHyperLink18" href="default.aspx?content=inside_careers.htm">Careers</a></li></ul></td><td valign="top" colspan="3" class="bb"><span id="_ctl0__ctl0_Content_Main_lblContent">[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS=Windows Server 2003, Enterprise* /fastdetect /bootlogo /nogui
</span></td></tr><table><td><td colspan="3" class="bb">...<li><a id="_ctl0__ctl0_Content_MenuHyperLink17" href="default.aspx?content=inside_press.htm">Press Room</a></li><li><a id="_ctl0__ctl0_Content_MenuHyperLink18" href="default.aspx?content=inside_careers.htm">Careers</a></li></ul></td><td valign="top" colspan="3" class="bb"><span id="_ctl0__ctl0_Content_Main_lblContent">[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS=Windows Server 2003, Enterprise* /fastdetect /bootlogo /nogui
</span></td></tr><table><td><td colspan="3" class="bb">...
Reasoning: This test consists of four requests: valid login, invalid login, login with predictable credentials, and another invalid login. If the response to the predictable credentials looks like the valid login (and different to the invalid logins), AppScan establishes that the application is vulnerable to this issue.

Valid Login

Test Login

---

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```html
...<div id="wrapper" style="width: 99%;">
  <div class="err" style="width: 99%;">
    <h1>An Error Has Occurred</h1>
    <h2>Summary:</h2>
    <h3>Error Message:</h3>
    <p>Syntax error in query expression \\
    "test@altoromutual.com";\'.
  </div>
</div>
```
## SQL Injection

**Severity:** High

**URL:** http://demo.testfire.net/bank/transaction.aspx

**Entity:** before (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitization of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

### Reasoning:
The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

### Raw Test Response:

```html
...<p><span id="ctl0_Content_lblDetails">System.Data.OleDbException: Syntax error in string in query expression '1=1 and t.trans_date &lt;= 1234 and t.trans_date &gt;= 1234'; and a.userid = 100116014 ORDER BY 1 DESC'.
</span></p>

...<p><span id="ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression '1=1 and t.trans_date &lt;= 1234 and t.trans_date &gt;= 1234'; and a.userid = 100116014 ORDER BY 1 DESC'.
</span></p>
```

10/2/2012
SQL Injection

Severity: High

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: after (Parameter)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

SQL Injection

Severity: **High**

URL: http://demo.testfire.net/bank/transfer.aspx

Entity: amUserId (Cookie)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```
...<div id="error" style="width: 99%;">
  <h1>An Error Has Occurred</h1>
  <h2>Summary:</h2>
  <p><b>Syntax error in string in query expression 'userid = 100116014'.</b></p>
  <h2>Error Message:</h2>
  <p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression 'userid = 100116014'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   ...</span></p>
...```

**Issue 5 of 12**

SQL Injection

Severity: **High**

URL: http://demo.testfire.net/bank/account.aspx

Entity: amUserId (Cookie)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.
### SQL Injection

#### Severity: High

**URL:** http://demo.testfire.net/bank/transaction.aspx

**Entity:** amUserId (Cookie)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```html
...<div id="err" style="width: 99%;">An Error Has Occurred</div><h2>Summary:</h2><p>Syntax error in string in query expression 'userid = 100116014'.</p><h2>Error Message:</h2><p>Syntax error in string in query expression 'userid = 100116014'.</p>...
```
Issue 7 of 12

**SQL Injection**

**Severity:** High

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** uid (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```html
...<div id="ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression '1=1 and t.trans_date <=' date '1234 and t.trans_date >=' date '1234 and a.userid = 100116014' ORDER BY 1 DESC'.
at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
  at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, Object&amp; executeResult)
  at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
  at System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
  at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
  ...</div>
```

10/2/2012
SQL Injection

**Severity:** High

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** passw (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```plaintext
...<div id="wrapper" style="width: 99%;">
  <div class="err" style="width: 99%;">
    <h1>An Error Has Occurred</h1>
    <h2>Summary:</h2>
    <p><b>Characters found after end of SQL statement.</b></p>
  </div>
</div>

</div>
```

10/2/2012
### SQL Injection

**Severity:** High

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** creditAccount (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```html
...<tr><td colspan="2" align="center"><input type="button" name="transfer" value="Transfer Money" onclick="doTransfer();" ID="transfer"></td></tr><tr><td colspan="2"><input name="soapRespon" id="soapResp" align="center" value="..." /></td></tr><tr><td colspan="2" align="center"><span id="_ctl0__ctl0_Content_Main_postResp" align="center"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error in string in query expression 'accountid=1001160141';'.
```n

---

### SQL Injection

**Severity:** High

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** debitAccount (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```html
...<tr><td colspan="2" align="center"><input type="button" name="transfer" value="Transfer Money" onclick="doTransfer();" ID="transfer"></td></tr><tr><td colspan="2"><input name="soapRespon" id="soapResp" align="center" value="..." /></td></tr><tr><td colspan="2" align="center"><span id="_ctl0__ctl0_Content_Main_postResp" align="center"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error in string in query expression 'accountid=1001160141';'.
```n

---
SQL Injection

Severity: **High**

**URL:** http://demo.testfire.net/bank/ws.asmx

**Entity:** [SOAP] debitAccount_1 (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:36:52 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1220


```
**SQL Injection**

**Severity:** High

**URL:** http://demo.testfire.net/bank/ws.asmx

**Entity:** [SOAP] creditAccount_2 (Parameter)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:37:00 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1235


 at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
## Unencrypted Login Request

**Severity:** High

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** login.aspx (Page)

**Risk:** It may be possible to steal user login information such as usernames and passwords that are sent unencrypted

**Causes:** Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted

**Fix:** Always use SSL and POST (body) parameters when sending sensitive information.

**Reasoning:** AppScan identified a login request that was not sent over SSL.

**Original Request**

uid=jsmith&passw=demo1234&btnSubmit=Login

---

## Issue 2 of 6

## Unencrypted Login Request

**Severity:** High

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** passw (Parameter)

**Risk:** It may be possible to steal user login information such as usernames and passwords that are sent unencrypted

**Causes:** Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted

**Fix:** Always use SSL and POST (body) parameters when sending sensitive information.

**Reasoning:** AppScan identified a password parameter that was not sent over SSL.

**Original Request**

uid=jsmith&passw=demo1234&btnSubmit=Login

---

## Issue 3 of 6

## Unencrypted Login Request

**Severity:** High

**URL:** http://demo.testfire.net/bank/apply.aspx

**Entity:** passwd (Parameter)

**Risk:** It may be possible to steal user login information such as usernames and passwords that are sent unencrypted

**Causes:** Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted

**Fix:** Always use SSL and POST (body) parameters when sending sensitive information.

**Reasoning:** AppScan identified a password parameter that was not sent over SSL.

**Original Request**

uid=jsmith&passw=demo1234&btnSubmit=Login

---

10/2/2012
Issue 4 of 6

Unencrypted Login Request

Severity: High
URL: http://demo.testfire.net/admin/admin.aspx
Entity: password1 (Parameter)
Risk: It may be possible to steal user login information such as usernames and passwords that are sent unencrypted
Causes: Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted
Fix: Always use SSL and POST (body) parameters when sending sensitive information.

Reasoning: AppScan identified a password parameter that was not sent over SSL.

Original Request

password1=Demo1234&Submit=Submit

Issue 5 of 6

Unencrypted Login Request

Severity: High
URL: http://demo.testfire.net/admin/admin.aspx
Entity: password2 (Parameter)
Risk: It may be possible to steal user login information such as usernames and passwords that are sent unencrypted
Causes: Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted
Fix: Always use SSL and POST (body) parameters when sending sensitive information.

Reasoning: AppScan identified a password parameter that was not sent over SSL.

Original Request

password1=Demo1234&password2=Demo1234&change=Change+Password

Issue 6 of 6
**Unencrypted Login Request**

**Severity:** High

**URL:** http://demo.testfire.net/admin/login.aspx

**Entity:** _ctl0:_ctl0:Content:Main:Password (Parameter)

**Risk:** It may be possible to steal user login information such as usernames and passwords that are sent unencrypted.

**Causes:** Sensitive input fields such as usernames, password and credit card numbers are passed unencrypted.

**Fix:** Always use SSL and POST (body) parameters when sending sensitive information.

**Reasoning:** AppScan identified a password parameter that was not sent over SSL.

**Original Request**

```
__VIEWSTATE=%2FwEPDwUKMTY5ODYzNWRk&__EVENTVALIDATION=%2FwEWBAKm%2FPqICgKaqvKtBqW7muPdCgq173pKUBA%3D%3Da_ct10%3A_ct10%3AContent%3AMain%3ACodeNumberTextBox=9876543210&_ctl0%3A_ctl0%3AContent%3AMain%3APassword=Demo1234&_ctl0%3A_ctl0%3AContent%3AMain%3ASubmitButton=Submit
```

**XPath Injection**

**Severity:** High

**URL:** http://demo.testfire.net/bank/queryxpath.aspx

**Entity:** _ctl0:_ctl0:Content:Main:TextBox1 (Parameter)

**Risk:** It is possible to access information stored in a sensitive data resource.

**Causes:** Sanitation of hazardous characters was not performed correctly on user input.

**Fix:** Review possible solutions for hazardous character injection.

**Reasoning:** The test result seems to indicate a vulnerability because the response contains an XPath exception. This suggests that the test managed to penetrate the application and reach the XPath query itself, by injecting hazardous characters.

**Raw Test Response:**

```
An Error Has Occurred

Summary:

String(/news/publication[contains(title,'"')?> and (isPublic/text()='True')/title/text()) has an invalid token.

