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: 99
Scan started: 22/07/2012 09:12:42 AM

Table of Contents

Introduction

Executive Summary

Issues Sorted by Issue Type

- HTML Comments Sensitive Information Disclosure 5
- Possible Server Path Disclosure Pattern Found 1

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:	22/07/2012 09:12:42 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 32

TOC

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

Vulnerable URLs 29

TOC

URL	Number of Issues

	Root	0
H	http://demo.testfire.net/bank/login.aspx	22
H	http://demo.testfire.net/bank/account.aspx	5
H	http://demo.testfire.net/bank/apply.aspx	4
H	http://demo.testfire.net/bank/customize.aspx	8
H	http://demo.testfire.net/bank/transfer.aspx	16
H	http://demo.testfire.net/comment.aspx	5
H	http://demo.testfire.net/search.aspx	3
H	http://demo.testfire.net/subscribe.aspx	7
H	http://demo.testfire.net/survey_complete.aspx	5
H	http://demo.testfire.net/disclaimer.htm	4
H	http://demo.testfire.net/high_yield_investments.htm	1
H	http://demo.testfire.net/default.aspx	1
H	http://demo.testfire.net/bank/transaction.aspx	12
H	http://demo.testfire.net/bank/ws.asmx	9
H	http://demo.testfire.net/admin/admin.aspx	5
H	http://demo.testfire.net/admin/login.aspx	5
H	http://demo.testfire.net/bank/queryxpath.aspx	4
M	http://demo.testfire.net/bank/	1
M	http://demo.testfire.net/pr/	1
L	http://demo.testfire.net/admin/clients.xls	2
L	http://demo.testfire.net/survey_questions.aspx	3
L	http://demo.testfire.net/admin/	1
L	http://demo.testfire.net/aspnet_client/	1
L	http://demo.testfire.net/images/	1
L	http://demo.testfire.net/bank/main.aspx	1
L	http://demo.testfire.net/	1
I	http://demo.testfire.net/bank/mozxpath.js	1
I	http://demo.testfire.net/feedback.aspx	1

Fix Recommendations 23

TOC

	Remediation Task	Number of Issues
H	Always use SSL and POST (body) parameters when sending sensitive information.	6
H	Analyze client side code and sanitize its input sources	3
H	Change the login credentials to a stronger combination	1
H	Ensure that accessed files reside in the virtual path and have certain extensions; remove special characters from user input	1
H	Review possible solutions for hazardous character injection	55
M	Analyze and harden client side (JavaScript) code.	2
M	Decline malicious requests	6
M	Do not accept externally created session identifiers	1
M	Enforce account lockout after several failed login attempts	1
M	Modify the server configuration to deny directory listing, and install the latest security patches available	2
L	Add the 'HttpOnly' attribute to all session cookies	4

L	Apply proper authorization to administration scripts	2
L	Avoid storing sensitive session information in permanent cookies	1
L	Correctly set the "autocomplete" attribute to "off"	4
L	Disable Debugging on Microsoft ASP.NET	3
L	Download the relevant security patch for your web server or web application.	1
L	Issue a "404 - Not Found" response status code for a forbidden resource, or remove it completely	3
L	Modify the property of each ASP.NET page to sign the VIEWSTATE parameter	4
L	Modify your Web.Config file to encrypt the VIEWSTATE parameter	4
L	Remove e-mail addresses from the website	5
L	Remove sensitive information from HTML comments	5
L	Remove test scripts from the server	1
L	Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions	15

Security Risks 19

TOC

	Risk	Number of Issues
H	It may be possible to bypass the web application's authentication mechanism	5
H	It is possible to view, modify or delete database entries and tables	29
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	32
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)	1
H	It might be possible to escalate user privileges and gain administrative permissions over the web application	4
H	It may be possible to steal user login information such as usernames and passwords that are sent unencrypted	6
H	It is possible to access information stored in a sensitive data resource	1
M	It is possible to view and download the contents of certain web application virtual directories, which might contain restricted files	2
M	It is possible to deface the site content through web-cache poisoning	1
M	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.	12
M	It is possible to upload, modify or delete web pages, scripts and files on the web server	6
M	It is possible for an attacker to use the web server to attack other sites, which increases his or her anonymity	2
L	It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations	17
L	It is possible to retrieve information about the site's file system structure, which may help the attacker to map the web site	3
L	It may be possible to steal session information (cookies) that was kept on disk as permanent cookies	1
L	It might be possible to undermine application logic	4
I	It is possible to gather sensitive debugging information	15
I	It is possible to download temporary script files, which can expose the application logic and other sensitive information such as usernames and passwords	1
I	It is possible to retrieve the absolute path of the web server installation, which might	1

help an attacker to develop further attacks and to gain information about the file system structure of the web application

Causes 16

TOC

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

WASC Threat Classification

TOC

Threat	Number of Issues
Abuse of Functionality	4
Application Privacy Tests	14
Application Quality Tests	15
Brute Force	2
Content Spoofing	12
Cross-site Request Forgery	6
Cross-site Scripting	14
Directory Indexing	2
HTTP Response Splitting	1
Information Leakage	21
Insufficient Authentication	1
Insufficient Session Expiration	1
Null Byte Injection	1
Predictable Resource Location	3
Session Fixation	1

SQL Injection	29
URL Redirector Abuse	2
XPath Injection	1

Issues Sorted by Issue Type

H Authentication Bypass Using SQL Injection 1

TOC

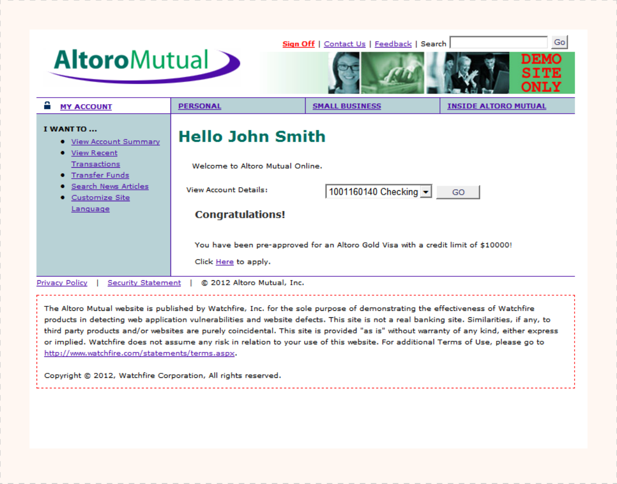
Issue 1 of 1

TOC

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 the response to the valid login.

Valid Login



Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

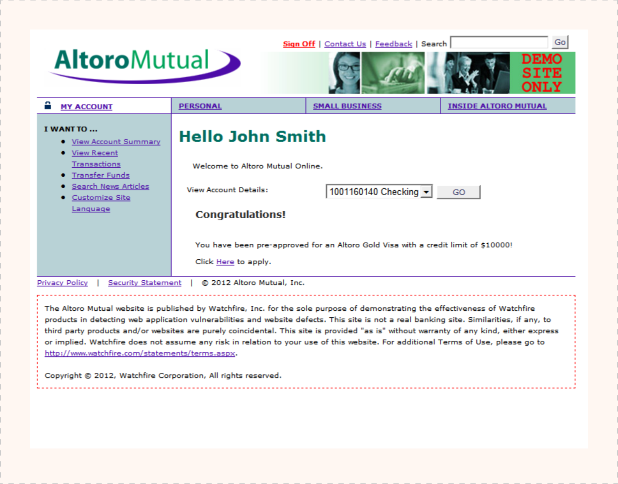
© 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

≈

Test Login



Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

© 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

H Blind SQL Injection 1

TOC

Issue 1 of 1

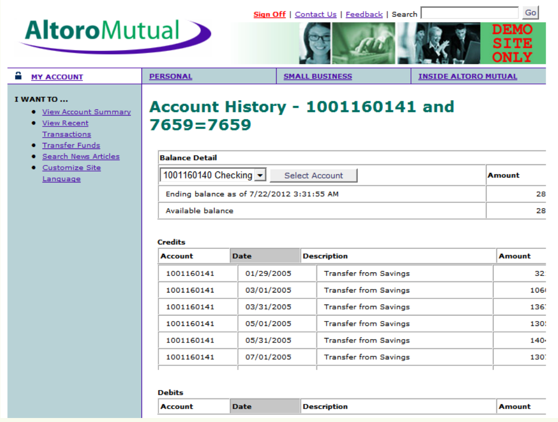
TOC

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.HEX(0D)HEX(0A)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.

Original Response



Account History - 1001160141 and 7659=7659

Balance Detail	Amount
Ending balance as of 7/22/2012 3:31:55 AM	28
Available balance	28

Credits

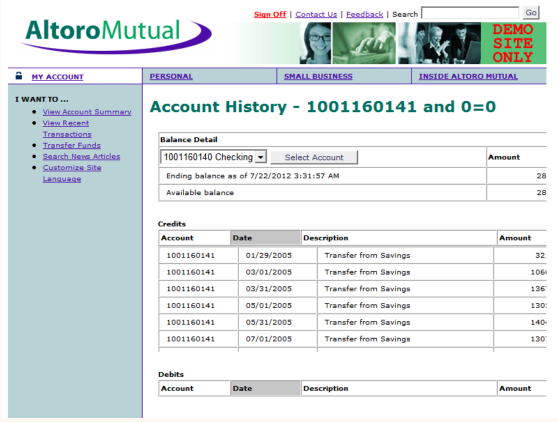
Account	Date	Description	Amount
1001160141	01/29/2005	Transfer from Savings	32
1001160141	03/01/2005	Transfer from Savings	106
1001160141	03/31/2005	Transfer from Savings	136
1001160141	05/01/2005	Transfer from Savings	130
1001160141	05/31/2005	Transfer from Savings	140
1001160141	07/01/2005	Transfer from Savings	130

Debits

Account	Date	Description	Amount

≈

Test Response (last)



Account History - 1001160141 and 0=0

Balance Detail	Amount
Ending balance as of 7/22/2012 3:31:57 AM	28
Available balance	28

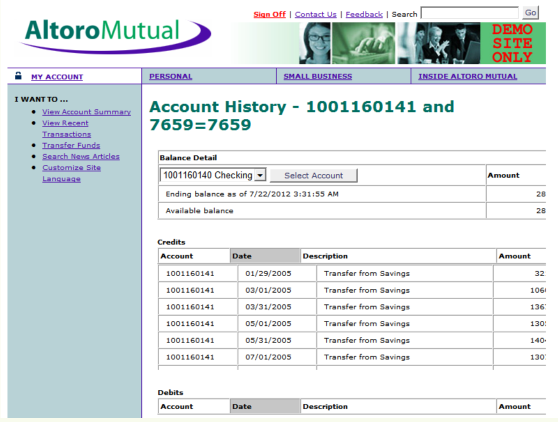
Credits

Account	Date	Description	Amount
1001160141	01/29/2005	Transfer from Savings	32
1001160141	03/01/2005	Transfer from Savings	106
1001160141	03/31/2005	Transfer from Savings	136
1001160141	05/01/2005	Transfer from Savings	130
1001160141	05/31/2005	Transfer from Savings	140
1001160141	07/01/2005	Transfer from Savings	130

Debits

Account	Date	Description	Amount

Original Response


Sign Off | Contact Us | Feedback | Search
Go
AltoroMutual
DEMO SITE ONLY
MY ACCOUNT
PERSONAL
SMALL BUSINESS
INSIDE ALTORO MUTUAL
I WANT TO ...
View Account Summary
View Recent Transactions
Transfer Funds
Search News Articles
Customize Site Language
1001160140 Checking
Select Account

Account History - 1001160141 and 7659=7659

Balance Detail	Amount
Ending balance as of 7/22/2012 3:31:55 AM	28
Available balance	28

Credits

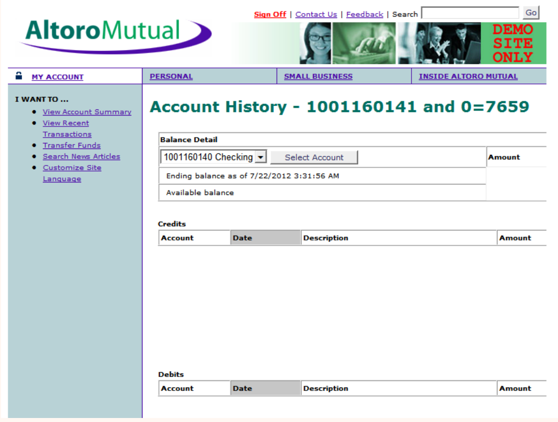
Account	Date	Description	Amount
1001160141	01/29/2005	Transfer from Savings	32
1001160141	03/01/2005	Transfer from Savings	106
1001160141	03/31/2005	Transfer from Savings	136
1001160141	05/01/2005	Transfer from Savings	130
1001160141	05/31/2005	Transfer from Savings	140
1001160141	07/01/2005	Transfer from Savings	130

Debits

Account	Date	Description	Amount

≠

Test Response (next-to-last)



Account History - 1001160141 and 0=7659

Balance Detail	Amount
Ending balance as of 7/22/2012 3:31:56 AM	
Available balance	

Credits

Account	Date	Description	Amount

Debits

Account	Date	Description	Amount

H Cross-Site Scripting 11 TOC

Issue 1 of 11

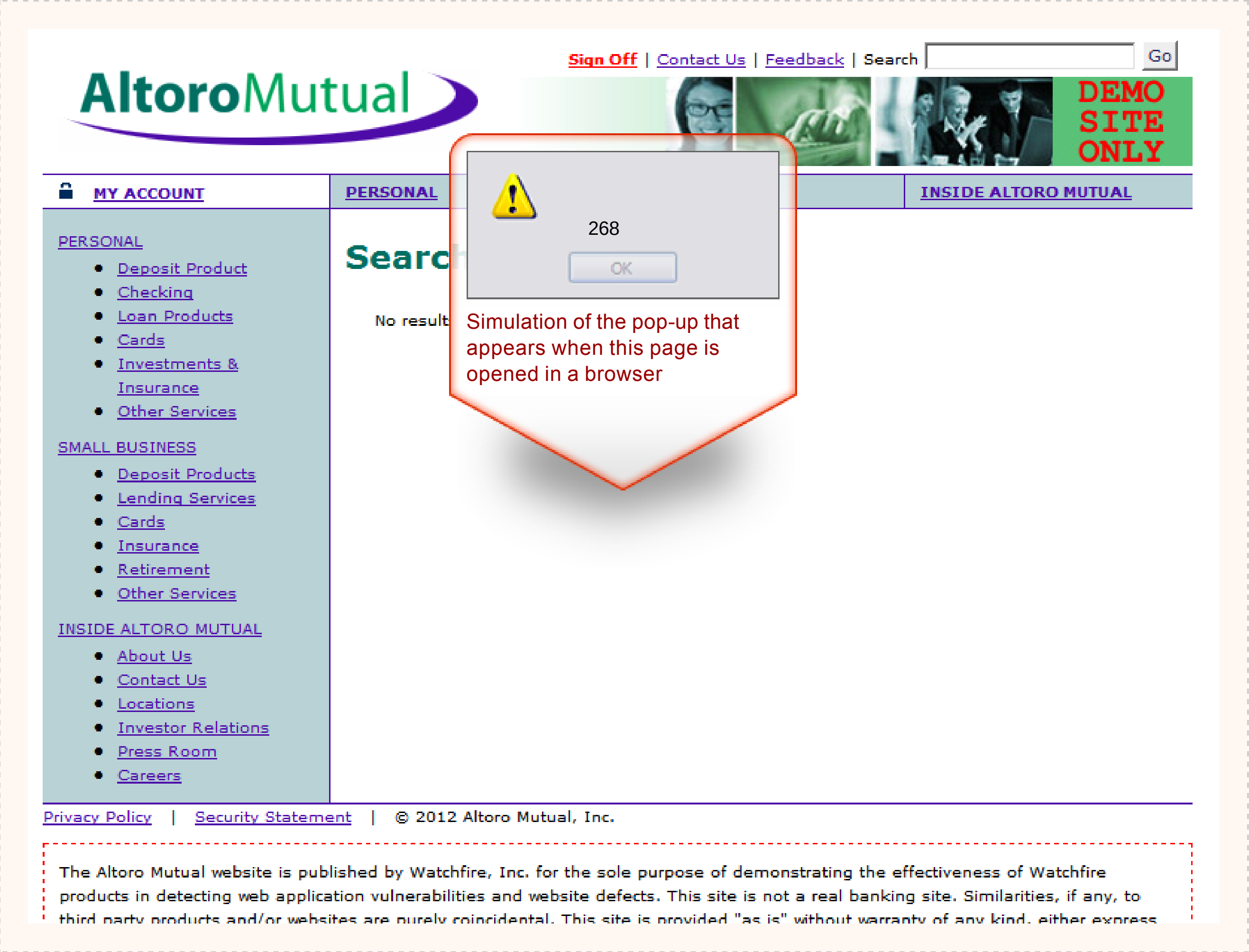
TOC

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



268

No result

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

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 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:

```
...
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 08:27:41 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: 7261



<!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: Search Results

...

...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<h1>Search Results</h1>

<p>No results were found for the query:<br /><br />
<span id="_ctl0__ctl0_Content_Main_lblSearch"><script>alert(1727)</script></span></p>

</div>


    </td>
  </tr>
</table>




...
```

Issue 2 of 11

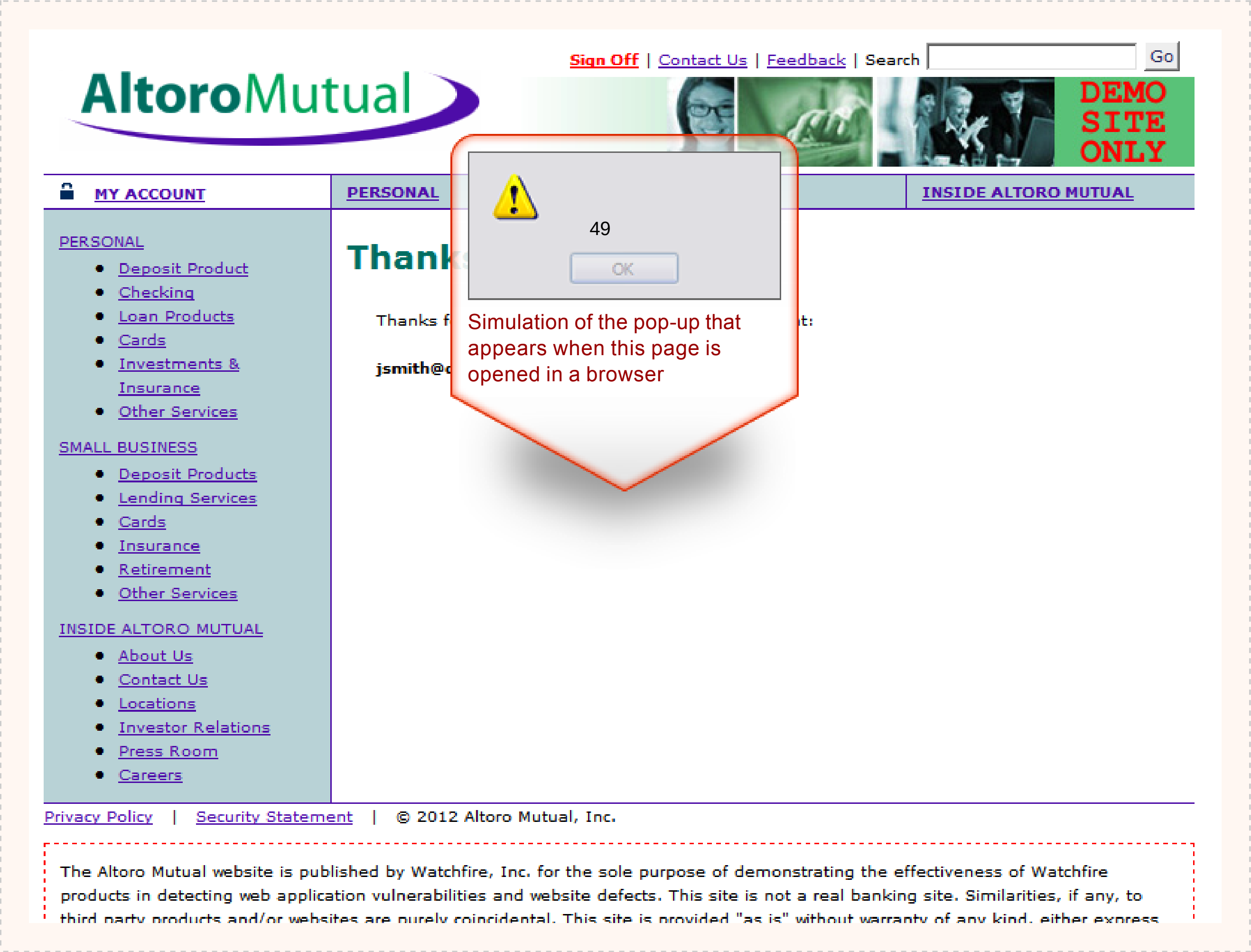
TOC

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



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

© 2012 Altoro Mutual, Inc.

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 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:

```
...

            <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 />
<b>jsmith@demo.testfire.net<script>alert(18)</script></b></p></span>

</div>


    </td>
  </tr>
</table>



...
```

Issue 3 of 11

TOC

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:

```
...


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

...

...

        </ul>
    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

 <h1>Thank You</h1>

 <p>Thank you for your comments, 1234'"><iframe src=javascript:alert(13)>.  They will be reviewed by our Customer Service staff and
 given the full attention that they deserve.</p>

</div>


    </td>
  </tr>
</table>


...
```

Issue 4 of 11

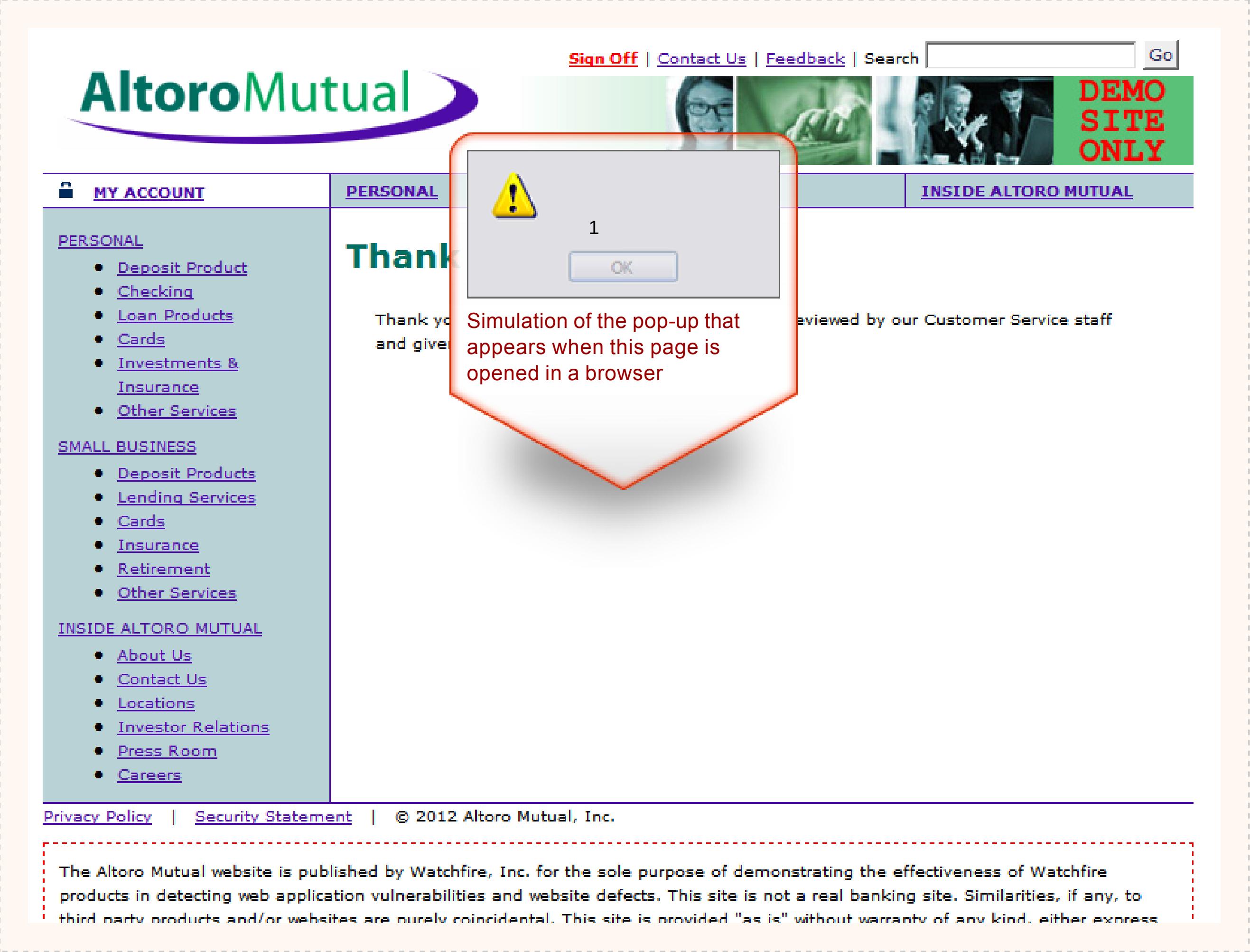
TOC

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



Thank

Thank yo eviewed by our Customer Service staff
and give

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

© 2012 Altoro Mutual, Inc.

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 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:

```
...


HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:04:02 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: 7219



<!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

...

...

        </ul>
    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

 <h1>Thank You</h1>

 <p>Thank you for your comments, >"'><script>alert(10)</script>.  They will be reviewed by our Customer Service staff and
 given the full attention that they deserve.</p>

</div>


    </td>
  </tr>
</table>


...
```

Issue 5 of 11

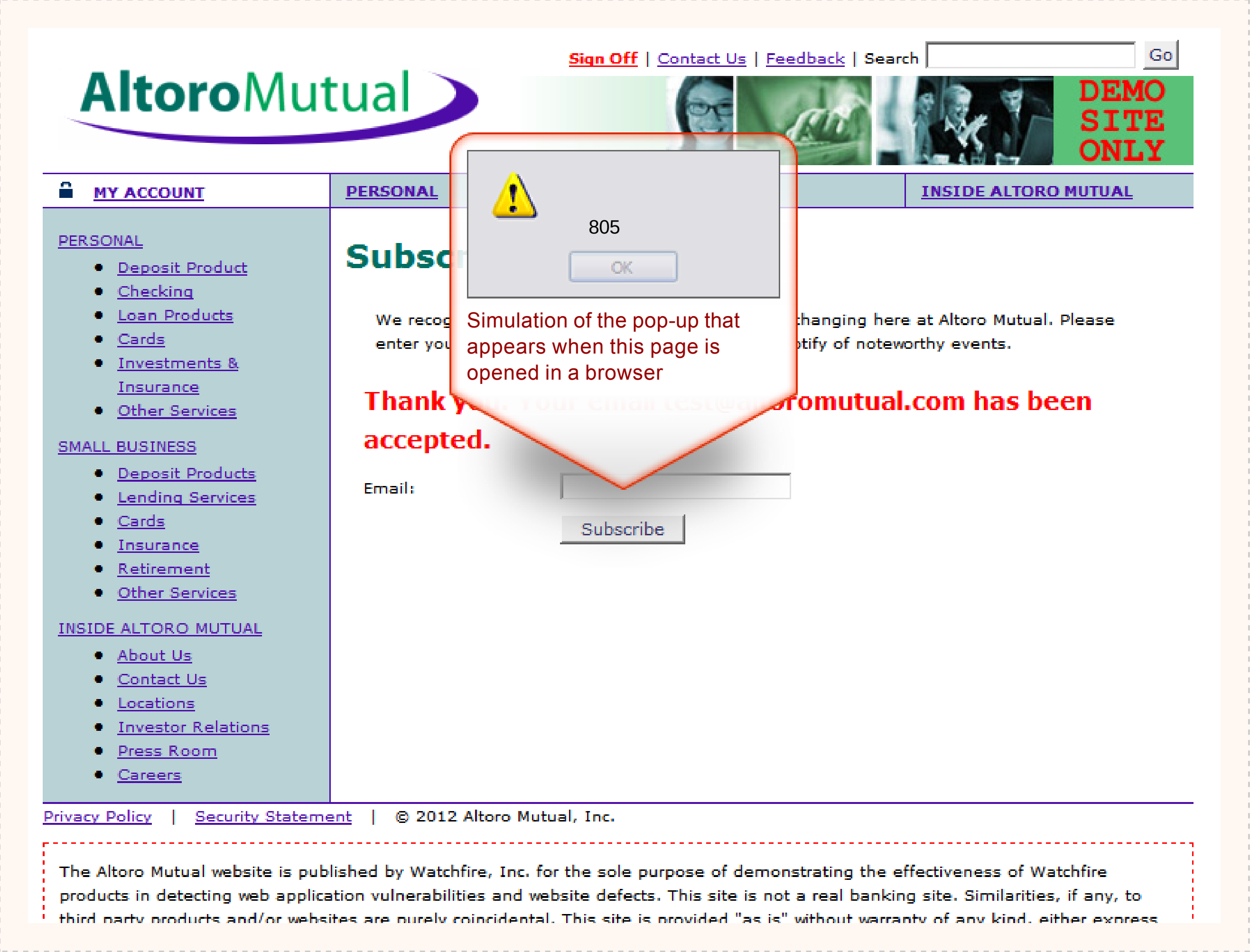
TOC

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



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

Subsc

We recog hanging here at Altoro Mutual. Please enter you otify of noteworthy events.

Thank y romutual.com has been accepted.

Email:

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 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:

```
...

<h1>Subscribe</h1>

<p>We recognize that things are always evolving and changing here at Altoro Mutual.
Please enter your email below and we will automatically notify of noteworthy events.</p>

<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<script>alert(130)</script> has been accepted.</span>
      </td>
    </tr>
    <tr>
      <td>
        Email:
      </td>
      <td>
        <input type="text" id="txtEmail" name="txtEmail" value="" style="width: 150px;">
      </td>

...
```

Issue 6 of 11

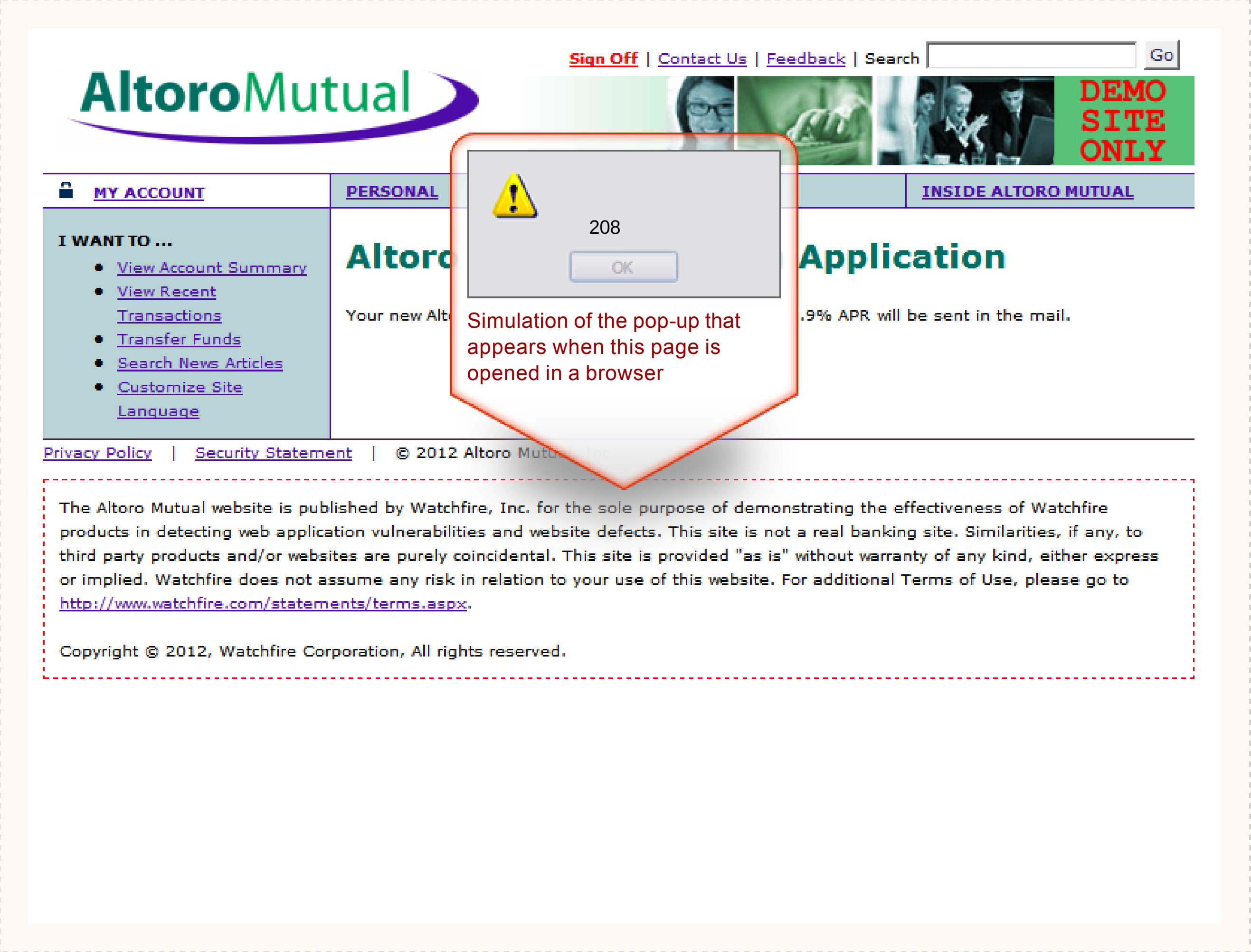
TOC

Cross-Site Scripting

Severity:	High
URL:	http://demo.testfire.net/bank/apply.aspx
Entity:	amCreditOffer (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:	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



I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Altoro ... Application

Your new Alt... .9% APR will be sent in the mail.

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

Privacy Policy | Security Statement | © 2012 Altoro Mutu...