Error Message:

String(/news/publication[contains(title,'"')?> and (isPublic/text()='True')/title/text()) has an invalid token.

System.Xml.XPath.XPathException: 'string(/news/publication[contains(title,'"')?> and (isPublic/text()='True')/title/text())' has an invalid token.
at MS.Internal.Xml.XPath.XPathParser.CheckToken(LexKind t)
at MS.Internal.Xml.XPath.XPathParser.ParseMethod(AstNode qyInput)
at MS.Internal.Xml.XPath.XPathParser.ParsePrimaryExpr(AstNode qyInput)
at MS.Internal.Xml.XPath.XPathParser.ParseFilterExpr(AstNode qyInput)
```

10/2/2012
Cross-Site Request Forgery

Severity: Medium

URL: http://demo.testfire.net/bank/login.aspx

Entity: login.aspx (Page)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Insufficient authentication method was used by the application

Fix: Decline malicious requests

Reasoning: The test result seems to indicate a vulnerability because the Test Response (on the right) is identical to the Original Response (on the left), indicating that the login attempt was successful, even though it included hazardous characters.
<table>
<thead>
<tr>
<th>Cross-Site Request Forgery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity:</strong> Medium</td>
</tr>
<tr>
<td><strong>URL:</strong> <a href="http://demo.testfire.net/bank/transfer.aspx">http://demo.testfire.net/bank/transfer.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong> transfer.aspx (Page)</td>
</tr>
<tr>
<td><strong>Risk:</strong> It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</td>
</tr>
<tr>
<td><strong>Causes:</strong> Insufficient authentication method was used by the application</td>
</tr>
<tr>
<td><strong>Fix:</strong> Decline malicious requests</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the same request was sent twice in different sessions, and the same response was received. This shows that none of the parameters are dynamic (session identifiers are sent only in cookies) and therefore that the application is vulnerable to CSRF.

**Test Request:**
```
POST /bank/transfer.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.referer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NETCLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 68
debitAccount=1001160141&creditAccount=1001160141&transferAmount=1234
```

**Test Response:**
```
POST /bank/transfer.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.referer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NETCLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 68
debitAccount=1001160141&creditAccount=1001160141&transferAmount=1234
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:25:26 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
...
Cross-Site Request Forgery

Severity: Medium

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: transaction.aspx (Page)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Insufficient authentication method was used by the application

Fix: Decline malicious requests

Reasoning: The test result seems to indicate a vulnerability because the same request was sent twice in different sessions, and the same response was received. This shows that none of the parameters are dynamic (session identifiers are sent only in cookies) and therefore that the application is vulnerable to CSRF.

Test Request:

POST /bank/transaction.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogusreferer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 176

__EVENTTARGET=4__EVENTARGUMENT=4__VIEWSTATE=42PaE9w3UKMTw5XNgp30tA4nkx4__EVENTVALIDATION=42PaEMy6KV3o82d8g3oeuaBAK3oaeosDgK3o2PRQ3oa52B3CgK3o2UuaQ43D5
3Dafter=1234before=1234

Test Response:

POST /bank/transaction.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogusreferer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 176

__EVENTTARGET=4__EVENTARGUMENT=4__VIEWSTATE=42PaE9w3UKMTw5XNgp30tA4nkx4__EVENTVALIDATION=42PaEMy6KV3o82d8g3oeuaBAK3oaeosDgK3o2PRQ3oa52B3CgK3o2UuaQ43D5
3Dafter=1234before=1234

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:03:11 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
...
Cross-Site Request Forgery

Severity: Medium

URL: http://demo.testfire.net/bank/customize.aspx

Entity: customize.aspx (Page)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Insufficient authentication method was used by the application

Fix: Decline malicious requests

Reasoning: The test result seems to indicate a vulnerability because the same request was sent twice in different sessions, and the same response was received. This shows that none of the parameters are dynamic (session identifiers are sent only in cookies) and therefore that the application is vulnerable to CSRF.

Test Request:

POST /bank/customize.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9; lang=Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.referer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 40

__VIEWSTATE=%2FwEPDwUJMjA2OTMxMDA4ZGQ%3D

Test Response:

POST /bank/customize.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9; lang=Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.referer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 40

__VIEWSTATE=%2FwEPDwUJMjA2OTMxMDA4ZGQ%3D

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:03:08 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=iso-8859-1
Content-Length: 5542

10/2/2012
## Cross-Site Request Forgery

**Severity:** Medium

**URL:** http://demo.testfire.net/bank/account.aspx

**Entity:** account.aspx (Page)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** Insufficient authentication method was used by the application

**Fix:** Decline malicious requests

### Reasoning:
The test result seems to indicate a vulnerability because the same request was sent twice in different sessions, and the same response was received. This shows that none of the parameters are dynamic (session identifiers are sent only in cookies) and therefore that the application is vulnerable to CSRF.

### Test Request:

```
POST /bank/account.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.referer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 23
listAccounts=1001160141
```

### Test Response:

```
POST /bank/account.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.referer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 23
listAccounts=1001160141
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:24:38 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
...
Cross-Site Request Forgery

Severity: Medium

URL: http://demo.testfire.net/admin/admin.aspx

Entity: admin.aspx (Page)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes: Insufficient authentication method was used by the application

Fix: Decline malicious requests

Reasoning: The test result seems to indicate a vulnerability because the same request was sent twice in different sessions, and the same response was received. This shows that none of the parameters are dynamic (session identifiers are sent only in cookies) and therefore that the application is vulnerable to CSRF.

Test Request:

POST /admin/admin.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=eu0qbsjnggirw49qopxa45; amSessionId=3545750533; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.reducer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 17
accttypes=Savings

Test Response:

POST /admin/admin.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=eu0qbsjnggirw49qopxa45; amSessionId=3545750533; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://bogus.reducer.ibm.com
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 17
accttypes=Savings
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:17:49 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1

Directory Listing

Issue 1 of 2

10/2/2012
## Directory Listing

**Severity:** Medium

**URL:** http://demo.testfire.net/bank/

**Entity:** bank/ (Page)

**Risk:** It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files

**Causes:** Directory browsing is enabled

**Fix:** Modify the server configuration to deny directory listing, and install the latest security patches available

**Reasoning:** The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

### Test Response

```
demo.testfire.net - /bank/

[To Parent Directory]
5/31/2007 12:10 PM  <dir>  20060803_bak
1/12/2011 11:14 PM  1831  account.aspx
1/12/2011 11:14 PM  4277  account.aspx.cs
1/12/2011 11:14 PM  771   apply.aspx
1/12/2011 11:14 PM  2528  apply.aspx.cs
1/12/2011 11:14 PM  2236  bank.master
1/12/2011 11:14 PM  1134  bank.master.cs
1/12/2011 11:14 PM  904   customize.aspx
1/12/2011 11:14 PM  1950  customize.aspx.cs
1/12/2011 11:14 PM  1806  login.aspx
1/12/2011 11:14 PM  5547  login.aspx.cs
1/12/2011 11:14 PM  78    logout.aspx
1/12/2011 11:14 PM  3361  logout.aspx.cs
1/12/2011 11:14 PM  938   main.aspx
1/12/2011 11:14 PM  3281  main.aspx.cs
5/31/2007 12:10 PM  <dir>  members
1/12/2011 11:14 PM  1414  mozpark.js
6/21/2011 11:29 PM  779   querypath.aspx
1/12/2011 11:14 PM  1833  querypath.aspx.cs
1/12/2011 11:14 PM  499   servererror.aspx
1/12/2011 11:14 PM  1700  transaction.aspx
1/12/2011 11:14 PM  3026  transaction.aspx.cs
1/12/2011 11:14 PM  3930  transfer.aspx
1/12/2011 11:14 PM  3508  transfer.aspx.cs
1/12/2011 11:14 PM  82   ws.asmx
```
**Directory Listing**

**Severity:** Medium

**URL:** http://demo.testfire.net/pr/

**Entity:** pr/ (Page)

**Risk:** It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files.

**Causes:** Directory browsing is enabled

**Fix:** Modify the server configuration to deny directory listing, and install the latest security patches available.

**Reasoning:** The response contains the content of a directory (directory listing). This indicates that the server allows the listing of directories, which is not usually recommended.

**Test Response**

```
demo.testfire.net - /pr/
```

<table>
<thead>
<tr>
<th>[To Parent Directory]</th>
<th>Date</th>
<th>Time</th>
<th>Size</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/12/2011 11:14 PM</td>
<td>63987</td>
<td>communityannualreport.pdf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6/21/2011 11:28 PM</td>
<td>779</td>
<td>dooc.xml</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/12/2011 11:14 PM</td>
<td>11281</td>
<td>Draft.rtf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/12/2011 11:14 PM</td>
<td>187754</td>
<td>Q3_earnings.rtf</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HTTP Response Splitting

Severity: Medium

URL: http://demo.testfire.net/bank/customize.aspx

Entity: lang (Parameter)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

It is possible to deface the site content through web-cache poisoning.

Causes: Sanitation of hazardous characters was not performed correctly on user input.

Fix: Review possible solutions for hazardous character injection.

Reasoning: The test result seems to indicate a vulnerability because the Global Validation feature found an embedded script in the response, which was probably injected by a previous test.

Raw Test Response:

```http
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:36:38 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
AppScanHeader: AppScanValue/1.2-3
SecondAppScanHeader: whatever; path=/
Cache-Control: private
Content-Type: text/html; charset=iso-8859-1
Content-Length: 5706
Set-Cookie: lang=FooBar

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

...
```

Inadequate Account Lockout

Issue 1 of 1

Severity: Medium

URL: http://demo.testfire.net/bank/login.aspx

Entity: passw (Parameter)

Risk: It might be possible to escalate user privileges and gain administrative permissions over the web application.

Causes: Insecure web application programming or configuration.

Fix: Enforce account lockout after several failed login attempts.

10/2/2012
Reasoning: Two legitimate login attempts were sent, with several false login attempts in between. The last response was identical to the first. This suggests that there is inadequate account lockout enforcement, allowing brute-force attacks on the login page. (This is true even if the first response was not a successful login page.)

Test Response (first)

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response
**Link Injection (facilitates Cross-Site Request Forgery)**

**Severity:** Medium

**URL:** http://demo.testfire.net/survey_complete.aspx

**Entity:** txtEmail (Parameter)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.
It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
It is possible to upload, modify or delete web pages, scripts and files on the web server.

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

**Test Response**
# Issue 3 of 6

## Link Injection (facilitates Cross-Site Request Forgery)

<table>
<thead>
<tr>
<th><strong>Severity:</strong></th>
<th><strong>Medium</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL:</strong></td>
<td><a href="http://demo.testfire.net/comment.aspx">http://demo.testfire.net/comment.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>name (Parameter)</td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
<td>It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user. It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to upload, modify or delete web pages, scripts and files on the web server.</td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
<td>Sanitation of hazardous characters was not performed correctly on user input</td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
<td>Review possible solutions for hazardous character injection</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

**Test Response**
Link Injection (facilitates Cross-Site Request Forgery)

Severity: Medium

URL: http://demo.testfire.net/bank/transfer.aspx
Entity: debitAccount (Parameter)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user. It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to upload, modify or delete web pages, scripts and files on the web server.

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response
Link Injection (facilitates Cross-Site Request Forgery)

Severity: Medium

URL: http://demo.testfire.net/bank/transfer.aspx

Entity: creditAccount (Parameter)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user. It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to upload, modify or delete web pages, scripts and files on the web server.

Causes: Sanitation of hazardous characters was not performed correctly on user input.

Fix: Review possible solutions for hazardous character injection.

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response

10/2/2012
Link Injection (facilitates Cross-Site Request Forgery)

Severity: Medium

URL: http://demo.testfire.net/bank/customize.aspx

Entity: lang (Parameter)

Risk: It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user. It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc. It is possible to upload, modify or delete web pages, scripts and files on the web server.

Causes: Sanitation of hazardous characters was not performed correctly on user input.

Fix: Review possible solutions for hazardous character injection.

Reasoning: The test result seems to indicate a vulnerability because the test response contained a link to the file "WF_XSRF.html".

Test Response
Open Redirect

Severity: Medium

URL: http://demo.testfire.net/disclaimer.htm

Entity: disclaimer.htm:32 (Page)

Risk: It is possible for an attacker to use the web server to attack other sites, which increases his or her anonymity.

Causes: The web application performs a redirection to an external site.

Fix: Analyze and harden client side (JavaScript) code.

Reasoning: Reasoning is not available for this issue.

```javascript
var iPos = document.URL.indexOf("url=")+4; var sDst = document.URL.substring(iPos,document.URL.length); // if redirection is in the application's domain, don't ask for authorization if (sDst.indexOf
```
Open Redirect

Severity: Medium

URL: http://demo.testfire.net/disclaimer.htm

Entity: disclaimer.htm:35 (Page)

Risk: It is possible for an attacker to use the web server to attack other sites, which increases his or her anonymity

Causes: The web application performs a redirection to an external site

Fix: Analyze and harden client side (JavaScript) code.

Reasoning: Reasoning is not available for this issue.

```
("http") == 0 && sDst.indexOf(document.location.hostname) != -1 ) { if (window.opener) { window.opener.location.href = "http" + sDst.substring(4); cl(); } else { window.location.href = "http" + sDst.substring(4); }
```

Phishing Through Frames

Severity: Medium

URL: http://demo.testfire.net/search.aspx

Entity: txtSearch (Parameter)

Risk: It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

Test Response
<table>
<thead>
<tr>
<th><strong>Phishing Through Frames</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity:</strong></td>
<td><strong>Medium</strong></td>
</tr>
<tr>
<td><strong>URL:</strong></td>
<td><a href="http://demo.testfire.net/survey_complete.aspx">http://demo.testfire.net/survey_complete.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>txtEmail (Parameter)</td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
<td>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
<td>Sanitation of hazardous characters was not performed correctly on user input</td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
<td>Review possible solutions for hazardous character injection</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

**Test Response**
Phishing Through Frames

**Severity:** Medium

**URL:** http://demo.testfire.net/comment.aspx

**Entity:** name (Parameter)

**Risk:** It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

**Test Response**
<table>
<thead>
<tr>
<th>Phishing Through Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity:</strong> Medium</td>
</tr>
<tr>
<td><strong>URL:</strong> <a href="http://demo.testfire.net/bank/transfer.aspx">http://demo.testfire.net/bank/transfer.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong> debitAccount (Parameter)</td>
</tr>
<tr>
<td><strong>Risk:</strong> It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</td>
</tr>
<tr>
<td><strong>Causes:</strong> Sanitation of hazardous characters was not performed correctly on user input</td>
</tr>
<tr>
<td><strong>Fix:</strong> Review possible solutions for hazardous character injection</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

**Test Response**
---

### Issue 5 of 6

<table>
<thead>
<tr>
<th>Phishing Through Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity:</strong></td>
</tr>
<tr>
<td><strong>URL:</strong></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

**Test Response**

---

10/2/2012
### Phishing Through Frames

<table>
<thead>
<tr>
<th><strong>Severity:</strong></th>
<th>Medium</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL:</strong></td>
<td><a href="http://demo.testfire.net/bank/customize.aspx">http://demo.testfire.net/bank/customize.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>lang (Parameter)</td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
<td>It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.</td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
<td>Sanitation of hazardous characters was not performed correctly on user input</td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
<td>Review possible solutions for hazardous character injection</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the test response contained a frame/iframe to URL "http://demo.testfire.net/phishing.html".

**Test Response**

Phishing Sample
### Session Identifier Not Updated

#### Issue 1 of 1

##### Session Identifier Not Updated

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity</strong></td>
<td>Medium</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://demo.testfire.net/bank/login.aspx">http://demo.testfire.net/bank/login.aspx</a></td>
</tr>
<tr>
<td><strong>Entity</strong></td>
<td>login.aspx (Page)</td>
</tr>
<tr>
<td><strong>Risk</strong></td>
<td>It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</td>
</tr>
<tr>
<td><strong>Causes</strong></td>
<td>Insecure web application programming or configuration</td>
</tr>
<tr>
<td><strong>Fix</strong></td>
<td>Do not accept externally created session identifiers</td>
</tr>
</tbody>
</table>

**Reasoning:** The test result seems to indicate a vulnerability because the session identifiers in the Original Request (on the left) and in the Response (on the right) are identical. They should have been updated in the response.
**Issue 1 of 4**

**Autocomplete HTML Attribute Not Disabled for Password Field**

**Severity:** Low

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** login.aspx (Page)

**Risk:** It may be possible to bypass the web application's authentication mechanism

**Causes:** Insecure web application programming or configuration

**Fix:** Correctly set the "autocomplete" attribute to "off"

**Reasoning:** AppScan has found that a password field does not enforce the disabling of the autocomplete feature.

**Raw Test Response:**

```html
...<table>
  <tr>
    <td>Password:<br></td>
    <td><input type="password" id="passw" name="passw" style="width: 150px;"> <input type="submit" name="btnSubmit" value="Login"> </td>
  </tr>
</table>
```

**Issue 2 of 4**

**Autocomplete HTML Attribute Not Disabled for Password Field**

**Severity:** Low

**URL:** http://demo.testfire.net/bank/apply.aspx

**Entity:** apply.aspx (Page)

**Risk:** It may be possible to bypass the web application's authentication mechanism

**Causes:** Insecure web application programming or configuration

**Fix:** Correctly set the "autocomplete" attribute to "off"

**Reasoning:** AppScan has found that a password field does not enforce the disabling of the autocomplete feature.

**Raw Test Response:**

```html
...<form>
  <input type="submit" name="btnSubmit" value="Login"><br>
</form>
```

*10/2/2012*
Autocomplete HTML Attribute Not Disabled for Password Field

Severity: Low

URL: http://demo.testfire.net/admin/login.aspx

Entity: login.aspx (Page)

Risk: It may be possible to bypass the web application's authentication mechanism

Causes: Insecure web application programming or configuration

Fix: Correctly set the "autocomplete" attribute to "off"

Reasoning: AppScan has found that a password field does not enforce the disabling of the autocomplete feature.

Raw Test Response:

```html
<!--
    userid = userCookie.Values["UserID"].ToString();
cLimit = Request.Cookies["Limit"].Value;
cInterest = Request.Cookies["Interest"].Value;
cType = Request.Cookies["CardType"].Value;
--><span id="_ctl0__ctl0_Content_Main_lblMessage"><p><b>No application is needed.</b>To approve your new $10000 Altoro Mutual Gold Visa<br />with a 7.9% APR simply enter your password below.&lt;/p&gt;&lt;p&gt;&lt;form method="post" name="Credit" action="apply.aspx"&gt;&lt;table border="0"&gt;&lt;tr&gt;&lt;td&gt;&lt;input type="password" name="passwd"&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/form&gt;&lt;/span&gt;

<!--
    Password is not revalidated but stored in mainframe for non-repudiation purposes.
--></div>

...
Autocomplete HTML Attribute Not Disabled for Password Field

Severity: Low
URL: http://demo.testfire.net/admin/admin.aspx
Entity: admin.aspx (Page)
Risk: It may be possible to bypass the web application's authentication mechanism
Causes: Insecure web application programming or configuration
Fix: Correctly set the "autocomplete" attribute to "off"

Reasoning: AppScan has found that a password field does not enforce the disabling of the autocomplete feature.