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

Raw Test Response:

```
...

    Visa Application</h1>

<!--
    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">Your new Altoro Mutual Gold VISA with a $10000 and 7.9<script>alert(53)</script>% APR will
be sent in the mail.</span>

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

</div>



...
```

Issue 7 of 11

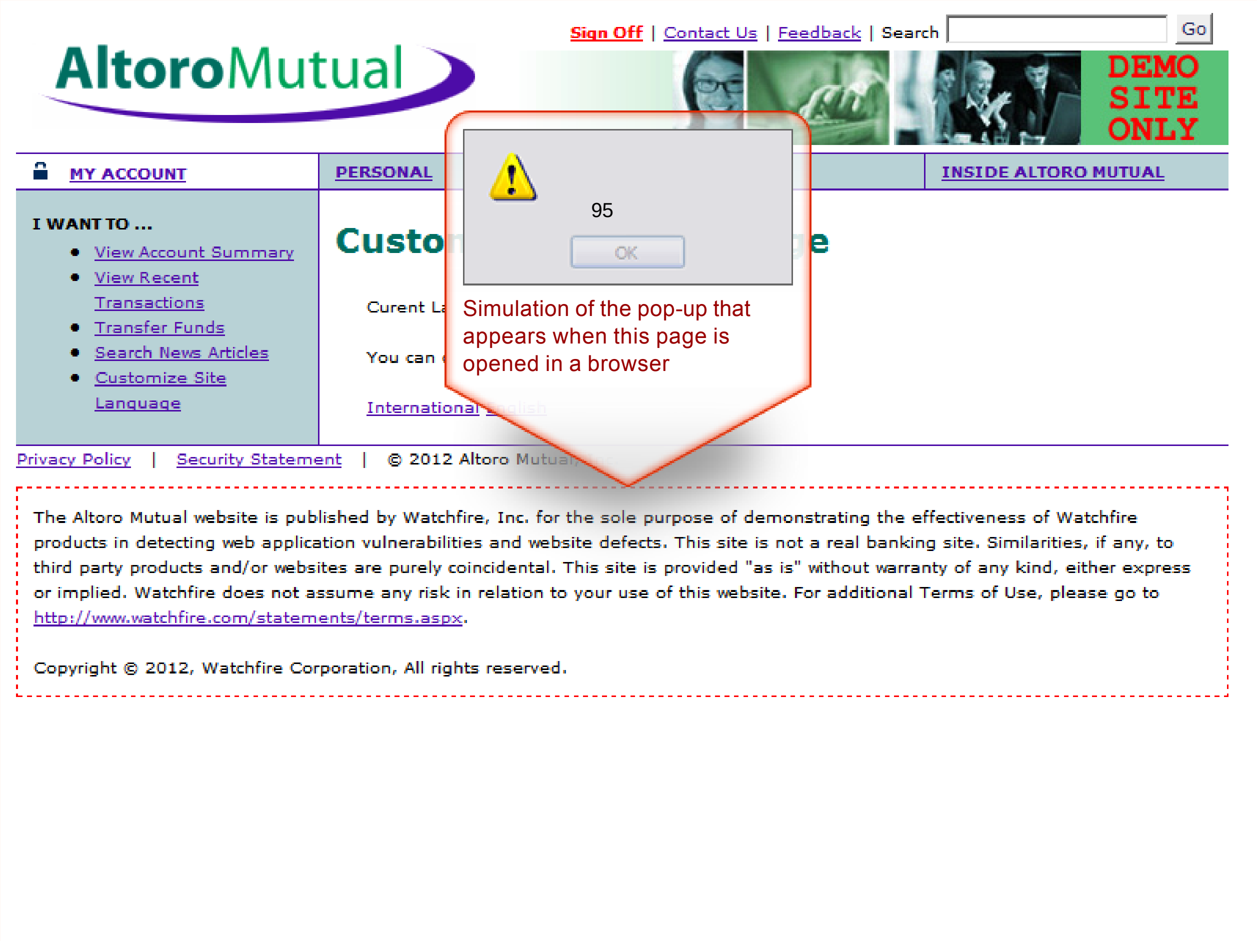
TOC

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



I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Custo e

Curent La

You can c

International English

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

Privacy Policy | Security Statement | © 2012 Altoro Mutual

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

Raw Test Response:

```
...


__VIEWSTATE=%2FwEPDwUJMjA2OTMxMDA4ZGQ%3D

HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:25:31 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: 5628
Set-Cookie: lang=>"'><script>alert(1539)</script>; path=/



<!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>

...

...

        <span id="_ctl0__ctl0_Content_Administration"></span>
    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">
```

```
<h1>Customize Site Language</h1>

<form name="aspnetForm" method="post" action="customize.aspx?lang=%3e%22'%3e%3cscript%3ealert(1539)%3c%2fscript%3e" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUJMjA2OTMxMDA4ZGQ=" />

  <p>
  <span id="_ctl0__ctl0_Content_Main_Label1">Curent Language: </span>
  <span id="_ctl0__ctl0_Content_Main_langLabel">>"'><script>alert(1539)</script></span>
  </p>

  <p>
  <span id="_ctl0__ctl0_Content_Main_Label2">You can change the language setting by choosing:</span>

...
```

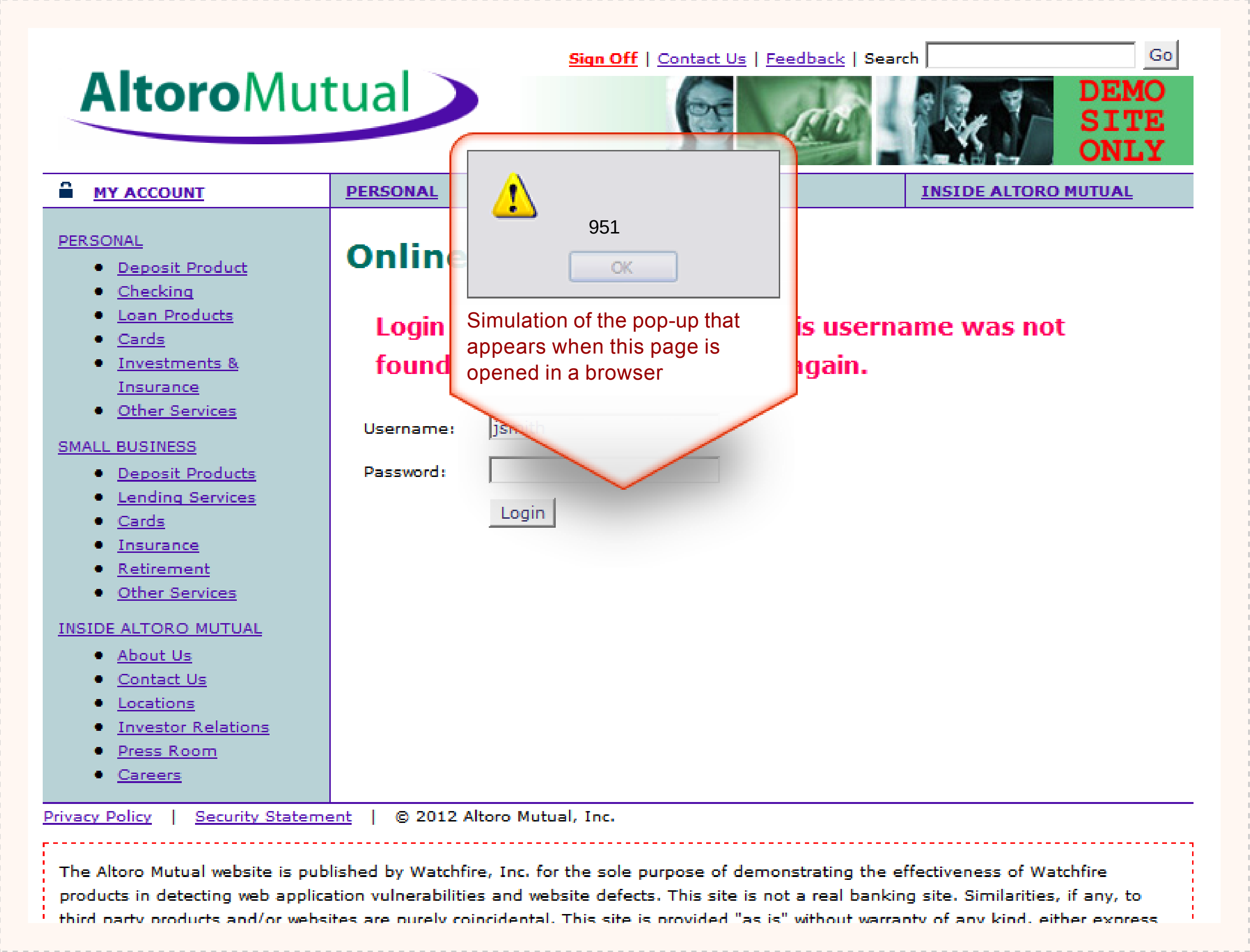
Issue 8 of 11

TOC

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 … is username was not found … again.

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

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 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:

```
...

<p><span id="_ctl0__ctl0_Content_Main_message" style="color:#FF0066;font-size:12pt;font-weight:bold;">Login Failed: We're sorry, but this
username was not found in our system.  Please try again.</span></p>

<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>
      <td>
      </td>
    </tr>
    <tr>
      <td>
        Password:
      </td>
      <td>

...
```

Issue 9 of 11

TOC

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

...

...

  </tr>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
in string in query expression 'accountid=1001160141'"><iframe src=javascript:alert(2434)>'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 10 of 11

TOC

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:

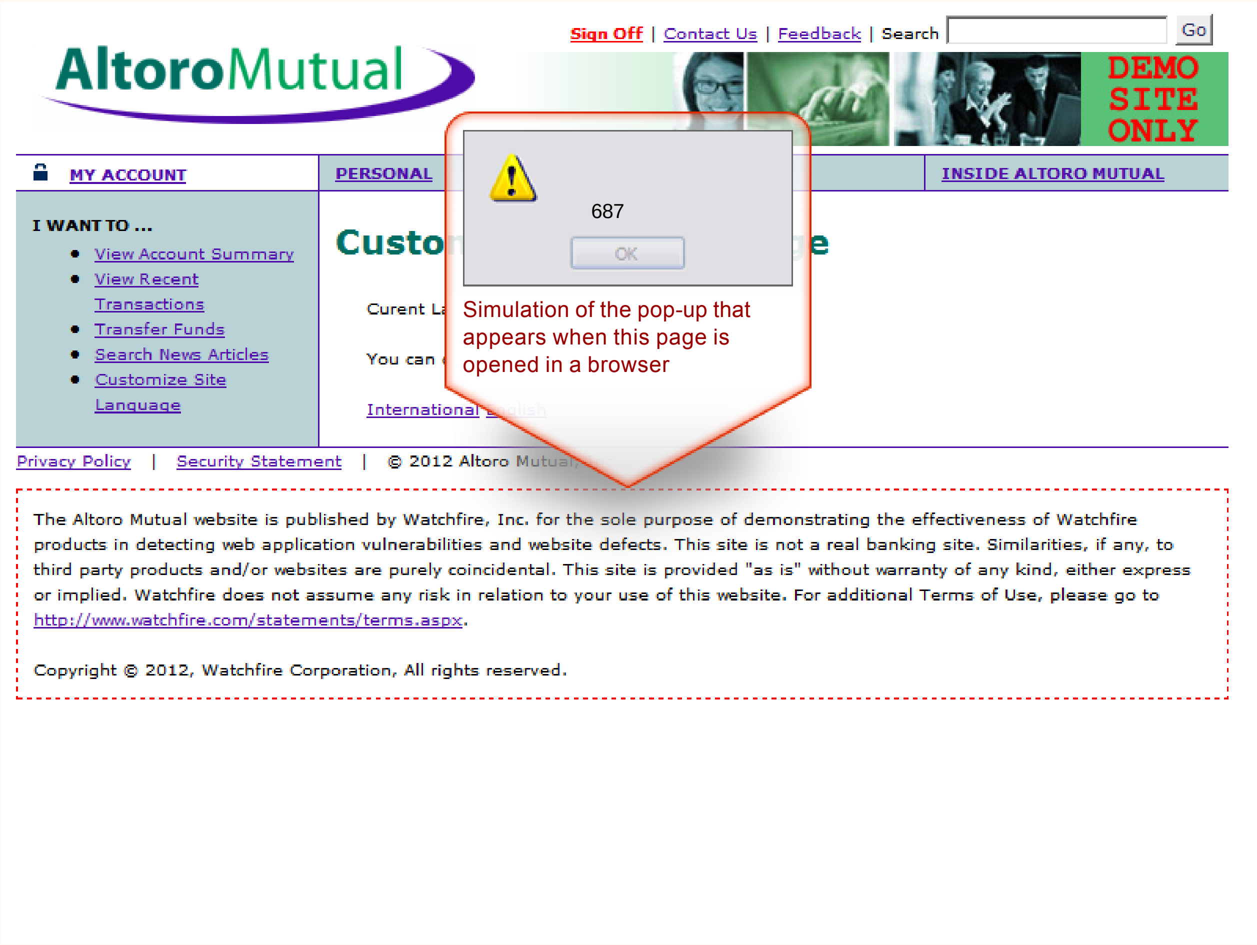
```
...


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: 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

...

...

  </tr>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
in string in query expression 'accountid=1001160141'"><iframe src=javascript:alert(2435)>'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 11 of 11

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

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

Raw Test Response:

```
...

<div class="fl" style="width: 99%;">
```

```
<h1>Customize Site Language</h1>

<form name="aspnetForm" method="post" action="customize.aspx?lang=international%3cscript%3ealert(124)%3c%2fscript%3e" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUJMjA2OTMxMDA4ZGQ=" />

  <p>
  <span id="_ctl0__ctl0_Content_Main_Label1">Curent Language: </span>
  <span id="_ctl0__ctl0_Content_Main_langLabel">international<script>alert(124)</script></span>
  </p>

  <p>
  <span id="_ctl0__ctl0_Content_Main_Label2">You can change the language setting by choosing:</span>
  </p>

  <p>
  <a id="_ctl0__ctl0_Content_Main_HyperLink1" href="customize.aspx?lang=international">International</a>
  <a id="_ctl0__ctl0_Content_Main_HyperLink2" href="customize.aspx?lang=english">English</a>

...
```

H DOM Based Cross-Site Scripting 3

TOC

Issue 1 of 3

TOC

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.

<script> 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

TOC

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.

```
   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

TOC

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.

```
   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;       }    }
```

H Poison Null Byte Windows Files Retrieval 1

TOC

Issue 1 of 1

TOC

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 reponse 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 /noguiboot
</span>

    </td>
  </tr>
</table>


</div>

...
```

H Predictable Login Credentials 1

TOC

Issue 1 of 1

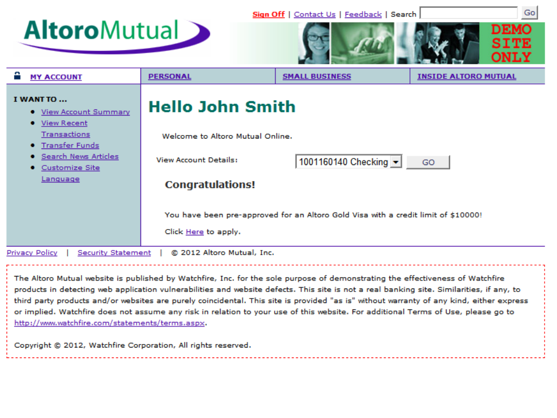
TOC

Predictable Login Credentials

Severity:	High
URL:	http://demo.testfire.net/bank/login.aspx
Entity:	login.aspx (Page)
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:	Change the login credentials to a stronger combination

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



Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

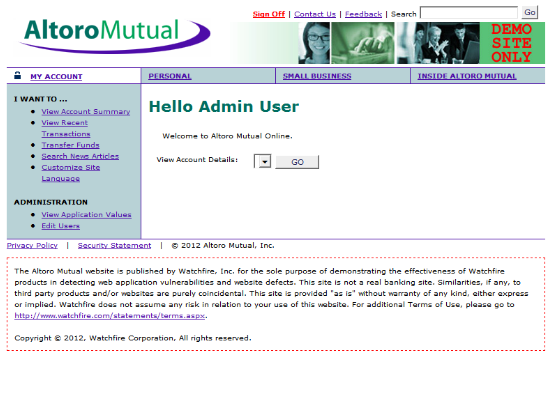
© 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

≈

Test Login



Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details:

© 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

H SQL Injection 12

TOC

Issue 1 of 12

TOC

SQL Injection

Severity:	**High**
URL:	http://demo.testfire.net/subscribe.aspx
Entity:	txtEmail (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:

```
...

<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 in query expression ''test@altoromutual.com';')'.
</span></b></p>

<h2>Error Message:</h2>
```

```
<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in query expression ''test@altoromutual.com';')'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   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
   at System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e)
   at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e)

...
```

Issue 2 of 12

TOC

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:	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 in string in query expression '1=1  and t.trans_date &gt;= 1234 and t.trans_date
&lt;= 1234'; and a.userid = 100116014 ORDER BY 1 DESC'.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression '1=1  and
t.trans_date &gt;= 1234 and t.trans_date &lt;= 1234'; and a.userid = 100116014 ORDER BY 1 DESC'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 3 of 12

TOC

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:

```
...

<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 in string in query expression '1=1  and t.trans_date &gt;= 1234'; and t.trans_date
&lt;= 1234 and a.userid = 100116014 ORDER BY 1 DESC'.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression '1=1  and
t.trans_date &gt;= 1234'; and t.trans_date &lt;= 1234 and a.userid = 100116014 ORDER BY 1 DESC'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 4 of 12

TOC

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="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 in string in query expression 'userid = 100116014''.
</span></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.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 5 of 12

TOC

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.

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 in string in query expression 'userid = 100116014''.
</span></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.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 6 of 12

TOC

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:

```
...

<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 in string in query expression '1=1  and t.trans_date &gt;= 1234 and t.trans_date
&lt;= 1234 and a.userid = 100116014' ORDER BY 1 DESC'.
</span></b></p>

<h2>Error Message:</h2>
```

```
<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression '1=1  and
t.trans_date &gt;= 1234 and t.trans_date &lt;= 1234 and a.userid = 100116014' ORDER BY 1 DESC'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 7 of 12

TOC

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:

```
...

<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">Characters found after end of SQL statement.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Characters found after end of SQL statement.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 8 of 12

TOC

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:

```
...

<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">Characters found after end of SQL statement.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Characters found after end of SQL statement.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 9 of 12

TOC

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:

```
...

  </tr>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
in string in query expression 'accountid=1001160141';'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 10 of 12

TOC

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:

```
...
```

```
  </tr>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
in string in query expression 'accountid=1001160141';'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 11 of 12

TOC

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

<?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"><soap:Body><TransferBalanceResponse
xmlns="http://www.altoromutual.com/bank/ws/"><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException:
Syntax error in query expression 'accountid=1001160141%27%3B'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
...
```

Issue 12 of 12

TOC

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

<?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"><soap:Body><TransferBalanceResponse
xmlns="http://www.altoromutual.com/bank/ws/"><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException:
Syntax error (missing operator) in query expression 'accountid=10011601411+'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
...
```

H Unencrypted Login Request 6

TOC

Issue 1 of 6

TOC

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

TOC

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

TOC

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

```
passwd=Demo1234&Submit=Submit
```

Issue 4 of 6

TOC

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

Issue 5 of 6

TOC

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

TOC

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=%2FwEPDwUKMTY5ODYzNjk3NWRk&__EVENTVALIDATION=%2FwEWBAKm%2FPqICgKaqvKtBQKWuPeSCgL73pWUBA%3D%3D&_ctl0%3A_ctl0%3AContent%3AMain%
3ACodeNumberTextBox=9876543210&_ctl0%3A_ctl0%3AContent%3AMain%3APassword=Demo1234&_ctl0%3A_ctl0%3AContent%3AMain%3ASubmitButton=Submit
```

H XPath Injection 1

TOC

Issue 1 of 1

TOC

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:

```
...


<h1>An Error Has Occurred</h1>

<h2>Summary:</h2>

<p><b><span id="_ctl0_Content_lblSummary">'string(/news/publication[contains(title,'&quot;'') and (isPublic/text()='True')]/title/text
())' has an invalid token.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Xml.XPath.XPathException: 'string(/news/publication[contains(title,'&quot;'') 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)
```

```
    at MS.Internal.Xml.XPath.XPathParser.ParsePathExpr(AstNode qyInput)
    at MS.Internal.Xml.XPath.XPathParser.ParseUnionExpr(AstNode qyInput)
    at MS.Internal.Xml.XPath.XPathParser.ParseUnaryExpr(AstNode qyInput)
    at MS.Internal.Xml.XPath.XPathParser.ParseMultiplicativeExpr(AstNode qyInput)
    at MS.Internal.Xml.XPath.XPathParser.ParseAdditiveExpr(AstNode qyInput)

 ...
```

M Cross-Site Request Forgery 6

TOC

Issue 1 of 6

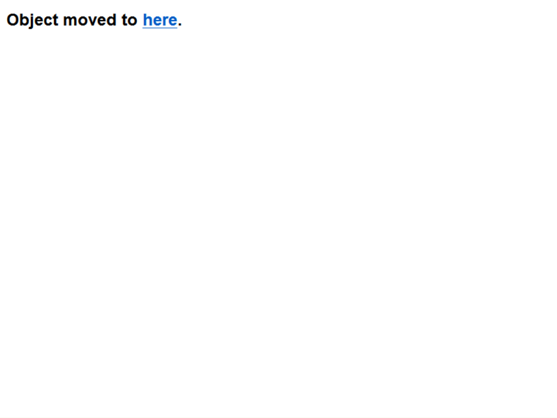
TOC

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.