Raw Test Response:

```html
...<td colspan="4">Add a new user.</td></tr>
...<td colspan="4">It is highly recommended that you leave the username as first
initial last name. The user id will be created automatically.</td></tr>
...```
Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/subscribe.aspx

**Entity:** subscribe.aspx (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
...<div id="wrapper" style="width: 99%;">
  <div class="err" style="width: 99%;">
    <h1>An Error Has Occurred</h1>
    <h2>Summary:</h2>
    <p><b><span id="ctl0_Content_lblSummary">Syntax error (missing operator) in query expression "&quot;&amp;quot;&lt;script&gt;alert(1524)&lt;/script&gt;"'.</span></b></p>
    <h2>Error Message:</h2>
    <p><span id="ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression "&quot;&amp;quot;&lt;script&gt;alert(1524)&lt;/script&gt;"'.
      at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
      at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp;amp; executeResult)
      at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp;amp; executeResult)
      at System.Data.OleDb.OleDbCommand.ExecuteNonQuery()
      at Altoro.Subscribe.Page_Load(Object sender, EventArgs e) in d:\downloads\AltoroMutual_v6\website\subscribe.aspx.cs:line 48
    </p>
  </div>
</div>
...```
Database Error Pattern Found

Severity: **Low**

URL: http://demo.testfire.net/subscribe.aspx

Entity: txtEmail (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

```html
...<div id="error" style="width: 99%;">
  <div id="error-message">
    <h1>An Error Has Occurred</h1>
    <h2>Summary:</h2>
    <p>Syntax error in string in query expression ''test@altoromutual.comWFXSSProbe''</p>
    <h2>Error Message:</h2>
    <p>Syntax error in string in query expression ''test@altoromutual.comWFXSSProbe''</p>
  </div>
</div>...```

Issue 3 of 16

Database Error Pattern Found

Severity: **Low**

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: before (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:
Database Error Pattern Found

Severity: Low

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: after (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

...
Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** login.aspx (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
...<div id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression '1=1 and t.trans_date >= 1234WFXSSProbe and t.trans_date <= 1234 and a.userid = 100116014'.
at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String srcTable, IDbCommand command, CommandBehavior behavior)
at System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
...```
Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** amUserId (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```html
...<div id="wrapper" style="width: 99%;"/>

<div class="err" style="width: 99%;">
<h1>An Error Has Occurred</h1>
<h2>Summary:</h2>
<p><b><span id="_ctl0_Content_lblSummary">Syntax error (missing operator) in query expression 'userid = 100116014WFXSProbe'.</span></b></p>
<h2>Error Message:</h2>
<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression 'userid = 100116014WFXSProbe'.

  at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
  at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
  at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, Object&amp; executeResult)
  at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
  at System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
  at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)

...</span></p>
```

Issue 7 of 16
Database Error Pattern Found

Severity: **Low**

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** amUserId (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
...<div id="err" style="width: 99%;">

<h1>An Error Has Occurred</h1>
<h2>Summary:</h2>
<p><b>Syntax error (missing operator) in query expression 'userid = 100116014WFXSSProbe'.</b></p>
<h2>Error Message:</h2>
<p>System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression 'userid = 100116014WFXSSProbe'.

at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)

...```

Issue 8 of 16

Database Error Pattern Found

Severity: **Low**

**URL:** http://demo.testfire.net/bank/account.aspx

**Entity:** amUserId (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.
Database Error Pattern Found

Severity: **Low**

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: `amUserId` (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

Reasoning: The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

Raw Test Response:

...
Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/bank/transaction.aspx

**Entity:** transaction.aspx (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```xml
...<div id="ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression '1=1 and t.trans_date sgt= 1234 and t.trans_date ltt= 1234 and a.userid = 100116104 ORDER BY 1 DESC'.
  at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
  at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp;amp; executeResult)
  at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp;amp; executeResult)
  at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
  at System.Data.Common.DbDataAdapter.FillInternal(DataSet dataset, DataTable[] datatables, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
  at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, Int32 startRecord, Int32 maxRecords, String srcTable, IDbCommand command, CommandBehavior behavior)
..."```
Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** passw (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```html
...<div id="err" style="width: 99%;">
  <h1>An Error Has Occurred</h1>
  <h2>Summary:</h2>
  <p><b>Syntax error in string in query expression 'username = 'jsmith' AND password = 'demo1234WFXSSProbe''</b>.</p>
</div>...
```

**Issue 12 of 16**
Database Error Pattern Found

Severity: **Low**

URL:  http://demo.testfire.net/bank/login.aspx

Entity:  uid (Global)

Risk:  It is possible to view, modify or delete database entries and tables

Causes:  Sanitation of hazardous characters was not performed correctly on user input

Fix:  Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
...  
</td>  
<td colspan="2" align="center">  
<input type="button" name="transfer" value="Transfer Money" onclick="doTransfer();" ID="transfer" />
</td>
</tr>
</table>
```

## Issue 14 of 16

### Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** debitAccount (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
...  
</td>  
<td colspan="2" align="center">  
<iframe src="javascript:alert(2435)"></iframe>
</td>
</tr>
</table>
```

```
...  
</td>  
<td colspan="2" align="center">  
<iframe src="javascript:alert(2435)"></iframe>
</td>
</tr>
</table>
```

```
...  
</td>  
<td colspan="2" align="center">  
<iframe src="javascript:alert(2435)"></iframe>
</td>
</tr>
</table>
```
Database Error Pattern Found

Severity: **Low**

URL: http://demo.testfire.net/bank/ws.asmx

Entity: [SOAP] creditAccount_2 (Global)

Risk: It is possible to view, modify or delete database entries and tables

Causes: Sanitation of hazardous characters was not performed correctly on user input

Fix: Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:36:54 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1243

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
Syntax error (missing operator) in query expression 'accountid=1001160141WFXSSProbe'.

at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
```
### Database Error Pattern Found

**Severity:** Low

**URL:** http://demo.testfire.net/bank/ws.asmx

**Entity:** [SOAP] debitAccount_1 (Global)

**Risk:** It is possible to view, modify or delete database entries and tables

**Causes:** Sanitation of hazardous characters was not performed correctly on user input

**Fix:** Review possible solutions for hazardous character injection

**Reasoning:** The test result seems to indicate a vulnerability because the response contains SQL Server errors. This suggests that the test managed to penetrate the application and reach the SQL query itself, by injecting hazardous characters.

**Raw Test Response:**

```
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:36:48 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Content-Type: text/xml; charset=utf-8
Content-Length: 1243

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
  </Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
```

### Direct Access to Administration Pages

**Issue 1 of 2**

**Severity:** Low

**URL:** http://demo.testfire.net/survey_questions.aspx

**Entity:** admin.aspx (Page)

**Risk:** It might be possible to escalate user privileges and gain administrative permissions over the web application

**Causes:** The web server or application server are configured in an insecure way

**Fix:** Apply proper authorization to administration scripts

**Reasoning:** AppScan requested a file which is probably not a legitimate part of the application. The response status was 200 OK.
This indicates that the test succeeded in retrieving the content of the requested file.

Test Request:

GET /admin/admin.aspx HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/xaml+xml,
application/vnd.ms-xpsdocument, application/x-ms-xbap,
application/x-ms-application, application/vnd.ms-excel,
application/xYLsdesword, */*
Referer: http://demo.testfire.net/survey_questions.aspx?step=a
Accept-Language: en-us
UA-CPU: x86
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1;
Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET
CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Connection: Keep-Alive
Host: demo.testfire.net
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55;
amSessionId=334738728;
amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=;
amUserId=100116014;
amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9

Test Response

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:17:49 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1
Content-Length: 786

...
Email Address Pattern Found in Parameter Value

**Severity:** Low

**URL:** http://demo.testfire.net/survey_complete.aspx

**Entity:** txtEmail (Parameter)

**Risk:** It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

**Causes:** Insecure web application programming or configuration

**Fix:** Remove e-mail addresses from the website

**Reasoning:** A parameter value contains an e-mail address that may be private.

**Raw Test Response:**

```
GET /survey_complete.aspx?txtEmail=jsmith@demo.testfire.net HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/png, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/xaml+xml, application/vnd.ms-xpsdocument, application/x-ms-application, application/msexcel, Application/vnd.ms-excel, Application/pdf, */*
Referer: http://demo.testfire.net/survey_complete.aspx
```

---

Raw Test Response:

```
GET /admin/admin.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9 Accept-Language: en-US Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Referer: http://demo.testfire.net/default.aspx?content=personal_other.htm Host: demo.testfire.net User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.5.30729)
...```

---

```
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:17:49 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1
Content-Length: 7861
...```

---

```
GET /admin/admin.aspx HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9 Accept-Language: en-US Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Referer: http://demo.testfire.net/default.aspx?content=personal_other.htm Host: demo.testfire.net User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.5.30729)
...```
Email Address Pattern Found in Parameter Value

Severity: **Low**

URL:  http://demo.testfire.net/subscribe.aspx

Entity:  txtEmail (Parameter)

Risk:  It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes:  Insecure web application programming or configuration

Fix:  Remove e-mail addresses from the website

**Reasoning:** A parameter value contains an e-mail address that may be private.

**Raw Test Response:**

```
POST /subscribe.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=k14vue55ie00airp0c2bhvqo; amSessionId=324838572; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=;
amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/subscribe.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.4506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 52
txEmail=test%40altoromutual.com&btnSubmit=Subscribe
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:03:13 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
```
## Hidden Directory Detected

**Severity:** Low  
**URL:** http://demo.testfire.net/images/  
**Entity:** images/ (Page)  
**Risk:** It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site  
**Causes:** The web server or application server are configured in an insecure way  
**Fix:** Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: The test tried to detect hidden directories on the server. The 403 Forbidden response reveals the existence of the directory, even though access is not allowed.

**Test Response**

**Directory Listing Denied**

This Virtual Directory does not allow contents to be listed.
Hidden Directory Detected

Severity: Low
URL: http://demo.testfire.net/admin/
Entity: admin/ (Page)
Risk: It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site
Causes: The web server or application server are configured in an insecure way
Fix: Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely

Reasoning: The test tried to detect hidden directories on the server. The 403 Forbidden response reveals the existence of the directory, even though access is not allowed.

Test Response

Directory Listing Denied

This Virtual Directory does not allow contents to be listed.
Hidden Directory Detected

Severity: Low

URL: http://demo.testfire.net/aspnet_client/

Entity: aspnet_client/ (Page)

Risk: It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site.

Causes: The web server or application server are configured in an insecure way.

Fix: Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely.

Reasoning: The test tried to detect hidden directories on the server. The 403 Forbidden response reveals the existence of the directory, even though access is not allowed.

Test Response

Directory Listing Denied

This Virtual Directory does not allow contents to be listed.
Microsoft ASP.NET Debugging Enabled

Severity: **Low**

URL:  [http://demo.testfire.net/survey_questions.aspx](http://demo.testfire.net/survey_questions.aspx)

Entity:  AppScan.aspx (Page)

Risk:  It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes:  Insecure web application programming or configuration

Fix:  Disable Debugging on Microsoft ASP.NET

**Reasoning:** AppScan sent a request in Debug mode. The response indicates that debugging support in ASP.NET can be enabled. This may allow access to information about the server and application.

**Raw Test Response:**

```
...
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:18:12 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 0
...
```

**Issue 2 of 3**

Microsoft ASP.NET Debugging Enabled

Severity: **Low**

URL:  [http://demo.testfire.net/bank/main.aspx](http://demo.testfire.net/bank/main.aspx)

Entity:  AppScan.aspx (Page)

Risk:  It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes:  Insecure web application programming or configuration

Fix:  Disable Debugging on Microsoft ASP.NET

**Reasoning:** AppScan sent a request in Debug mode. The response indicates that debugging support in ASP.NET can be enabled. This may allow access to information about the server and application.

**Raw Test Response:**

```
...
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:18:12 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 0
...
```
### Microsoft ASP.NET Debugging Enabled

**Severity:** Low

**URL:** http://demo.testfire.net/admin/clients.xls

**Entity:** AppScan.aspx (Page)

**Risk:** It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

**Causes:** Insecure web application programming or configuration

**Fix:** Disable Debugging on Microsoft ASP.NET

**Reasoning:** AppScan sent a request in Debug mode. The response indicates that debugging support in ASP.NET can be enabled. This may allow access to information about the server and application.

#### Raw Test Response:

```
HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:18:12 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 
```

---

### Missing HttpOnly Attribute in Session Cookie

**Severity:** Critical

**URL:**

**Entity:**

**Risk:**

**Causes:**

**Fix:**

---

**Issue 1 of 4**

---

10/2/2012
## Missing HttpOnly Attribute in Session Cookie

**Severity:** Low  
**URL:** http://demo.testfire.net/  
**Entity:** amSessionId (Cookie)  
**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user  
**Causes:** The web application sets session cookies without the HttpOnly attribute  
**Fix:** Add the ‘HttpOnly’ attribute to all session cookies

**Reasoning:** AppScan found that a session cookie is used without the “HttpOnly” attribute.

### Original Response

GET / HTTP/1.1  
Accept-Language: en-US  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Host: demo.testfire.net  
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)

HTTP/1.1 200 OK  
Date: Sun, 22 Jul 2012 08:03:24 GMT  
Server: Microsoft-IIS/6.0  
X-Powered-By: ASP.NET  
X-AspNet-Version: 2.0.50727  
Cache-Control: no-cache  
Pragma: no-cache  
Expires: -1  
Content-Type: text/html; charset=iso-8859-1  
Content-Length: 9645  
Set-Cookie: ASP.NET_SessionId=n5pgfuf5tly12ds553uu5bn55; path=/; HttpOnly  
Set-Cookie: amSessionId=332438668; path=/; ...
## Issue 3 of 4

### Missing HttpOnly Attribute in Session Cookie

<table>
<thead>
<tr>
<th>Severity:</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL:</td>
<td><a href="http://demo.testfire.net/bank/login.aspx">http://demo.testfire.net/bank/login.aspx</a></td>
</tr>
<tr>
<td>Entity:</td>
<td>amUserId (Cookie)</td>
</tr>
<tr>
<td>Risk:</td>
<td>It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user</td>
</tr>
<tr>
<td>Causes:</td>
<td>The web application sets session cookies without the HttpOnly attribute</td>
</tr>
<tr>
<td>Fix:</td>
<td>Add the 'HttpOnly' attribute to all session cookies</td>
</tr>
</tbody>
</table>

**Reasoning:** AppScan found that a session cookie is used without the "HttpOnly" attribute.

**Original Response**

```
POST /bank/login.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=k14vue55ie00airp0c2bhvqo; amSessionId=324838572; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/bank/login.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 41
uid=jsmith&passw=demo1234&btnSubmit=Login

HTTP/1.1 302 Found
Date: Sun, 22 Jul 2012 08:03:20 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Location: /bank/main.aspx
Cache-Control: no-cache
Pragma: no-cache

...```

## Issue 4 of 4
**Missing HttpOnly Attribute in Session Cookie**

**Severity:** Low

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** amUserInfo (Cookie)

**Risk:** It is possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.

**Causes:** The web application sets session cookies without the HttpOnly attribute

**Fix:** Add the 'HttpOnly' attribute to all session cookies

**Reasoning:** AppScan found that a session cookie is used without the "HttpOnly" attribute.

**Original Response**

```
POST /bank/login.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=k14vue5jie00alrp02xb0vq; amSessionId=324838572; amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer-CardType=Gold&Limit=1000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/bank/login.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 41

uid=jsmith&passw=demo1234&btnSubmit=Login
```

**Permanent Cookie Contains Sensitive Session Information**

**Severity:** Low

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** amUserInfo (Cookie)

**Risk:** It may be possible to steal session information (cookies) that was kept on disk as permanent cookies

**Causes:** The web application stores sensitive session information in a permanent cookie (on disk)

**Fix:** Avoid storing sensitive session information in permanent cookies

**Reasoning:** AppScan found that a session id cookie is stored on the client machine.

**Original Response**
Unencrypted __VIEWSTATE Parameter

Severity: Low

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: __VIEWSTATE (Parameter)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Modify your Web.Config file to encrypt the VIEWSTATE parameter

Reasoning: AppScan decoded the __VIEWSTATE parameter value and found it to be unencrypted.

Original Request

...
Issue 2 of 4

Unencrypted __VIEWSTATE Parameter

Severity: **Low**

**URL:** http://demo.testfire.net/bank/queryxpath.aspx

**Entity:** __VIEWSTATE (Parameter)

**Risk:** It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

**Causes:** Insecure web application programming or configuration

**Fix:** Modify your Web.Config file to encrypt the VIEWSTATE parameter

**Reasoning:** AppScan decoded the __VIEWSTATE parameter value and found it to be unencrypted.

**Original Request**

...<form name="aspnetForm" method="get" action="queryxpath.aspx?ctl0%3a_ctl0%3aContent%3aMain%3aTextBox1=Enter+title+(e.g.+IBM)&amp;ctl0%3a_ctl0%3aContent%3aMain%3aButton1=Query" id="aspnetForm" action="queryxpath.aspx" runat="server" method="get" name="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTEzMDczNTAxOWQk" />
<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWAwLNx+2YBwKw59eKgKjJjPAw==" />
<span id="ctl0__ctl0_Content_Main_Label1">Search our news articles database</span>
<br />
<input name="ctl0__ctl0_Content_Main_TextBox1" type="text" value="Enter title (e.g. IBM)" id="ctl0__ctl0_Content_Main_TextBox1" style="width:300px;" />
<input type="submit" name="ctl0__ctl0_Content_Main_Button1" value="Query" id="ctl0__ctl0_Content_Main_Button1" style="width:75px;" />
<br />
<span id="ctl0__ctl0_Content_Main_Label2">News title not found, try again</span>
</form>...
### Unencrypted __VIEWSTATE Parameter

<table>
<thead>
<tr>
<th><strong>Severity:</strong></th>
<th><strong>Low</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL:</strong></td>
<td><a href="http://demo.testfire.net/bank/customize.aspx">http://demo.testfire.net/bank/customize.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>__VIEWSTATE (Parameter)</td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
<td>It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations</td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
<td>Insecure web application programming or configuration</td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
<td>Modify your Web.Config file to encrypt the VIEWSTATE parameter</td>
</tr>
</tbody>
</table>

**Reasoning:** AppScan decoded the __VIEWSTATE parameter value and found it to be unencrypted.

**Original Request**