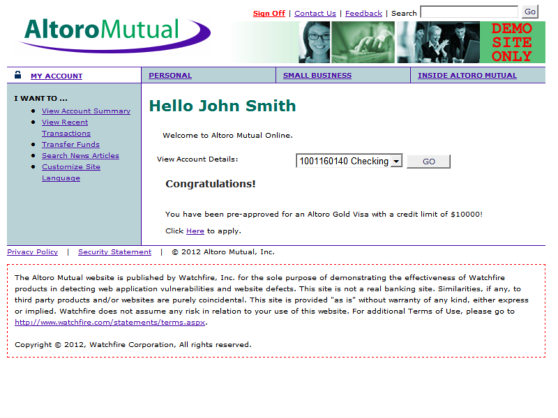
Original Response



Object moved to here.

≈

Test Response



I WANT TO ...
- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking GO

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

Issue 2 of 6

TOC

Cross-Site Request Forgery

Severity:	Medium
URL:	http://demo.testfire.net/bank/transfer.aspx
Entity:	transfer.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/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; .NET CLR 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; .NET CLR 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
...
```

Issue 3 of 6

TOC

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://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: 176

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%
2FwEPDwUKMTYzNDg3OTA4NmRk&__EVENTVALIDATION=%
2FwEWBgKV3oKhDgK3oeuaBAK3oaesDgK3oZPRBQK3oa%2BJCgK3oZuuAQ%3D%
3D&after=1234&before=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://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: 176

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%
2FwEPDwUKMTYzNDg3OTA4NmRk&__EVENTVALIDATION=%
2FwEWBgKV3oKhDgK3oeuaBAK3oaesDgK3oZPRBQK3oa%2BJCgK3oZuuAQ%3D%
3D&after=1234&before=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
...
```

Issue 4 of 6

TOC

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
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; 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 1.1.4322; .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
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; 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 1.1.4322; .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
...
```

Issue 5 of 6

TOC

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
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; .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
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; .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
...
```

Issue 6 of 6

TOC

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
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=eu0qbsjngqgirw45q0opxa45;
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.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: 17

accttypes=Savings
```

Test Response

```
POST /admin/admin.aspx HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=eu0qbsjngqgirw45q0opxa45;
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.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: 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
...
```

M Directory Listing 2

TOC

Issue 1 of 2

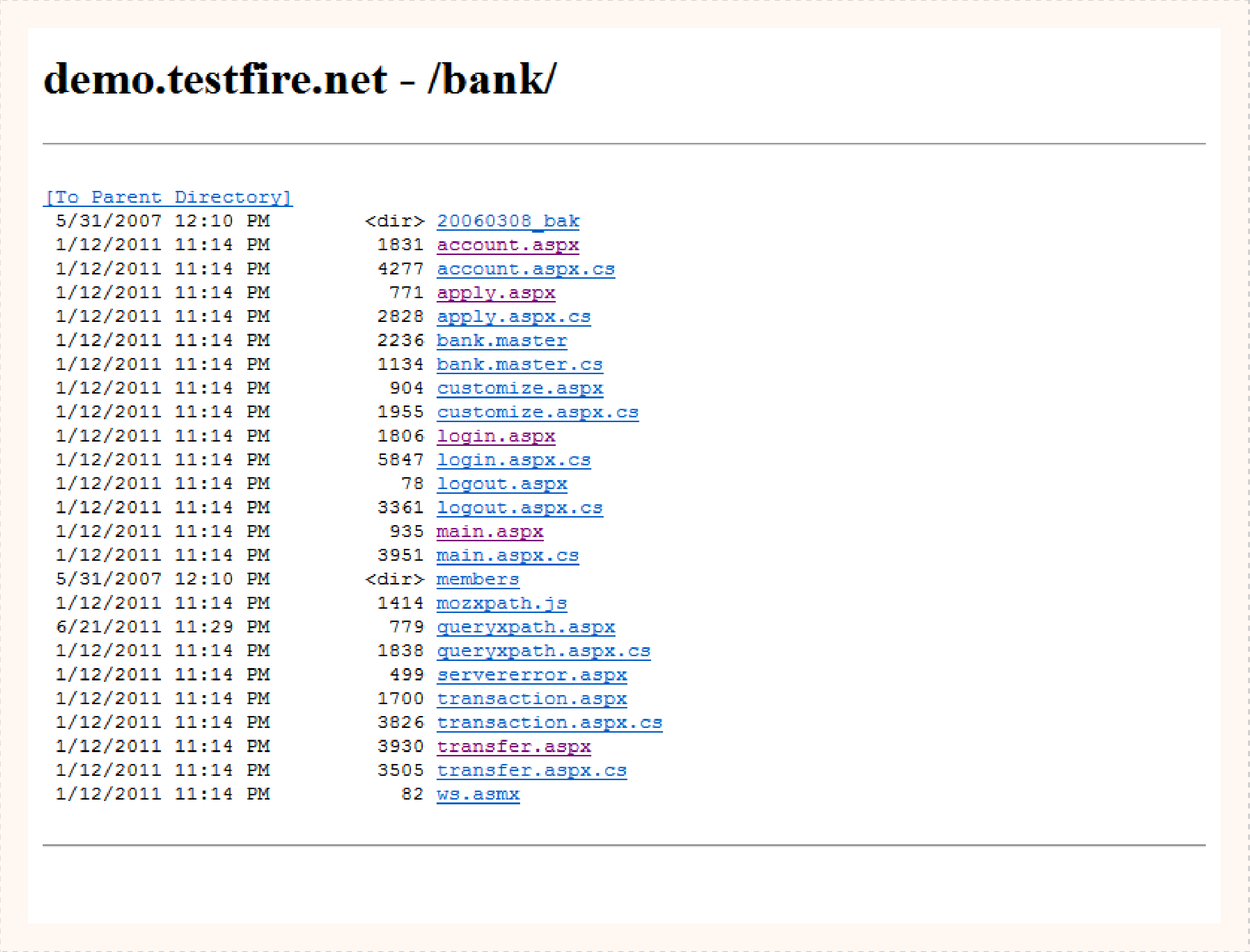
TOC

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> 20060308_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         2828 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         1955 customize.aspx.cs
 1/12/2011 11:14 PM         1806 login.aspx
 1/12/2011 11:14 PM         5847 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          935 main.aspx
 1/12/2011 11:14 PM         3951 main.aspx.cs
 5/31/2007 12:10 PM        <dir> members
 1/12/2011 11:14 PM         1414 mozxpath.js
 6/21/2011 11:29 PM          779 queryxpath.aspx
 1/12/2011 11:14 PM         1838 queryxpath.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         3826 transaction.aspx.cs
 1/12/2011 11:14 PM         3930 transfer.aspx
 1/12/2011 11:14 PM         3505 transfer.aspx.cs
 1/12/2011 11:14 PM           82 ws.asmx
```

Issue 2 of 2

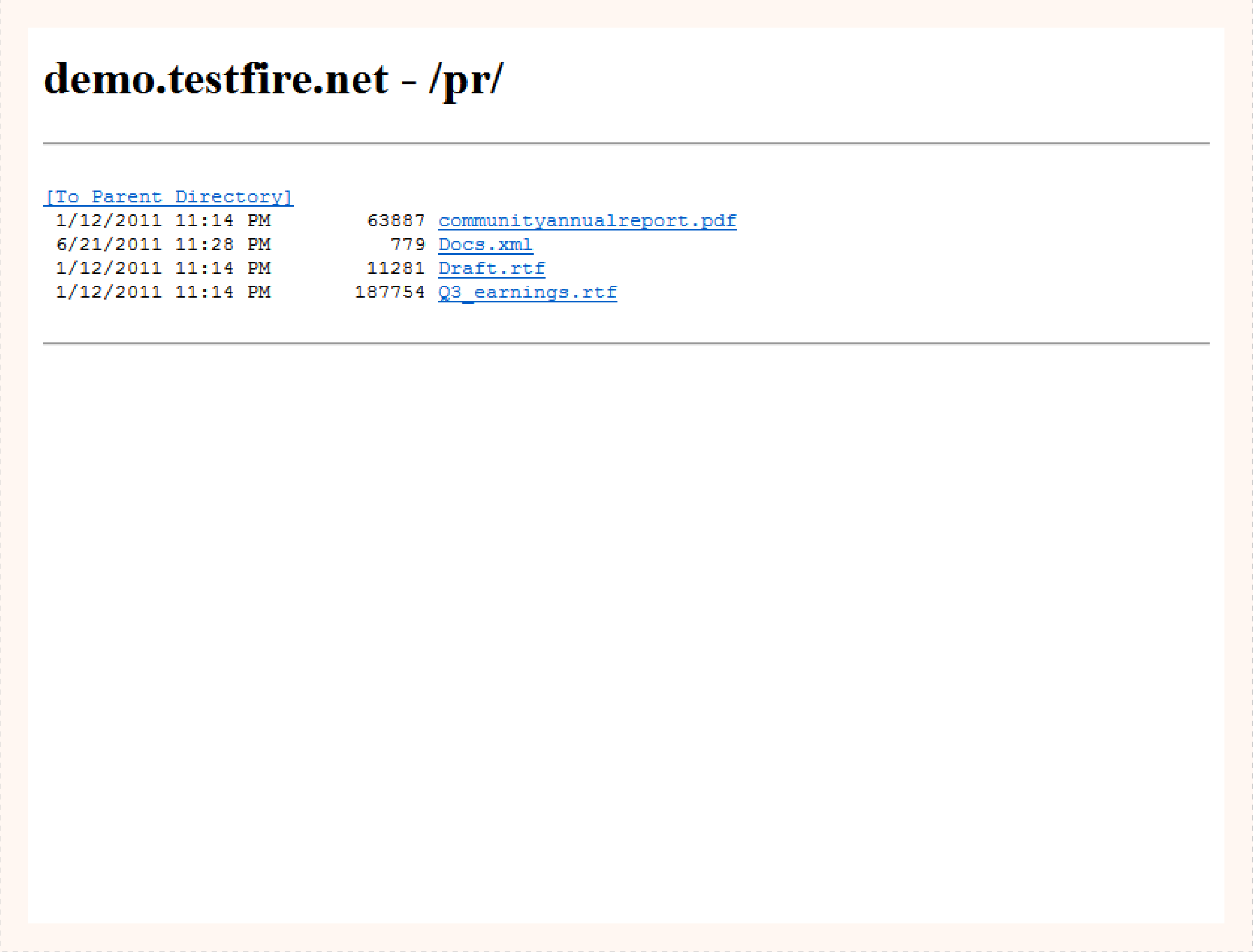
TOC

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/

```
[To Parent Directory]
 1/12/2011 11:14 PM        63887 communityannualreport.pdf
 6/21/2011 11:28 PM          779 Docs.xml
 1/12/2011 11:14 PM        11281 Draft.rtf
 1/12/2011 11:14 PM       187754 Q3_earnings.rtf
```

M HTTP Response Splitting 1 TOC

Issue 1 of 1 TOC

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:

```
...

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 08:36:38 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">

...
```

M Inadequate Account Lockout 1

TOC

Issue 1 of 1

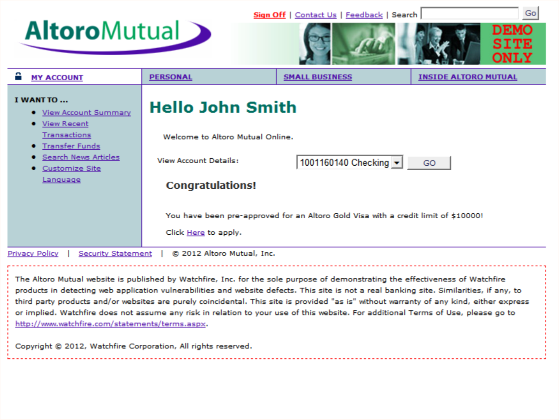
TOC

Inadequate Account Lockout

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

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)



Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

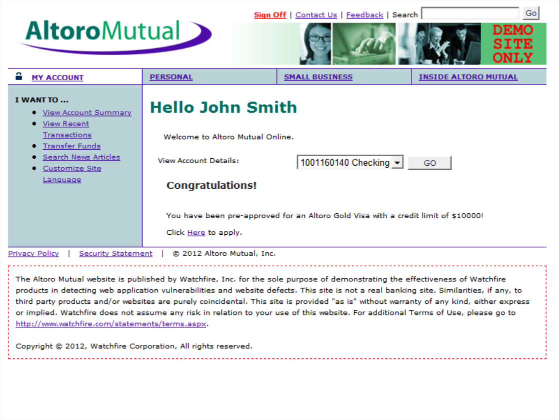
Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

≈

Test Response (last)


Hello John Smith

Welcome to Altoro Mutual Online.

View Account Details: 1001160140 Checking

Congratulations!

You have been pre-approved for an Altoro Gold Visa with a credit limit of $10000!

Click Here to apply.

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

M Link Injection (facilitates Cross-Site Request Forgery) 6

TOC

Issue 1 of 6

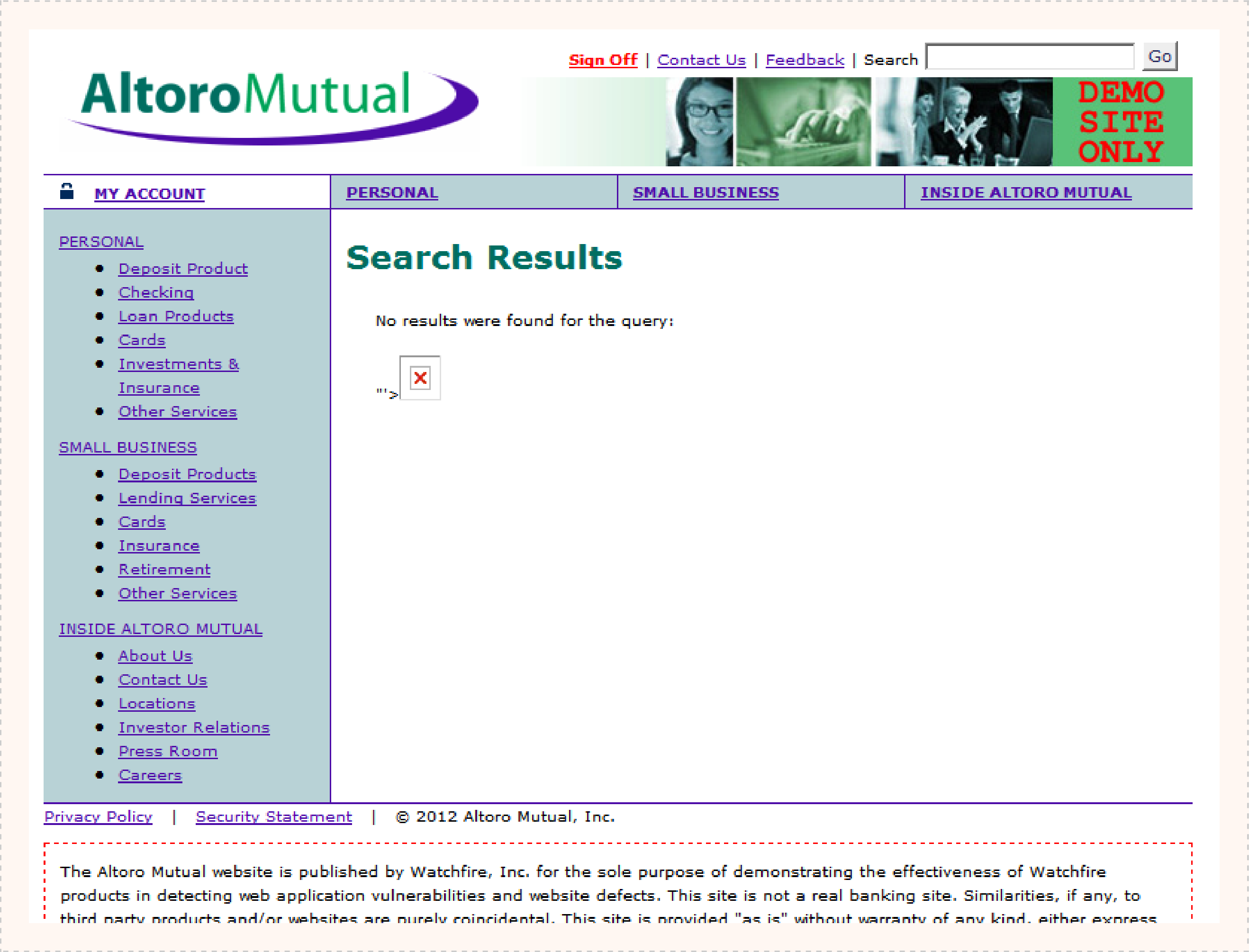
TOC

Link Injection (facilitates Cross-Site Request Forgery)

Severity:	Medium
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 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



Search Results

No results were found for the query:

"'>

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 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