```html
...<td valign="top" colspan="3" class="bb">
</td> </tr>
<tr><td valign="top" colspan="3" class="bb">
<div class="f1" style="width: 99%;">
<h1>Customize Site Language</h1>
<form name="aspnetForm" method="post" action="customize.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUJMjA2OTMxMDA4ZGQ=" />
<p>
<span id="_ctl0__ctl0_Content_Main_Label1">Current Language: </span>
<span id="_ctl0__ctl0_Content_Main_langLabel"> </span>
</p>
<p>
<span id="_ctl0__ctl0_Content_Main_Label2">You can change the language setting by choosing: </span>
</p><p>
</p></form></div></td>  
<tr><td valign="top" colspan="3" class="bb">
...  
```

### Issue 4 of 4

<table>
<thead>
<tr>
<th><strong>Severity:</strong></th>
<th><strong>Low</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URL:</strong></td>
<td><a href="http://demo.testfire.net/admin/login.aspx">http://demo.testfire.net/admin/login.aspx</a></td>
</tr>
<tr>
<td><strong>Entity:</strong></td>
<td>__VIEWSTATE (Parameter)</td>
</tr>
<tr>
<td><strong>Risk:</strong></td>
<td>It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations</td>
</tr>
<tr>
<td><strong>Causes:</strong></td>
<td>Insecure web application programming or configuration</td>
</tr>
<tr>
<td><strong>Fix:</strong></td>
<td>Modify your Web.Config file to encrypt the VIEWSTATE parameter</td>
</tr>
</tbody>
</table>

**Reasoning:** AppScan decoded the __VIEWSTATE parameter value and found it to be unencrypted.

**Original Request**

```html
... </td> </tr>
<tr><td valign="top" colspan="3" class="bb">
<div class="f1" style="width: 99%;">
<h1>Customize Site Language</h1>
<form name="aspnetForm" method="post" action="customize.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUJMjA2OTMxMDA4ZGQ=" />
<p>
<span id="_ctl0__ctl0_Content_Main_Label1">Current Language: </span>
<span id="_ctl0__ctl0_Content_Main_langLabel"></span>
</p>
<p>
<span id="_ctl0__ctl0_Content_Main_Label2">You can change the language setting by choosing: </span>
</p>
</form></div></td>  
<tr><td valign="top" colspan="3" class="bb">
...  
```

10/2/2012
Unsigned __VIEWSTATE Parameter

Issue 1 of 4

Severity: Low

URL: http://demo.testfire.net/bank/transaction.aspx

Entity: __VIEWSTATE (Parameter)

Risk: It might be possible to undermine application logic

Causes: Insecure web application programming or configuration

Fix: Modify the property of each ASP.NET page to sign the VIEWSTATE parameter

Reasoning: AppScan determined that the __VIEWSTATE parameter value is unsigned.

Original Request

...
### Issue 2 of 4

**Unsigned __VIEWSTATE Parameter**

<table>
<thead>
<tr>
<th>Severity:</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL:</td>
<td><a href="http://demo.testfire.net/bank/queryxpath.aspx">http://demo.testfire.net/bank/queryxpath.aspx</a></td>
</tr>
<tr>
<td>Entity:</td>
<td>__VIEWSTATE (Parameter)</td>
</tr>
<tr>
<td>Risk:</td>
<td>It might be possible to undermine application logic</td>
</tr>
<tr>
<td>Causes:</td>
<td>Insecure web application programming or configuration</td>
</tr>
<tr>
<td>Fix:</td>
<td>Modify the property of each ASP.NET page to sign the VIEWSTATE parameter</td>
</tr>
</tbody>
</table>

**Reasoning:** AppScan determined that the __VIEWSTATE parameter value is unsigned.

#### Original Request

```html
...<td valign="top" colspan="3" class="bb">

<div class="f1" style="width: 99%;">
<h1>Search News Articles</h1>
<form name="aspnetForm" method="get" action="queryxpath.aspx?_ctl0%3a_ctl0%3aContent%3aMain%3aTextBox1=Enter+title+(e.g.+IBM)\&_ctl0%3a_ctl0%3aContent%3aMain%3aButton1=Query" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTEzMDczNTAxOWRk" />
<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWAwLNx+2YBwKw59aKCPKc3oPAw==" />
<span id="ctl0__ctl0_Content_Main_Label1">Search our news articles database</span>
<br /><br />
<input name="ctl0__ctl0:Content:Main:TextBox1" type="text" value="Enter title (e.g. IBM)" id="ctl0__ctl0__ctl0_Content_Main_TextBox1" style="width:300px;" />
<input type="submit" name="ctl0__ctl0:Content:Main:Button1" value="Query" id="ctl0__ctl0__ctl0_Content_Main_Button1" style="width:75px;" />
<br />
<br />
<span id="ctl0__ctl0__ctl0_Content_Main_Label2">News title not found, try again</span>
</form>
...```

### Issue 3 of 4

**Unsigned __VIEWSTATE Parameter**

<table>
<thead>
<tr>
<th>Severity:</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL:</td>
<td><a href="http://demo.testfire.net/bank/customize.aspx">http://demo.testfire.net/bank/customize.aspx</a></td>
</tr>
<tr>
<td>Entity:</td>
<td>__VIEWSTATE (Parameter)</td>
</tr>
<tr>
<td>Risk:</td>
<td>It might be possible to undermine application logic</td>
</tr>
<tr>
<td>Causes:</td>
<td>Insecure web application programming or configuration</td>
</tr>
<tr>
<td>Fix:</td>
<td>Modify the property of each ASP.NET page to sign the VIEWSTATE parameter</td>
</tr>
</tbody>
</table>
Reasoning: AppScan determined that the __VIEWSTATE parameter value is unsigned.

Original Request

...
Issue 1 of 15

Application Error

Severity: Informational

URL: http://demo.testfire.net/comment.aspx

Entity: cfile (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

... Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/feedback.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 79

name=1234&email_addr=753+Main+Street&subject=1234&comments=1234&submit=Submit+

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:04:05 GMT
Server: Microsoft- IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1

...
# Application Error

**Severity:** **Informational**  
**URL:** [http://demo.testfire.net/subscribe.aspx](http://demo.testfire.net/subscribe.aspx)  
**Entity:** txtEmail (Parameter)  
**Risk:** It is possible to gather sensitive debugging information  
**Causes:** Proper bounds checking were not performed on incoming parameter values. No validation was done in order to make sure that user input matches the data type expected  
**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```plaintext
...
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/subscribe.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 32

txtEmail=%27&btnSubmit=Subscribe

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:37:41 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1

...
```

---

# Application Error

**Severity:** **Informational**  
**URL:** [http://demo.testfire.net/bank/queryxpath.aspx](http://demo.testfire.net/bank/queryxpath.aspx)  
**Entity:** _ctl0:_ctl0:Content:Main:TextBox1 (Parameter)  
**Risk:** It is possible to gather sensitive debugging information  
**Causes:** Proper bounds checking were not performed on incoming parameter values. No validation was done in order to make sure that user input matches the data type expected  
**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```plaintext
...
```
Issue 4 of 15

Application Error

Severity: Informational

URL: http://demo.testfire.net/bank/ws.asmx

Entity: WSDL (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

GET /bank/queryxpath.aspx?__VIEWSTATE=%2FwEPDwUKMTEzMDczNTAxOWRk&__EVENTVALIDATION=%2FwEWAwLNx%2B2YbWx54eKCgKcjoPA6w%3D%3D&ctl03A_ctl0%3AContent%3AMain%3ATextBox1=%27&_ctl0%3A_ctl0%3AContent%3AMain%3AButton1=Query HTTP/1.1
Cookie: ASP.NET_SessionId=wh542tn0dduonh55mkqtnr55; amSessionId=334738728; amUserlnfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=GoldLimit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/bank/queryxpath.aspx
Host: demo.testfire.net

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:29:56 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1

GET /bank/ws.asmx?WSDL=%00 HTTP/1.1
Cookie: ASP.NET_SessionId=eu0qbsjngqgirw45q0opxa45; amSessionId=3546750533; amUserlnfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=; amUserId=100116014; amCreditOffer=CardType=GoldLimit=10000&Interest=7.9
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/bank/ws.asmx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)

HTTP/1.1 500 Internal Server Error
Date: Sun, 22 Jul 2012 09:06:49 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/plain; charset=utf-8
Content-Length: 44