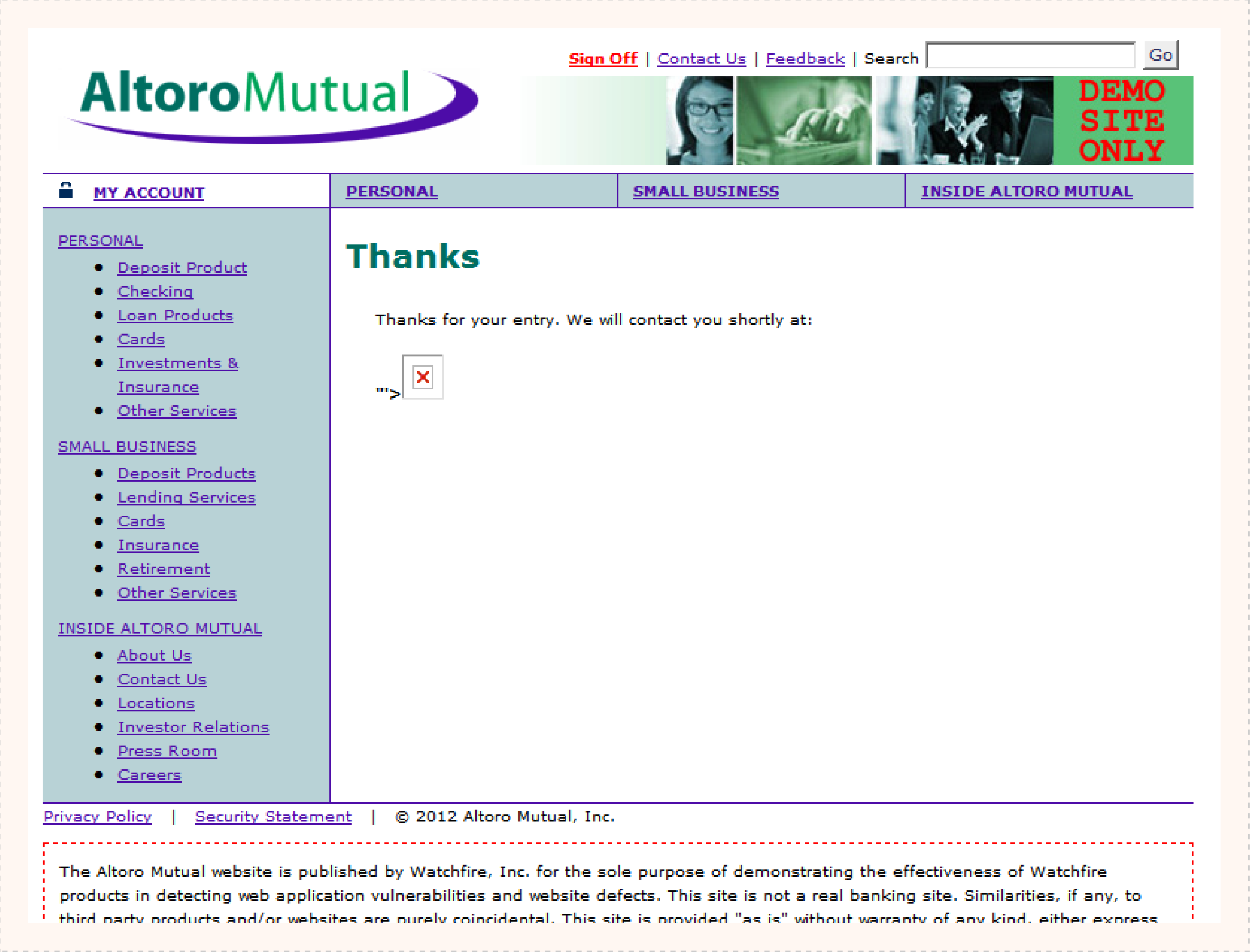
Issue 2 of 6

TOC

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



Thanks

Thanks for your entry. We will contact you shortly at:

">

© 2012 Altoro Mutual, Inc.

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 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

Issue 3 of 6

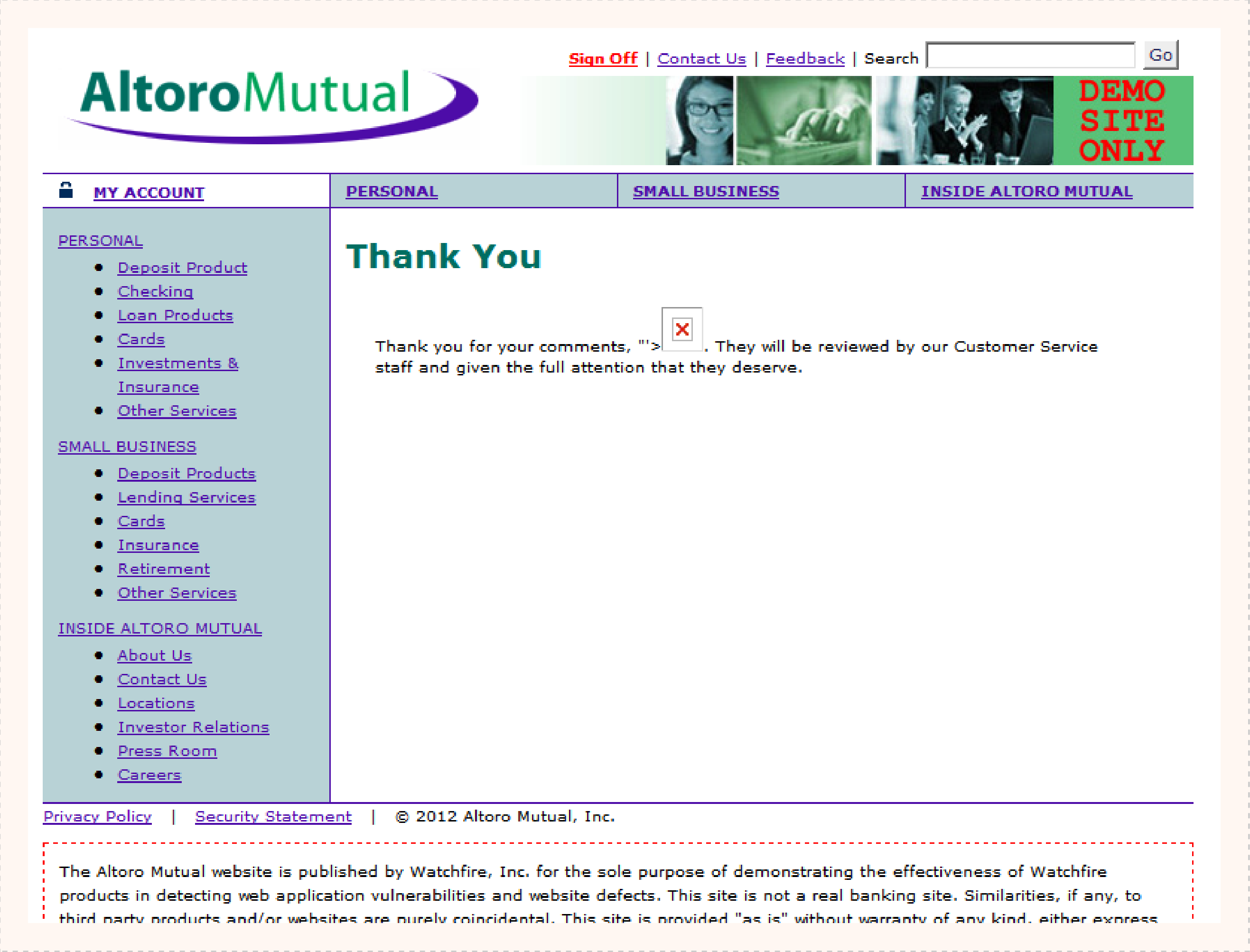
TOC

Link Injection (facilitates Cross-Site Request Forgery)

Severity:	Medium
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 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



Thank You

Thank you for your comments, "'> . They will be reviewed by our Customer Service staff and given the full attention that they deserve.

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 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

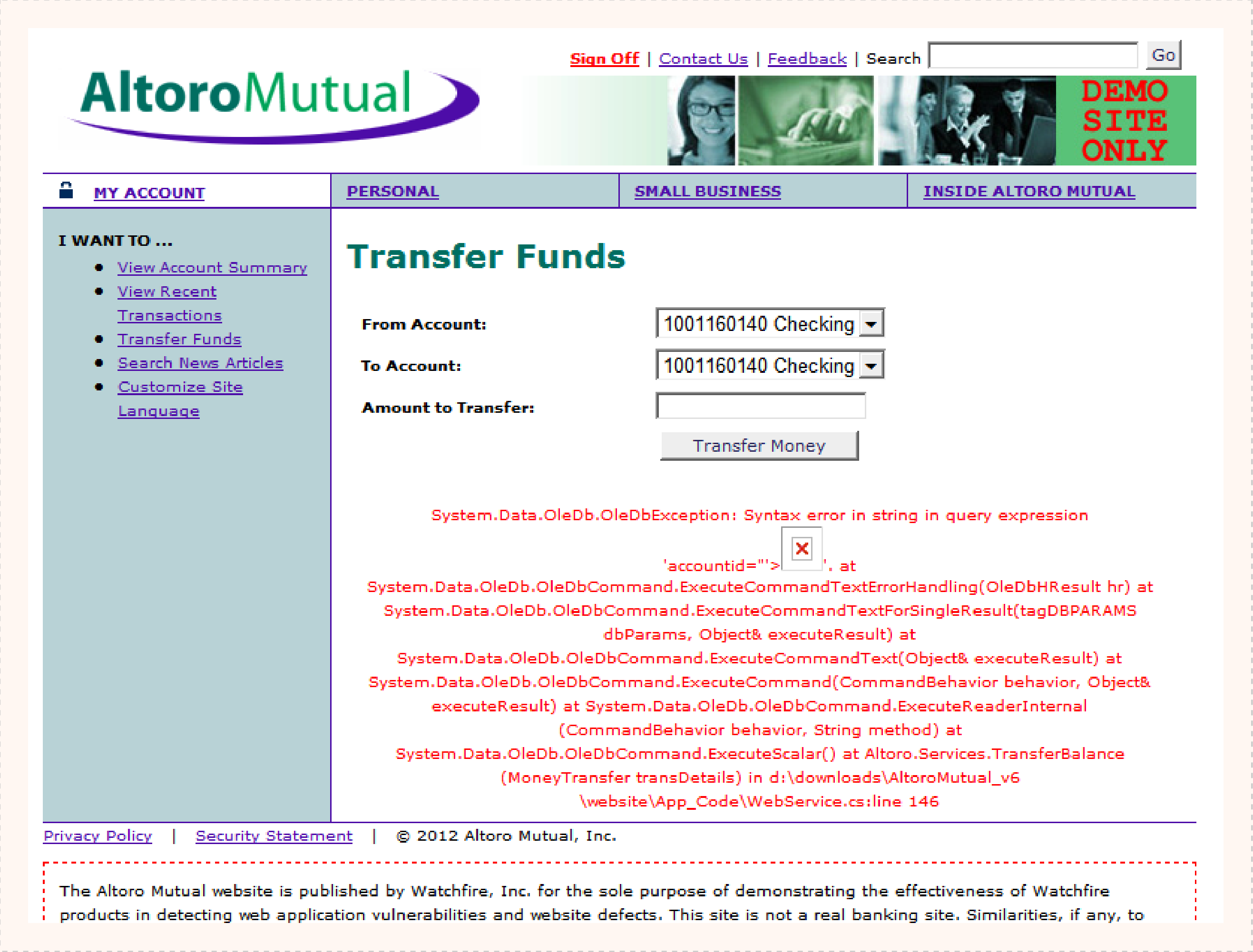
Issue 4 of 6

TOC

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



Sign Off | Contact Us | Feedback | Search [ ] Go

AltoroMutual

DEMO SITE ONLY

MY ACCOUNT | PERSONAL | SMALL BUSINESS | INSIDE ALTORO MUTUAL

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

Transfer Funds

From Account:	1001160140 Checking
To Account:	1001160140 Checking
Amount to Transfer:	
	Transfer Money

```
System.Data.OleDb.OleDbException: Syntax error in string in query expression
'accountid='">', at
System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr) at
System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS
dbParams, Object& executeResult) at
System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult) at
System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&
executeResult) at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal
(CommandBehavior behavior, String method) at
System.Data.OleDb.OleDbCommand.ExecuteScalar() at Altoro.Services.TransferBalance
(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6
\website\App_Code\WebService.cs:line 146
```

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 and website defects. This site is not a real banking site. Similarities, if any, to

Issue 5 of 6

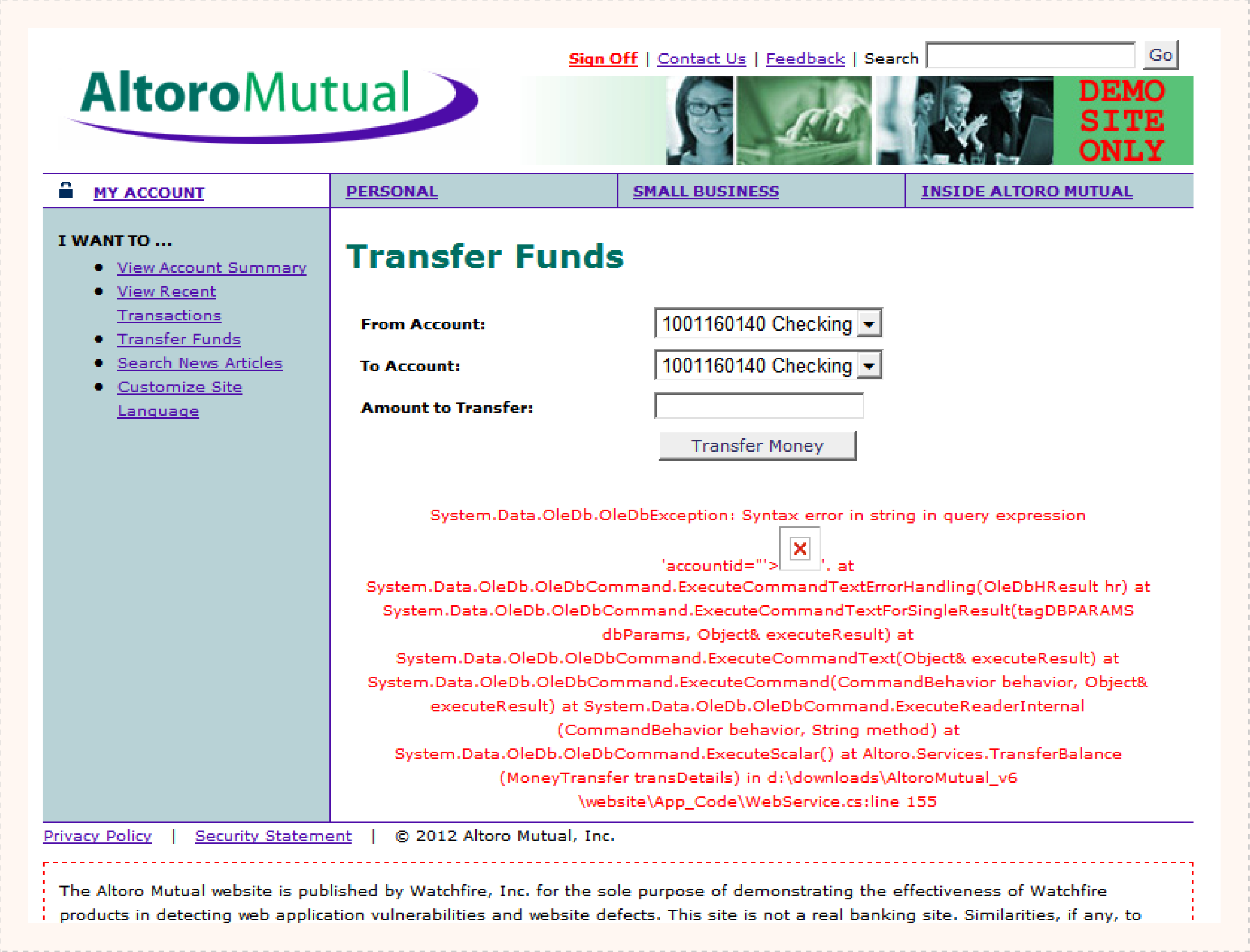
TOC

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



Transfer Funds

From Account: 1001160140 Checking

To Account: 1001160140 Checking

Amount to Transfer:

System.Data.OleDb.OleDbException: Syntax error in string in query expression 'accountid="'>☒'. at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr) at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult) at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult) at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult) at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal (CommandBehavior behavior, String method) at System.Data.OleDb.OleDbCommand.ExecuteScalar() at Altoro.Services.TransferBalance (MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6 \website\App_Code\WebService.cs:line 155

© 2012 Altoro Mutual, Inc.

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 and website defects. This site is not a real banking site. Similarities, if any, to

Issue 6 of 6

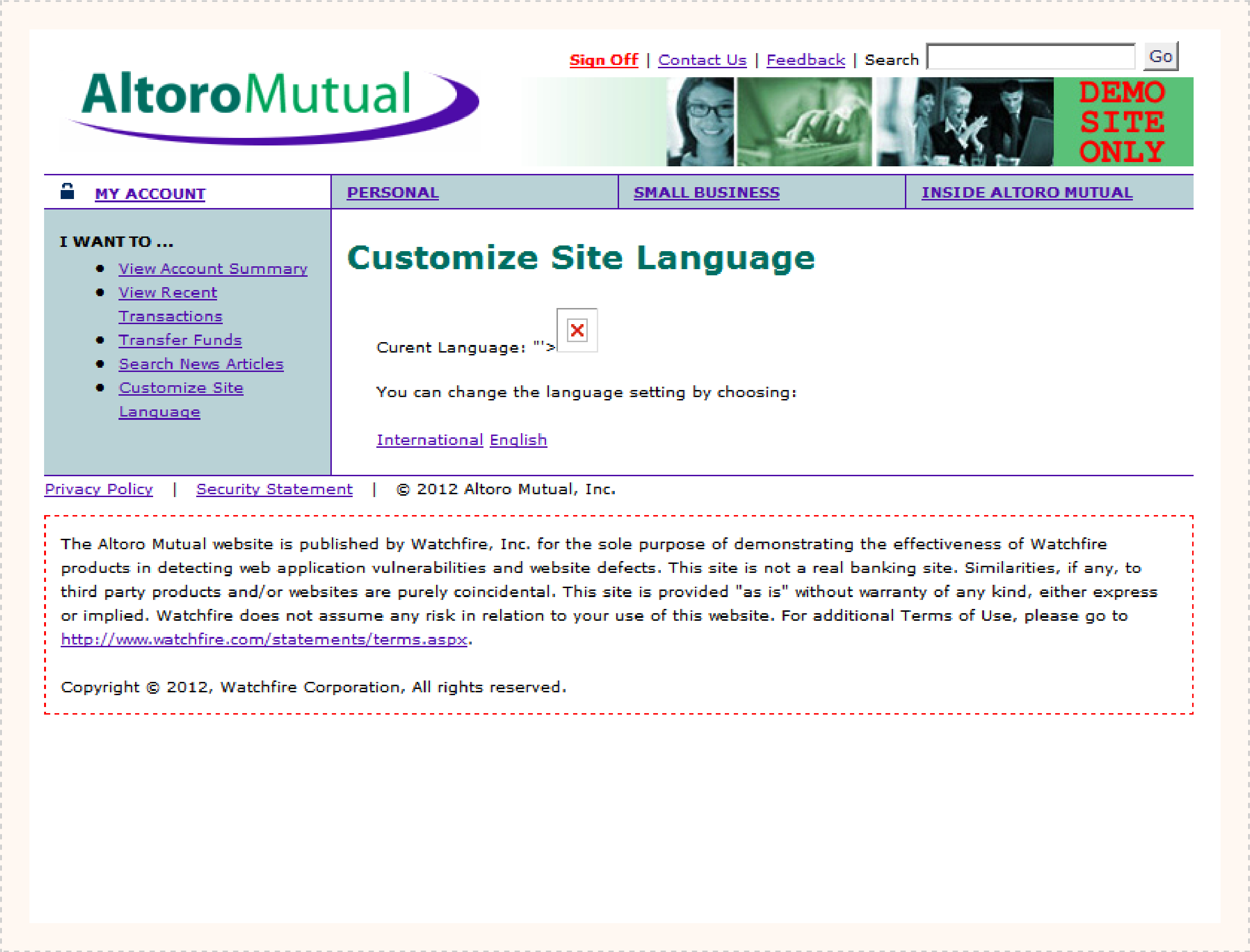
TOC

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



Customize Site Language

Curent Language: "'>

You can change the language setting by choosing:

International English

Privacy Policy | Security Statement | © 2012 Altoro Mutual, Inc.