XML Web service description was not found.
**Issue 5 of 15**

**Application Error**

**Severity:** Informational

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** debitAccount (Parameter)

**Risk:** It is possible to gather sensitive debugging information

**Causes:**
- Proper bounds checking were not performed on incoming parameter values
- No validation was done in order to make sure that user input matches the data type expected

**Fix:**
- Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```html
... 
</tr> 
</tr> 
<td colspan="2" align="center"><input type="button" name="transfer" value="Transfer Money" onclick="doTransfer();" ID="transfer"></td> 
</tr> 
<td colspan="2">&nbsp;</td> 
</tr> 
<td colspan="2" align="center"> 
<span id="_ctl0__ctl0_Content_Main_postResp" align="center"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression 'accountid='.
 at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
 at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line 146</span></span> 
</td> 
</tr> 
... 
```

**Issue 6 of 15**

...
Application Error

**Severity:** Informational

**URL:** http://demo.testfire.net/bank/transfer.aspx

**Entity:** creditAccount (Parameter)

**Risk:** It is possible to gather sensitive debugging information

**Causes:** Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```
... 
<tr>
    <td colspan="2" align="center"><input type="button" name="transfer" value="Transfer Money" onclick="doTransfer();" ID="transfer"></td>
</tr>
<tr>
    <td colspan="2"><nbsp;</td>
</tr>
<tr>
    <td colspan="2" align="center">
    <span id="_ctl0__ctl0_Content_Main_postResp" align="center" style='color: Red'>System.Data.OleDb.OleDbException: Syntax error in string expression 'accountid='.
    at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
    at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line 155
</span>
</td>
</tr>
... 
```

**Issue 7 of 15**
Application Error

Severity: Informational

URL: http://demo.testfire.net/bank/login.aspx

Entity: uid (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

...
Issue 9 of 15

**Application Error**

**Severity:** Informational

**URL:** http://demo.testfire.net/bank/login.aspx

**Entity:** passw (Parameter)

**Risk:** It is possible to gather sensitive debugging information

**Causes:**
- Proper bounds checking were not performed on incoming parameter values
- No validation was done in order to make sure that user input matches the data type expected

**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```plaintext
...
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/bank/login.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 36
uid=jsmith&passw=%27&btnSubmit=Login
HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:40:53 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1

...
```

Issue 10 of 15

10/2/2012
Application Error

Severity: Informational

URL: http://demo.testfire.net/bank/ws.asmx

Entity: [SOAP] creditAccount_2 (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:37:03 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1207

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException: Syntax error in query expression 'accountid=82'.
  at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)

Issue 11 of 15

Application Error

Severity: Informational

URL: http://demo.testfire.net/bank/ws.asmx

Entity: [SOAP] debitAccount_1 (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values
No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

...
Application Error

Severity: Informational

URL: http://demo.testfire.net/bank/ws.asmx

Entity: [SOAP] transferDate (Parameter)

Risk: It is possible to gather sensitive debugging information

Causes: Proper bounds checking were not performed on incoming parameter values

No validation was done in order to make sure that user input matches the data type expected

Fix: Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

Reasoning: The application has responded with an error message, indicating an undefined state that may expose sensitive information.

Raw Test Response:

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:36:55 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-Apache-Version: 2.0.50727
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1207

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)

HTTP/1.1 500 Internal Server Error
Date: Sun, 22 Jul 2012 08:36:48 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-Apache-Version: 2.0.50727
Cache-Control: private
Content-Type: text/xml; charset=utf-8
Content-Length: 481

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
<soap:Body><soap:Fault><faultcode>soap:Client</faultcode><faultstring>Server was unable to read request. --- &gt; There is an error in XML document (8, 37). --- &gt; The string '\"%27\"' is not a valid AllXsd value.</faultstring><detail /></soap:Fault></soap:Body></soap:Envelope>
**Issue 13 of 15**

### Application Error

**Severity:** Informational

**URL:** http://demo.testfire.net/bank/ws.asmx

**Entity:** [SOAP] transferAmount_3 (Parameter)

**Risk:** It is possible to gather sensitive debugging information

**Causes:**
- Proper bounds checking were not performed on incoming parameter values
- No validation was done in order to make sure that user input matches the data type expected

**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```xml
<soap:Body>
<soap:Fault>
<faultcode>soap:Client</faultcode><faultstring>Server was unable to read request. ---&gt; There is an error in XML document (11, 41). ---&gt; Input string was not in a correct format.</ faultstring><detail />
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

...
**Application Error**

**Severity:** Informational

**URL:** http://demo.testfire.net/bank/transaction.aspx

**Entity:** before (Parameter)

**Risk:** It is possible to gather sensitive debugging information

**Causes:**
- Proper bounds checking were not performed on incoming parameter values
- No validation was done in order to make sure that user input matches the data type expected

**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**

```
... 
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://demo.testfire.net/bank/transaction.aspx
Host: demo.testfire.net
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.5.30729)
Content-Length: 175

__EVENTTARGET__=__EVENTARGUMENT__=__VIEWSTATE__='@2PwEPdwuHNMTyZNDg3O4NnRkA__EVENTVALIDATION__='@2PwEMbhKV3oXKdBk3ocuaBAK3oaaDSgK3o2PRbK3oa9ZBJCg3uQAQ33533aafter=1234before=427

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:30:26 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1

... 
```

---

**Issue 15 of 15**

**Application Error**

**Severity:** Informational

**URL:** http://demo.testfire.net/bank/transaction.aspx

**Entity:** after (Parameter)

**Risk:** It is possible to gather sensitive debugging information

**Causes:**
- Proper bounds checking were not performed on incoming parameter values
- No validation was done in order to make sure that user input matches the data type expected

**Fix:** Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions

**Reasoning:** The application has responded with an error message, indicating an undefined state that may expose sensitive information.

**Raw Test Response:**
Issue 1 of 1

Application Test Script Detected

Severity: Informational

URL: http://demo.testfire.net/survey_questions.aspx

Entity: test.aspx (Page)

Risk: It is possible to download temporary script files, which can expose the application logic and other sensitive information such as usernames and passwords

Causes: Temporary files were left in production environment

Fix: Remove test scripts from the server

Reasoning: AppScan requested a file which is probably not a legitimate part of the application. The response status was 200 OK. This indicates that the test succeeded in retrieving the content of the requested file.

Test Request: __EVENTTARGET__=__EVENTARGUMENT__=__VIEWSTATE__=12PFwGwFwVHMTsNd830TAYmKpRw__EVENTVALIDATION__=12PwEMpghV3oX6Dgk3oaewzAR3oaeaDgK3oIPWSQ3oaA4-Zb7qRY3uAgQh3o34after=924 before=1234

Test Response

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:30:15 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Content-Type: text/html; charset=iso-8859-1

...
Issue 1 of 3

Email Address Pattern Found

Severity: Informational

URL: http://demo.testfire.net/subscribe.aspx

Entity: subscribe.aspx (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

... 

<h1>Subscribe</h1>

<p>We recognize that things are always evolving and changing here at Altoro Mutual.

10/2/2012

GET /test.aspx HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, application/xaml+xml,
application/vnd.ms-xpsdocument, application/x-ms-xbap,
application/x-ms-application, application/vnd.ms-excel,
application/vnd.ms-powerpoint, application/vnd.ms-word,
application/xml,*/*
Referer: http://demo.testfire.net/survey_questions.aspx?step=a
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1;
Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Connection: Keep-Alive
Host: demo.testfire.net
Cookie: ASP.NET_SessionId=wh542tn0duonh55nkqt.nr55;
amSessionId=334738728;
amUserInfo=UserName=anNtaXRo&Password=ZGVtbzEyMzQ=;
amUserId=100116014;
amCreditOffer=CardType=Gold&Limit=10000&Interest=7.9

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:20:32 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 558

...
Email Address Pattern Found

Severity: Informational

URL: http://demo.testfire.net/survey_complete.aspx

Entity: survey_complete.aspx (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Insecure web application programming or configuration

Fix: Remove e-mail addresses from the website

Reasoning: The response contains an e-mail address that may be private.

Raw Test Response:

...
<table>
<thead>
<tr>
<th>Email Address Pattern Found</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Severity:</strong> Informational</td>
</tr>
<tr>
<td><strong>URL:</strong> <a href="http://demo.testfire.net/bank/mozxpath.js">http://demo.testfire.net/bank/mozxpath.js</a></td>
</tr>
<tr>
<td><strong>Entity:</strong> mozxpath.js (Page)</td>
</tr>
<tr>
<td><strong>Risk:</strong> It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations</td>
</tr>
<tr>
<td><strong>Causes:</strong> Insecure web application programming or configuration</td>
</tr>
<tr>
<td><strong>Fix:</strong> Remove e-mail addresses from the website</td>
</tr>
</tbody>
</table>

**Reasoning:** The response contains an e-mail address that may be private.

**Raw Test Response:**