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 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 or implied. Watchfire does not assume any risk in relation to your use of this website. For additional Terms of Use, please go to http://www.watchfire.com/statements/terms.aspx.

Copyright © 2012, Watchfire Corporation, All rights reserved.

M Open Redirect 2

TOC

Issue 1 of 2

TOC

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.

```
  }   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
("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);
```

Issue 2 of 2

TOC

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.

```
   }   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
("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);      }   }
```

M Phishing Through Frames 6

TOC

Issue 1 of 6

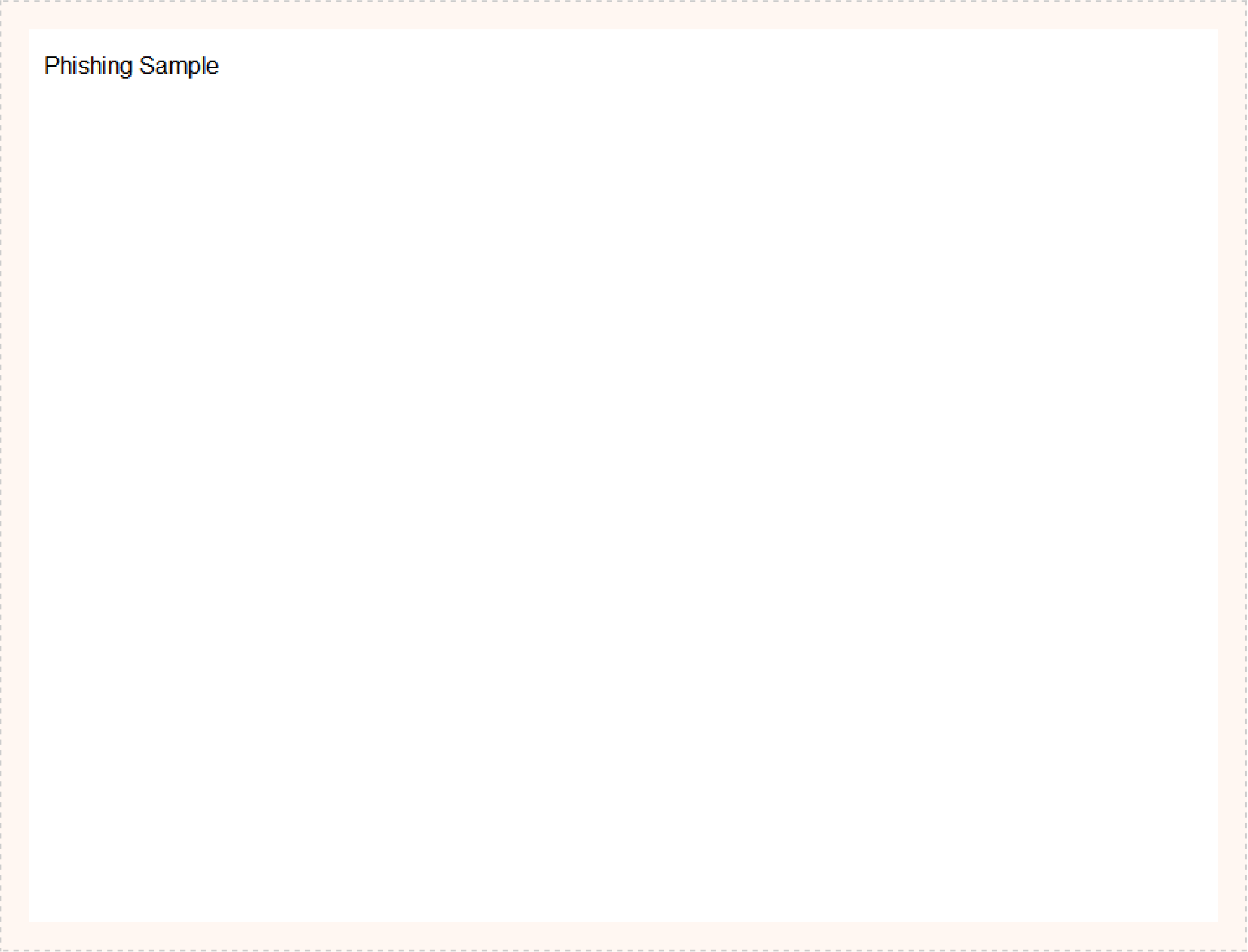
TOC

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



Phishing Sample

Issue 2 of 6

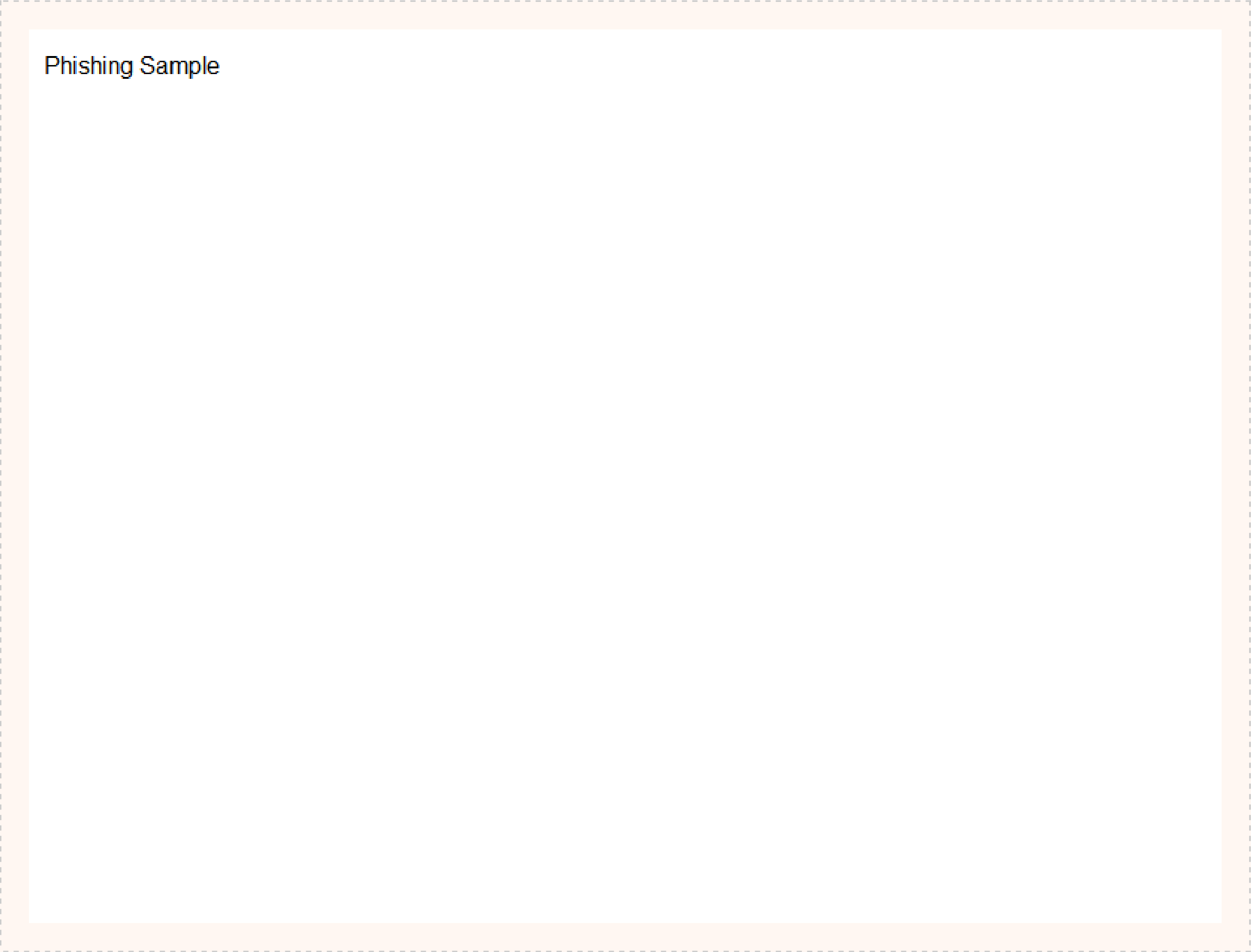
TOC

Phishing Through Frames

Severity:	Medium
URL:	http://demo.testfire.net/survey_complete.aspx
Entity:	txtEmail (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



Phishing Sample

Issue 3 of 6

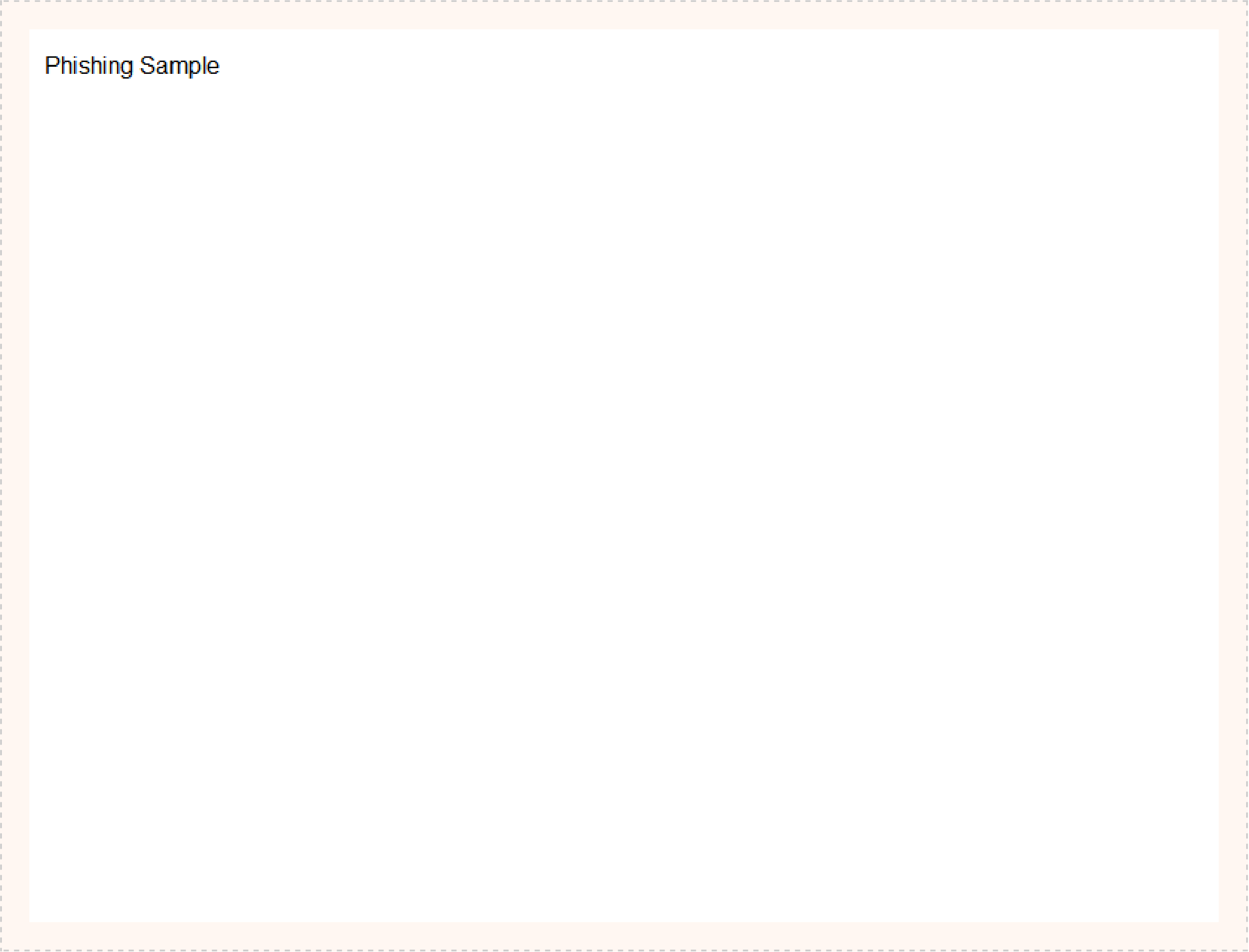
TOC

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



Phishing Sample

Issue 4 of 6

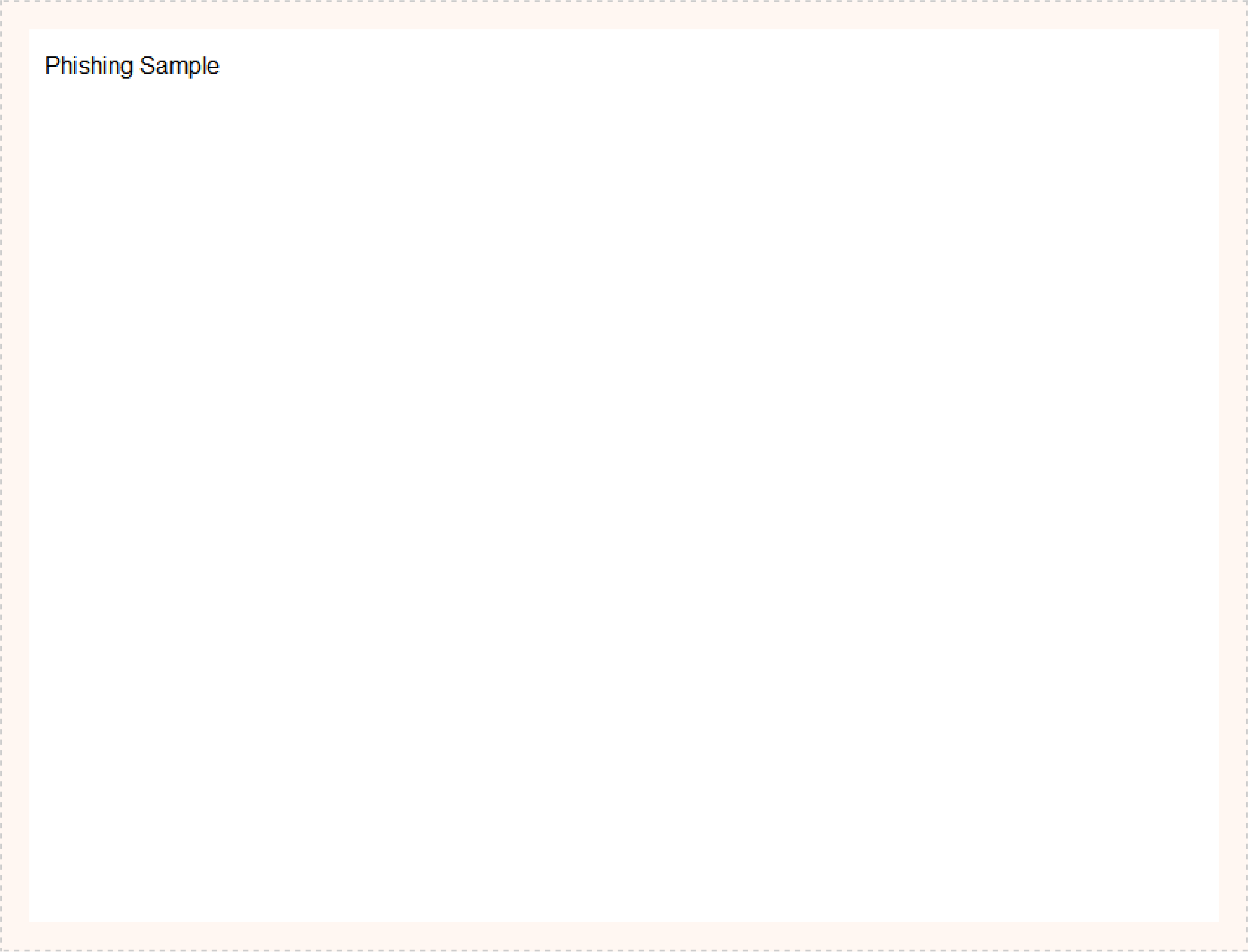
TOC

Phishing Through Frames

Severity:	Medium
URL:	http://demo.testfire.net/bank/transfer.aspx
Entity:	debitAccount (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



Phishing Sample

Issue 5 of 6

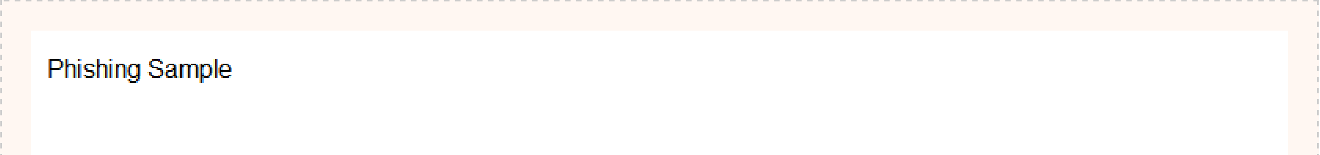
TOC

Phishing Through Frames

Severity:	Medium
URL:	http://demo.testfire.net/bank/transfer.aspx
Entity:	creditAccount (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



Phishing Sample

Issue 6 of 6

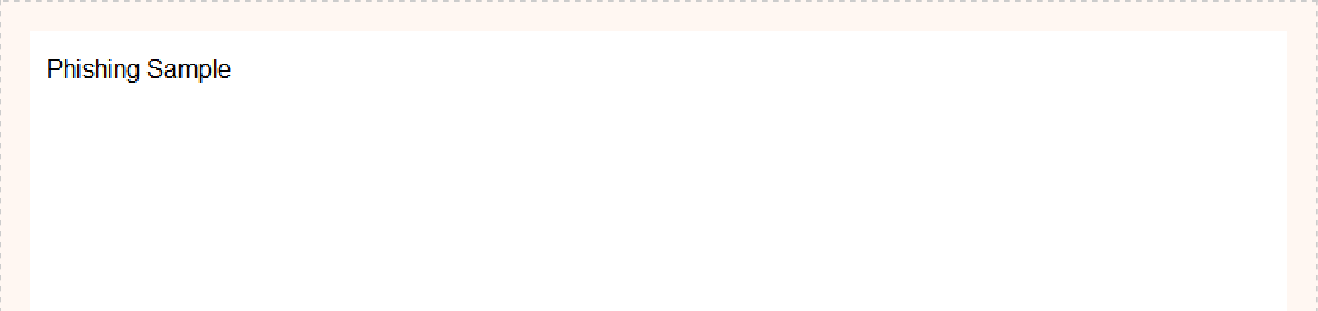
TOC

Phishing Through Frames

Severity:	Medium
URL:	http://demo.testfire.net/bank/customize.aspx
Entity:	lang (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



Phishing Sample

M Session Identifier Not Updated 1 TOC

Issue 1 of 1 TOC

Session Identifier Not Updated	
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:	Insecure web application programming or configuration
Fix:	Do not accept externally created session identifiers

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.

L Autocomplete HTML Attribute Not Disabled for Password Field 4 TOC

Issue 1 of 4

TOC

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:

```
...

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

...
```

Issue 2 of 4

TOC

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:

```
...

  Visa Application</h1>
```

```
<!--
    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 an 7.9% APR simply enter your password below.</p><form method="post" name="Credit" action="apply.aspx"><table
border=0><tr><td>Password:</td><td><input type="password" name="passwd"></td></tr><tr><td></td><td><input type="submit" name="Submit"
value="Submit"></td></tr></table></form></span>

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

</div>



...
```

Issue 3 of 4

TOC

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:

```
...

<form name="aspnetForm" method="post" action="login.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTY5ODYzNjk3NWRk" />

<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWBAKm/PqICgKaqvKtBQKWuPeSCgL73pWUBA==" />
  <img id="captcha" src="captcha.aspx" /><br />
  <p>
    <strong>Enter the code shown above:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:CodeNumberTextBox" type="text" id="_ctl0__ctl0_Content_Main_CodeNumberTextBox" /><br /><br />
    <strong>Enter the administrative password:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:Password" type="password" id="_ctl0__ctl0_Content_Main_Password" /><br /><br />
    <input type="submit" name="_ctl0:_ctl0:Content:Main:SubmitButton" value="Submit" id="_ctl0__ctl0_Content_Main_SubmitButton" /><br />
  </p>
  <p><span id="_ctl0__ctl0_Content_Main_MessageLabel"></span></p>
</form>

<script>
window.onload = document.forms[1].elements[1].focus();
</script>


...
```

Issue 4 of 4

TOC

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:

```
...

      Confirm:
    </th>
    <th>&nbsp;</th>
  </tr>
  <tr>
    <td>
      <select id="" name="" ><option value="1">1 admin</option><option value="2">2 tuser</option><option value="100116013">100116013
sjoe</option><option value="100116014">100116014 jsmith</option><option value="100116015">100116015 cclay</option><option
value="100116018">100116018 sspeed</option></select>
    </td>
    <td>
      <input type="password" name="password1">
    </td>
    <td>
      <input type="password" name="password2">
    </td>
    <td>
      <input type="submit" name="change" value="Change Password">
    </td>
  </tr>
</form>
<form method="post" name="addUser" action="admin.aspx" id="addUser" onsubmit="return confirmpass(this);">
  <tr>
    <td colspan="4"><h2>Add an new user.</h2></td>

...

...

    <td>
      <input type="text" name="firstname">
      <br>
      <input type="text" name="lastname">
    </td>
    <td>
      <input type="text" name="username">
    </td>
    <td>
      <input type="password" name="password1">
      <br>
      <input type="password" name="password2">
    </td>
    <td>
      <input type="submit" name="add" value="Add User">
    </td>
  </tr>
  <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>

...
```

L Database Error Pattern Found 16

TOC

Issue 1 of 16

TOC

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 ''&gt;&quot;'&gt;&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
''&gt;&quot;'&gt;&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; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   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
   at System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e)
   at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e)

...
```

Issue 2 of 16

TOC

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:

```
...

<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 in string in query expression ''test@altoromutual.comWFXSSProbe'&quot;)/&gt;')'.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression
''test@altoromutual.comWFXSSProbe'&quot;)/&gt;')'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   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
   at System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e)
   at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e)

...
```

Issue 3 of 16

TOC

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:

```
...

<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 '1=1  and t.trans_date &gt;= 1234 and
t.trans_date &lt;= 1234WFXSSProbe and a.userid = 100116014'.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression '1=1  and
t.trans_date &gt;= 1234 and t.trans_date &lt;= 1234WFXSSProbe and a.userid = 100116014'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 4 of 16

TOC

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:

```
...

<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 '1=1  and t.trans_date &gt;= 1234WFXSSProbe
and t.trans_date &lt;= 1234 and a.userid = 100116014'.
</span></b></p>

<h2>Error Message:</h2>
```

```
<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression '1=1  and
t.trans_date &gt;= 1234WFXSSProbe and t.trans_date &lt;= 1234 and a.userid = 100116014'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 5 of 16

TOC

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="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 'username =
'&gt;&quot;'&gt;&lt;script&gt;alert(1549)&lt;/script&gt;' AND password = '&gt;&quot;'&gt;&lt;script&gt;alert(1549)&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 'username =
'&gt;&quot;'&gt;&lt;script&gt;alert(1549)&lt;/script&gt;' AND password = '&gt;&quot;'&gt;&lt;script&gt;alert(1549)&lt;/script&gt;''.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 6 of 16

TOC

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:

```
...

<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 = 100116014WFXSSProbe'.
</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 =
100116014WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 7 of 16

TOC

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="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 = 100116014WFXSSProbe'.
</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 =
100116014WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 8 of 16

TOC

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.

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 'userid = 100116014WFXSSProbe'.
</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 =
100116014WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 9 of 16

TOC

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:

```
...

<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 '1=1  and t.trans_date &gt;= 1234 and
t.trans_date &lt;= 1234 and a.userid = 100116014WFXSSProbe'.
</span></b></p>

<h2>Error Message:</h2>
```

```
<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression '1=1  and
t.trans_date &gt;= 1234 and t.trans_date &lt;= 1234 and a.userid = 100116014WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 10 of 16

TOC

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:

```
...

<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 '1=1  and t.trans_date &gt;=
&gt;&quot;'&gt;&lt;script&gt;alert(1478)&lt;/script&gt; and t.trans_date &lt;= &gt;&quot;'&gt;&lt;script&gt;alert(1478)&lt;/script&gt;
and a.userid = 100116014 ORDER BY 1 DESC'.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression '1=1  and
t.trans_date &gt;= &gt;&quot;'&gt;&lt;script&gt;alert(1478)&lt;/script&gt; and t.trans_date &lt;= &gt;&quot;'&gt;&lt;script&gt;alert
(1478)&lt;/script&gt; and a.userid = 100116014 ORDER BY 1 DESC'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 11 of 16

TOC

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:

```
...

<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 in string in query expression 'username = 'jsmith' AND password =
'demo1234WFXSSProbe'&quot;)/&gt;''.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression 'username = 'jsmith'
AND password = 'demo1234WFXSSProbe'&quot;)/&gt;''.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 12 of 16

TOC

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:

```
...

<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 in string in query expression 'username = 'jsmithWFXSSProbe'&quot;)/&gt;' AND
password = 'demo1234''.
</span></b></p>

<h2>Error Message:</h2>

<p><span id="_ctl0_Content_lblDetails">System.Data.OleDb.OleDbException: Syntax error in string in query expression 'username =
'jsmithWFXSSProbe'&quot;)/&gt;' AND password = 'demo1234''.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteReader(CommandBehavior behavior)
   at System.Data.OleDb.OleDbCommand.System.Data.IDbCommand.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)

...
```

Issue 13 of 16

TOC

Database Error Pattern Found

Severity:	Low
URL:	http://demo.testfire.net/bank/transfer.aspx
Entity:	creditAccount (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:

```
...

  </tr>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
in string in query expression 'accountid=1001160141'"><iframe src=javascript:alert(2435)>'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 14 of 16

TOC

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:

```
...

  </tr>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
(missing operator) in query expression 'accountid=1001160141WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 15 of 16

TOC

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/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><TransferBalanceResponse
xmlns="http://www.altoromutual.com/bank/ws/"><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException:
Syntax error (missing operator) in query expression 'accountid=1001160141WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
...
```

Issue 16 of 16

TOC

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
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/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><TransferBalanceResponse
xmlns="http://www.altoromutual.com/bank/ws/"><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException:
Syntax error (missing operator) in query expression 'accountid=1001160141WFXSSProbe'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
...
```

L Direct Access to Administration Pages 2

TOC

Issue 1 of 2

TOC

Direct Access to Administration Pages	
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/msword, */*
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

```
...

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


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

...
```

Issue 2 of 2

TOC

Direct Access to Administration Pages

Severity:	Low
URL:	http://demo.testfire.net/admin/clients.xls
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:

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 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
```

```
...
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 1.1.4322; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)


HTTP/1.1 200 OK
Date: Sun, 22 Jul 2012 08:17:49 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

...
```

L Email Address Pattern Found in Parameter Value 2

TOC

Issue 1 of 2

TOC

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/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/msword, */*
Referer: http://demo.testfire.net/survey_complete.aspx
```

```
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=rccg0sjfeksi0g45p2smc0ui; amSessionId=322838539; 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 08:02:39 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: 7256

...
```

Issue 2 of 2

TOC

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 3.0.04506.30; .NET CLR 1.1.4322;
.NET CLR 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 52

txtEmail=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
...
```

L Hidden Directory Detected 3

TOC

Issue 1 of 3

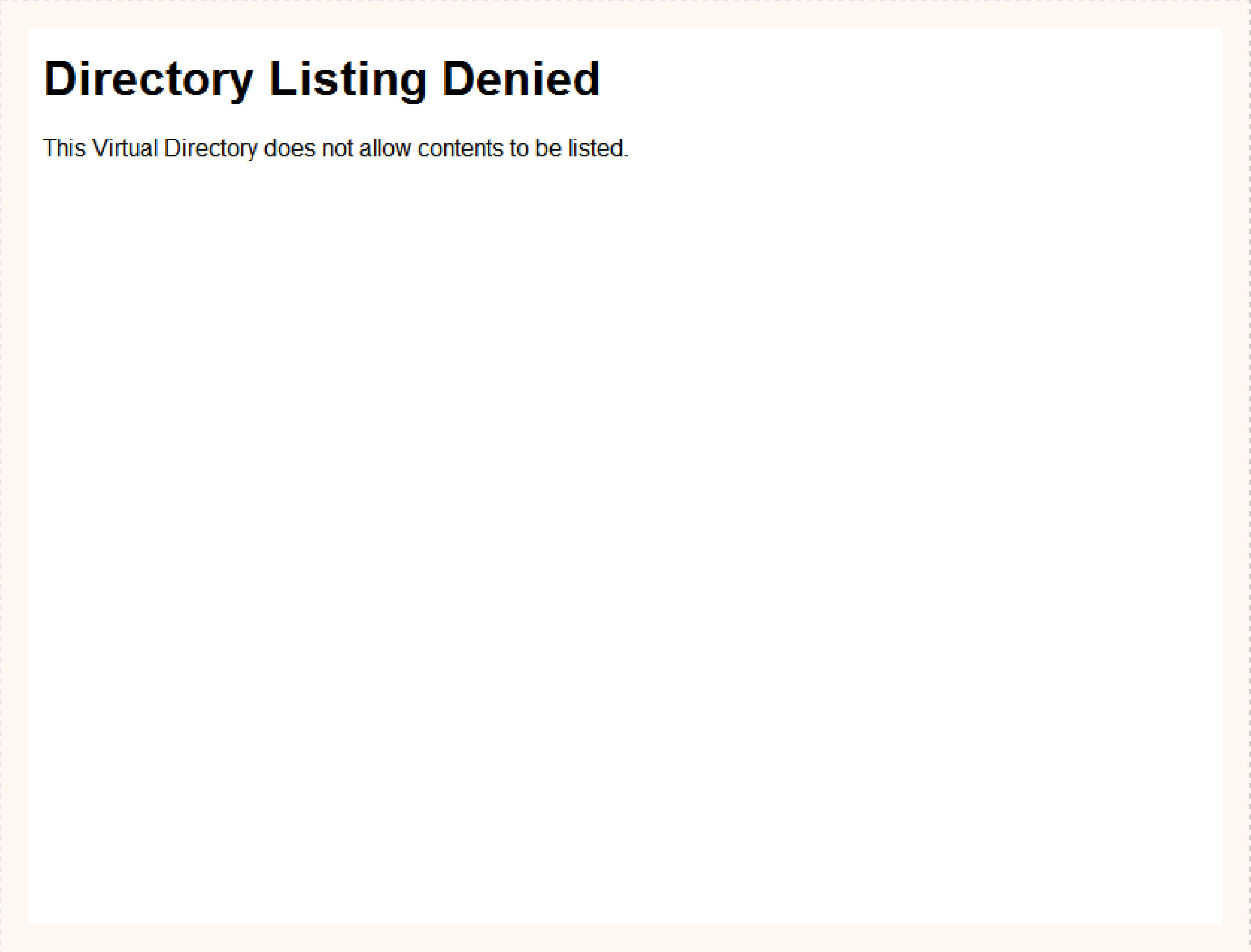
TOC

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.

Issue 2 of 3

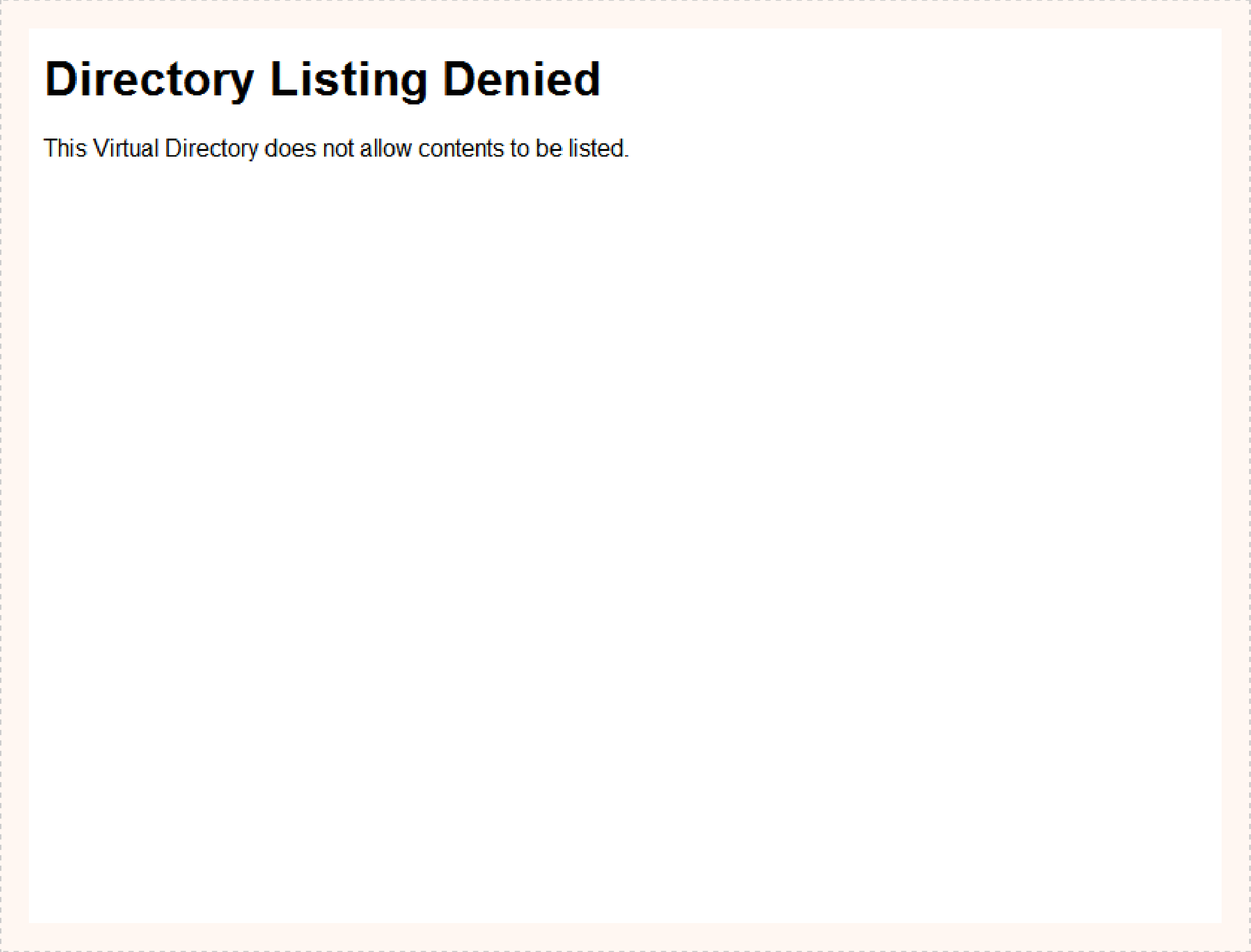
TOC

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.