```html
... Content-length: 1414 Content-Type: application/x-javascript Last-Modified: Thu, 13 Jan 2011 04:14:33 GMT Accept-Ranges: bytes ETag: "9670cb61d8b2cb1:104e" Server: Microsoft-IIS/6.0 X-Powered-By: ASP.NET Date: Sun, 22 Jul 2012 08:03:08 GMT
// mozXPath [http://km0ti0n.blunted.co.uk/mozxpath/] km0ti0n@gmail.com // Code licensed under Creative Commons Attribution-ShareAlike License // http://creativecommons.org/licenses/by-sa/2.5/ if( document.implementation.hasFeature("XPath", "3.0") ) {
  XMLDocument.prototype.selectNodes = function(cXPathString, xNode)
  {
    if( !xNode ) { xNode = this; }

    var oNSResolver = this.createNSResolver(this.documentElement)
  ...
```

### Issue 1 of 5

**HTML Comments Sensitive Information Disclosure**

**Issue 1 of 5**

10/2/2012
TOC

HTML Comments Sensitive Information Disclosure

Severity: Informational

URL: http://demo.testfire.net/bank/account.aspx

Entity: To modify account information do not connect to SQL source directly. Make all changes (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Debugging information was left by the programmer in web pages

Fix: Remove sensitive information from HTML comments

Reasoning: AppScan discovered HTML comments containing what appears to be sensitive information.

Original Response

...  
<br style="line-height: 10px;"/>
<br/>
<ol class="sidebar">
<li><a id="_ctl0__ctl0_Content_MenuHyperLink1" href="main.aspx">View Account Summary</a></li>
<li><a id="_ctl0__ctl0_Content_MenuHyperLink2" href="transaction.aspx">View Recent Transactions</a></li>
<li><a id="_ctl0__ctl0_Content_MenuHyperLink3" href="transfer.aspx">Transfer Funds</a></li>
<li><a id="_ctl0__ctl0_Content_MenuHyperLink4" href="queryxpath.aspx">Search News Articles</a></li>
<li><a id="_ctl0__ctl0_Content_MenuHyperLink5" href="customize.aspx">Customize Site Language</a></li>
</ol>

Reasoning: AppScan discovered HTML comments containing what appears to be sensitive information.

Issue 2 of 5
<div class="f1" style="width: 99%;">
<h1>Online Banking Login</h1>

<!-- To get the latest admin login, please contact SiteOps at 415-555-6159 -->

<form action="login.aspx" method="post" name="login" id="login" onsubmit="return (confirminput(login));">
<table>
<tr>
<td><strong>Username:</strong></td>
<td><input type="text" id="uid" name="uid" value="jsmith" style="width: 150px;" />
</td>
<td></td>
</tr>
<tr>
<td><strong>Password:</strong></td>
<td><input type="password" id="passw" name="passw" style="width: 150px;" />
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>
<input type="submit" name="btnSubmit" value="Login" />
</td>
</tr>
</table>
</form>

<script>
function setfocus() {
  if (document.login.uid.value=="") {
    document.login.uid.focus();
  } else {
    document.login.passw.focus();
  }
}

function confirminput(myform) {
  if (myform.uid.value.length && myform.passw.value.length) {
    return (true);
  } else if (!myform.uid.value.length) {
    myform.reset();
    myform.uid.focus();
    alert ("You must enter a valid username");
    return (false);
  } else {
    myform.passw.focus();
    alert ("You must enter a valid password");
    return (false);
  }
}

window.onload = setfocus;
</script>
</div>
HTML Comments Sensitive Information Disclosure

Severity: Informational

URL: http://demo.testfire.net/bank/apply.aspx

Entity: Password is not revalidated but stored in (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Debugging information was left by the programmer in web pages

Fix: Remove sensitive information from HTML comments

Reasoning: AppScan discovered HTML comments containing what appears to be sensitive information.

Original Response

...<br style="line-height: 10px;"/>
<b>I WANT TO ...</b>
<ul class="sidebar">
  <li><a id="_ctl0__ctl0_Content_MenuHyperLink1" href="main.aspx">View Account Summary</a></li>
  <li><a id="_ctl0__ctl0_Content_MenuHyperLink2" href="transaction.aspx">View Recent Transactions</a></li>
  <li><a id="_ctl0__ctl0_Content_MenuHyperLink3" href="transfer.aspx">Transfer Funds</a></li>
  <li><a id="_ctl0__ctl0_Content_MenuHyperLink4" href="queryxpath.aspx">Search News Articles</a></li>
  <li><a id="_ctl0__ctl0_Content_MenuHyperLink5" href="customize.aspx">Customize Site Language</a></li>
</ul><span id="_ctl0__ctl0_Content_Administration"></span>
<td valign="top" colspan="3" class="bb">
<div class="fl" style="width: 99%;">
<h1>Altoro Mutual</h1>
<span id="_ctl0__ctl0_Content_Main_lblType">Gold</span> Visa Application</h1>
...<br/>

userid = userCookie.Values["UserID"].ToString();
cLimit = Request.Cookies["Limit"].Value;
cInterest = Request.Cookies["Interest"].Value;
cType = Request.Cookies["CardType"].Value;
-->

Password is not revalidated but stored in mainframe for non-repudiation purposes.
-->
</div>
</td>
</tr>
</table>...
HTML Comments Sensitive Information Disclosure

Severity: Informational

URL: http://demo.testfire.net/admin/admin.aspx

Entity: Be careful what you change. All changes are made directly to Altoro.mdb database. (Page)

Risk: It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations

Causes: Debugging information was left by the programmer in web pages

Fix: Remove sensitive information from HTML comments

Reasoning: AppScan discovered HTML comments containing what appears to be sensitive information.

Original Response

...
if (myform.password1.value.length && (myform.password1.value==myform.password2.value))
    return true;
else
    myform.password1.value="";
    myform.password2.value="";
    myform.password1.focus();
    alert ("Passwords do not match");
    return false;
}
</script>
**Issue 5 of 5**

**HTML Comments Sensitive Information Disclosure**

**Severity:** Informational

**URL:** http://demo.testfire.net/admin/login.aspx

**Entity:** Password: Altoro1234 (Page)

**Risk:** It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations.

**Causes:** Debugging information was left by the programmer in web pages

**Fix:** Remove sensitive information from HTML comments

**Reasoning:** AppScan discovered HTML comments containing what appears to be sensitive information.

**Original Response**

```html
... 
Content-Type: text/html; charset=iso-8859-1
Content-Length: 8215

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head id="_ctl0__ctl0_head"><title>Altoro Mutual: Administration</title><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><link href="../style.css" rel="stylesheet" type="text/css" /></head>
<body style="margin-top:5px;">
<div id="header" style="margin-bottom:5px; width: 99%;">
<form id="frmSearch" method="get" action="/search.aspx">
<table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr>
<td rowspan="2"><a id="_ctl0__ctl0_HyperLink1" href="../default.aspx" style="height:80px;width:183px;"
<img src="../images/logo.gif" border="0" /></a></td>
<td align="right" valign="top">
... ...
</td>
</tr>
</table>
</form>
</div>
<div id="wrapper" style="width: 99%;">
<table cellspacing="0" width="100%">
<tr>
<td width="25%" class="bt br bb"><div id="Header1"><img id="_ctl0__ctl0_Content_Image1" src="../images/pf_lock.gif" alt="Secure Login" align="absbottom" border="0" style="height:14px;width:12px;" /> &nbsp; <a id="_ctl0__ctl0_Content_AccountLink" title="You do not appear to have authenticated yourself with the application.  Click here to enter your valid username and password." class="focus" href="../bank/login.aspx" style="font-weight:700; font-size:10px;">ONLINE BANKING LOGIN</a></div></td>
<td width="25%" class="cc bt br bb"><div id="Header2"><a id="_ctl0__ctl0_Content_LinkHeader2" class="focus" href="../default.aspx?content=personal.htm">PERSONAL</a></div></td>
<td width="25%" class="cc bt bb"><div id="Header3"><a id="_ctl0__ctl0_Content_LinkHeader3" class="focus" href="../default.aspx?content=business.htm">SMALL BUSINESS</a></div></td>
<td width="25%" class="cc bt bb"><div id="Header4"><a id="_ctl0__ctl0_Content_LinkHeader4" class="focus" href="../default.aspx?content=personal.htm">PERSONAL</a></div></td>
</tr>
</table>
... 
</div>
</body>
</html>
```
Possible Server Path Disclosure Pattern Found

Issue 1 of 1

10/2/2012
### Possible Server Path Disclosure Pattern Found

**Severity:** Informational

**URL:** http://demo.testfire.net/feedback.aspx

**Entity:** feedback.aspx (Page)

**Risk:** It is possible to retrieve the absolute path of the web server installation, which might help an attacker to develop further attacks and to gain information about the file system structure of the web application.

**Causes:** Latest patches or hotfixes for 3rd. party products were not installed

**Fix:** Download the relevant security patch for your web server or web application.

---

**Reasoning:** The response contains the absolute paths and/or filenames of files on the server.

**Raw Test Response:**

```html
...<p>Our Frequently Asked Questions area will help you with many of your inquiries.<br />
If you can't find your question, return to this page and use the e-mail form below.</p>
<p><b>IMPORTANT!</b> This feedback facility is not secure. Please do not send any account information in a message sent from here.</p>
<form name="cmt" method="post" action="comment.aspx">
<--- Dave- Hard code this into the final script - Possible security problem.<br />
Re-generated every Tuesday and old files are saved to .bak format at L:\backup\website\oldfiles  ---->
<input type="hidden" name="cfile" value="comments.txt">
<table border=0>
<tr>
<td align=right>To:</td>
<td valign=top><b>Online Banking</b></td>
</tr>
<tr>
<td align=right>Your Name:</td>
<td>...</td>
</tr>
</table>
...```

10/2/2012