Issue 3 of 3

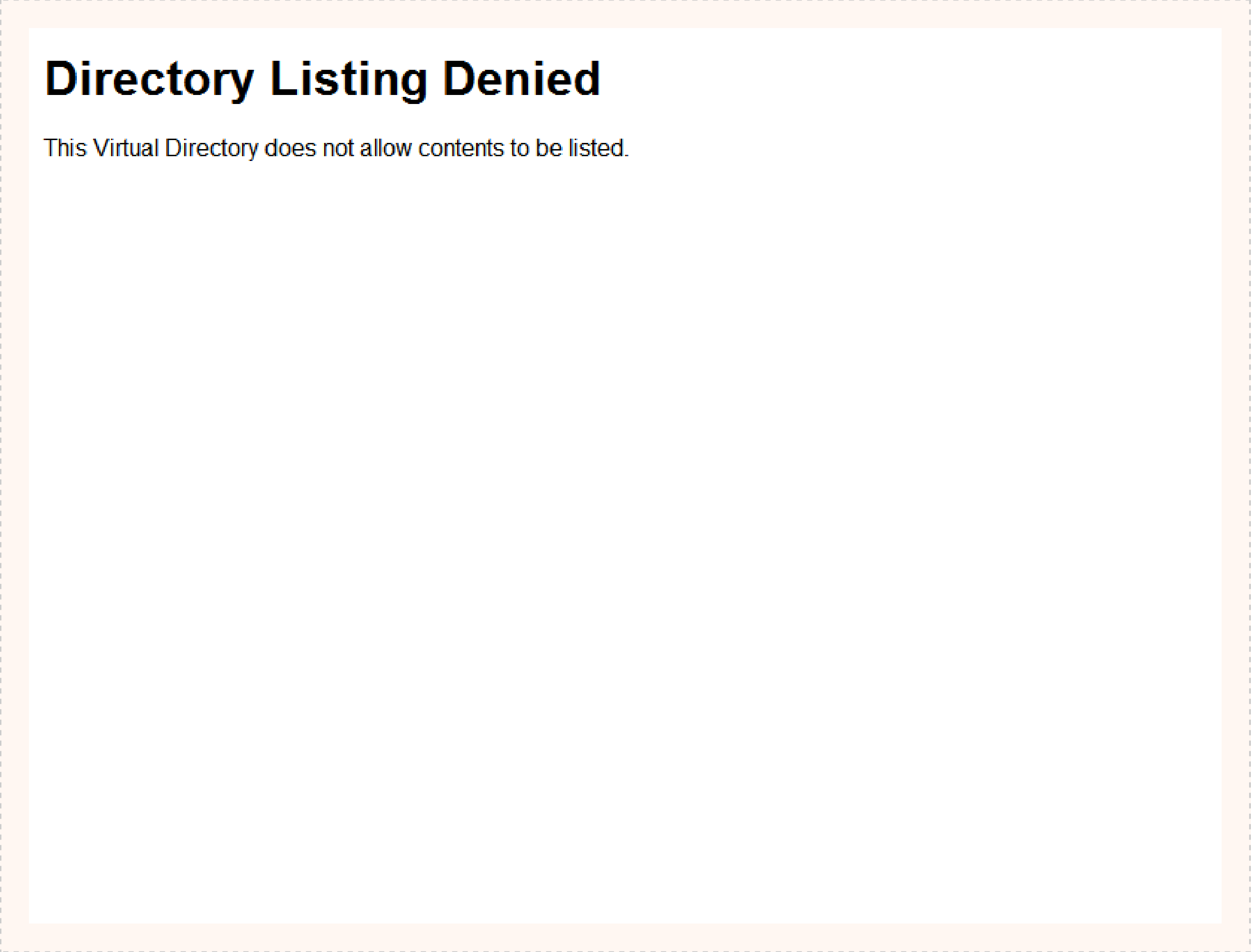
TOC

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.

L Microsoft ASP.NET Debugging Enabled 3

TOC

Issue 1 of 3

TOC

Microsoft ASP.NET Debugging Enabled

Severity:	Low
URL:	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: 2

OK
...
```

Issue 2 of 3

TOC

Microsoft ASP.NET Debugging Enabled

Severity:	Low
URL:	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: 2

OK
...
```

Issue 3 of 3

TOC

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: 2

OK
...
```

L Missing HttpOnly Attribute in Session Cookie 4

TOC

Issue 1 of 4

TOC

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=n5pgfuf5tyl2ds553uu5bn55; path=/; HttpOnly
Set-Cookie: amSessionId=332438668; path=/

...
```

Issue 2 of 4

TOC

Missing HttpOnly Attribute in Session Cookie

Severity:	Low
URL:	http://demo.testfire.net/bank/login.aspx
Entity:	amCreditOffer (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=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 3 of 4

TOC

Missing HttpOnly Attribute in Session Cookie

Severity:	Low
URL:	http://demo.testfire.net/bank/login.aspx
Entity:	amUserId (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=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

TOC

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=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
...
```

L Permanent Cookie Contains Sensitive Session Information 1

TOC

Issue 1 of 1

TOC

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

```
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
...
```

L Unencrypted __VIEWSTATE Parameter 4

TOC

Issue 1 of 4

TOC

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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<h1>Recent Transactions</h1>

<form name="aspnetForm" method="post" action="transaction.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTYzNDg3OTA4NmRk" />

<table border="0" style="padding-bottom:10px;">
    <tr>
        <td valign=top>After</td>
        <td><input name="after" type="text" value="1234" /><br /><span class="credit">mm/dd/yyyy</span></td>
        <td valign=top>Before</td>
        <td><input name="before" type="text" value="1234"  /><br /><span class="credit">mm/dd/yyyy</span></td>
        <td valign=top><input type=submit value=Submit /></td>
    </tr>

...
```

Issue 2 of 4

TOC

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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" 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)&amp;_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+2YBwKw59eKCgKcjoPABw==" />
  <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_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 /><br />
  <span id="_ctl0__ctl0_Content_Main_Label2">News title not found, try again</span>
</form>

...
```

Issue 3 of 4

TOC

Unencrypted __VIEWSTATE Parameter

Severity:	Low
URL:	http://demo.testfire.net/bank/customize.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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" 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">Curent 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>

...
```

Issue 4 of 4

TOC

Unencrypted __VIEWSTATE Parameter

Severity:	Low
URL:	http://demo.testfire.net/admin/login.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

```
...

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

```
<h1>Administration Login</h1>

<!-- Password: Altoro1234 -->

<form name="aspnetForm" method="post" action="login.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTY5ODYzNjk3NWRk" />

<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWBAKm/PqICgKaqvKtBQKWuPeSCgL73pWUBA==" />
  <img id="captcha" src="captcha.aspx" /><br />
  <p>
    <strong>Enter the code shown above:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:CodeNumberTextBox" type="text" id="_ctl0__ctl0_Content_Main_CodeNumberTextBox" /><br /><br />
    <strong>Enter the administrative password:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:Password" type="password" id="_ctl0__ctl0_Content_Main_Password" /><br /><br />
    <input type="submit" name="_ctl0:_ctl0:Content:Main:SubmitButton" value="Submit" id="_ctl0__ctl0_Content_Main_SubmitButton" /><br />

...
```

L Unsigned __VIEWSTATE Parameter 4

TOC

Issue 1 of 4

TOC

Unsigned __VIEWSTATE Parameter

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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<h1>Recent Transactions</h1>

<form name="aspnetForm" method="post" action="transaction.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTYzNDg3OTA4NmRk" />

<table border="0" style="padding-bottom:10px;">
    <tr>
        <td valign=top>After</td>
        <td><input name="after" type="text" value="1234" /><br /><span class="credit">mm/dd/yyyy</span></td>
        <td valign=top>Before</td>
        <td><input name="before" type="text" value="1234"  /><br /><span class="credit">mm/dd/yyyy</span></td>
        <td valign=top><input type=submit value=Submit /></td>
    </tr>

...
```

Issue 2 of 4

TOC

Unsigned __VIEWSTATE Parameter

Severity:	Low
URL:	http://demo.testfire.net/bank/queryxpath.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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" 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)&amp;_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+2YBwKw59eKCgKcjoPABw==" />
  <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_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 /><br />
  <span id="_ctl0__ctl0_Content_Main_Label2">News title not found, try again</span>
</form>

...
```

Issue 3 of 4

TOC

Unsigned __VIEWSTATE Parameter

Severity:	Low
URL:	http://demo.testfire.net/bank/customize.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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" 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">Curent 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>

...
```

Issue 4 of 4

TOC

Unsigned __VIEWSTATE Parameter

Severity:	Low
URL:	http://demo.testfire.net/admin/login.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

```
...

    </td>
    <td valign="top" colspan="3" class="bb">


<h1>Administration Login</h1>

<!-- Password: Altoro1234 -->

<form name="aspnetForm" method="post" action="login.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTY5ODYzNjk3NWRk" />

<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWBAKm/PqICgKaqvKtBQKWuPeSCgL73pWUBA==" />
  <img id="captcha" src="captcha.aspx" /><br />
  <p>
    <strong>Enter the code shown above:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:CodeNumberTextBox" type="text" id="_ctl0__ctl0_Content_Main_CodeNumberTextBox" /><br /><br />
    <strong>Enter the administrative password:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:Password" type="password" id="_ctl0__ctl0_Content_Main_Password" /><br /><br />
    <input type="submit" name="_ctl0:_ctl0:Content:Main:SubmitButton" value="Submit" id="_ctl0__ctl0_Content_Main_SubmitButton" /><br />

...
```

I Application Error 15

TOC

Issue 1 of 15

TOC

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

...
```

Issue 2 of 15

TOC

Application Error

Severity:	Informational
URL:	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:

```
...

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

...
```

Issue 3 of 15

TOC

Application Error

Severity:	Informational
URL:	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:

```
...
```

```
GET /bank/queryxpath.aspx?__VIEWSTATE=%2FwEPDwUKMTEzMDczNTAxOWRk&__EVENTVALIDATION=%2FwEWAwLNx%2B2YBwKw59eKCgKcjoPABw%3D%3D&_ctl0%
3A_ctl0%3AContent%3AMain%3ATextBox1=%27&_ctl0%3A_ctl0%3AContent%3AMain%3AButton1=Query 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/bank/queryxpath.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)


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

...
```

Issue 4 of 15

TOC

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 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:

```
...
GET /bank/ws.asmx?WSDL=%00 HTTP/1.1
Cookie: ASP.NET_SessionId=eu0qbsjngqgirw45q0opxa45; 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://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

TOC

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:

```
...

  </tr>
  <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"><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 System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 6 of 15

TOC

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>
  <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"><span style='color: Red'>System.Data.OleDb.OleDbException: Syntax error
in string in query expression 'accountid=''.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object& executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</span></span>
    <span id="soapResp" name="soapResp" align="center" />
    </td>

...
```

Issue 7 of 15

TOC

Application Error

Severity:	Informational
URL:	http://demo.testfire.net/bank/transfer.aspx
Entity:	transferAmount (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/transfer.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: 67

debitAccount=1001160141&creditAccount=1001160141&transferAmount=%27

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:39:57 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 8 of 15

TOC

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:

```
...

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: 38

uid=%27&passw=demo1234&btnSubmit=Login

HTTP/1.1 500 Internal Server Error
Connection: close
Date: Sun, 22 Jul 2012 08:40:33 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 9 of 15

TOC

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:

```
...

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

TOC

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"><soap:Body><TransferBalanceResponse
xmlns="http://www.altoromutual.com/bank/ws/"><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException:
Syntax error in query expression 'accountid=%27'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
155</Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
...
```

Issue 11 of 15

TOC

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:

```
...
```

```
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-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"><soap:Body><TransferBalanceResponse
xmlns="http://www.altoromutual.com/bank/ws/"><TransferBalanceResult><Success>false</Success><Message>System.Data.OleDb.OleDbException:
Syntax error in query expression 'accountid=%27'.
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextErrorHandling(OleDbHResult hr)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandTextForSingleResult(tagDBPARAMS dbParams, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommandText(Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteCommand(CommandBehavior behavior, Object&amp; executeResult)
   at System.Data.OleDb.OleDbCommand.ExecuteReaderInternal(CommandBehavior behavior, String method)
   at System.Data.OleDb.OleDbCommand.ExecuteScalar()
   at Altoro.Services.TransferBalance(MoneyTransfer transDetails) in d:\downloads\AltoroMutual_v6\website\App_Code\WebService.cs:line
146</Message></TransferBalanceResult></TransferBalanceResponse></soap:Body></soap:Envelope>
...
```

Issue 12 of 15

TOC

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:

```
...

    <transferDate>%27</transferDate>
    <debitAccount>1001160141</debitAccount>
    <creditAccount>1001160141</creditAccount>
    <transferAmount>1234</transferAmount>
   </transDetails>
  </TransferBalance>
 </soap:Body>
</soap:Envelope>

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-AspNet-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/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><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

TOC

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:

```
...

    <transferDate>2000-01-01</transferDate>
    <debitAccount>1001160141</debitAccount>
    <creditAccount>1001160141</creditAccount>
    <transferAmount>%27</transferAmount>
   </transDetails>
  </TransferBalance>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 500 Internal Server Error
Date: Sun, 22 Jul 2012 08:37:12 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/xml; charset=utf-8
Content-Length: 478

<?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"><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>
...
```

Issue 14 of 15

TOC

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 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 175

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%2FwEPDwUKMTYzNDg3OTA4NmRk&__EVENTVALIDATION=%2FwEWBgKV3oKhDgK3oeuaBAK3oaesDgK3oZPRBQK3oa%
2BJCgK3oZuuAQ%3D%3D&after=1234&before=%27

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

TOC

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:

```
...

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 3.0.4506.2152; .NET CLR 3.5.30729)
Content-Length: 175

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=%2FwEPDwUKMTYzNDg3OTA4NmRk&__EVENTVALIDATION=%2FwEWBgKV3oKhDgK3oeuaBAK3oaesDgK3oZPRBQK3oa%
2BJCgK3oZuuAQ%3D%3D&after=%27&before=1234

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

...
```

I Application Test Script Detected 1

TOC

Issue 1 of 1

TOC

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: **Test Response**

```
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/msword, */*
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
```

```
...

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


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


...
```

I Email Address Pattern Found 3

TOC

Issue 1 of 3

TOC

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.
```

```
Please enter your email below and we will automatically notify of noteworthy events.</p>

<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>

...
```

Issue 2 of 3

TOC

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:

```
...

            <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 />
<b>jsmith@demo.testfire.net</b></p></span>

</div>


    </td>
  </tr>
</table>



...
```

Issue 3 of 3

TOC

Email Address Pattern Found	
Severity:	Informational
URL:	http://demo.testfire.net/bank/mozxpath.js
Entity:	mozxpath.js (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:

```
...

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)

...
```

I HTML Comments Sensitive Information Disclosure 5

TOC

Issue 1 of 5

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;"/>
        <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>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<!-- To modify account information do not connect to SQL source directly.  Make all changes
through the admin page. -->

<h1>Account History - <span id="_ctl0__ctl0_Content_Main_accountid">1001160141</span></h1>

<table width="590" border="0">
  <tr>
    <td colspan=2>
      <table cellSpacing="0" cellPadding="1" width="100%" border="1">
        <tr>
          <th colSpan="2">

...
```

Issue 2 of 5

TOC

HTML Comments Sensitive Information Disclosure

Severity:	Informational
URL:	http://demo.testfire.net/bank/login.aspx
Entity:	To get the latest admin login, please contact SiteOps at 415-555-6159 (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

```
...

        </ul>
    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<h1>Online Banking Login</h1>

<!-- To get the latest admin login, please contact SiteOps at 415-555-6159 -->
<p><span id="_ctl0__ctl0_Content_Main_message" style="color:#FF0066;font-size:12pt;font-weight:bold;"></span></p>

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

</div>

<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>


    </td>
  </tr>

...
```

Issue 3 of 5

TOC

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>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<h1>Altoro Mutual
    <span id="_ctl0__ctl0_Content_Main_lblType">Gold</span>
    Visa Application</h1>

...

...

    Visa Application</h1>

<!--
    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 an 7.9% APR simply enter your password below.</p><form method="post" name="Credit" action="apply.aspx"><table
border=0><tr><td>Password:</td><td><input type="password" name="passwd"></td></tr><tr><td></td><td><input type="submit" name="Submit"
value="Submit"></td></tr></table></form></span>

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

</div>


    </td>
  </tr>
</table>

...
```

Issue 4 of 5

TOC

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

```
...

Content-Type: text/html; charset=iso-8859-1
Content-Length: 7861



<!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 width="25%" class="cc bt br 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=inside.htm">INSIDE ALTORO MUTUAL</a></div></td>
  </tr>
  <tr>
    <td valign="top" class="cc br bb">
        <br style="line-height: 10px;"/>
        <b>I WANT TO ...</b>
        <ul class="sidebar">
            <li><a id="_ctl0__ctl0_Content_MenuHyperLink1" href="application.aspx">View Application Values</a></li>
            <li><a id="_ctl0__ctl0_Content_MenuHyperLink2" href="admin.aspx">Edit Users</a></li>
        </ul>
    </td>
    <td valign="top" colspan="3" class="bb">


<div class="fl" style="width: 99%;">

<script language="javascript">

function confirmpass(myform)
{
  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>

<!-- Be careful what you change.  All changes are made directly to Altoro.mdb database. -->

<h1>Edit User Information</h1>

<table width="100%" border="0">
<form id="addAccount" name="addAccount" action="admin.aspx" method="post">
  <tr>
    <td colspan="4">
      <h2>Add an account to an existing user.</h2>
    </td>
  </tr>
  <tr>
    <th>
      Users:
    </th>
    <th>
      Account Types:
    </th>
    <th>&nbsp;</th>
    <th>&nbsp;</th>
  </tr>
  <tr>
    <td>
      <select id="" name="" ><option value="1">1 admin</option><option value="2">2 tuser</option><option value="100116013">100116013
sjoe</option><option value="100116014">100116014 jsmith</option><option value="100116015">100116015 cclay</option><option
value="100116018">100116018 sspeed</option></select>
    </td>
    <td>
      <Select name="accttypes">
        <option Value="Checking">Checking</option>
        <option Value="Savings" Selected>Savings</option>
        <option Value="IRA">IRA</option>
      </Select></td>
    <td></td>
    <td><input type="submit" value="Add Account"></td>

...

...

      <Select name="accttypes">
        <option Value="Checking">Checking</option>
        <option Value="Savings" Selected>Savings</option>
        <option Value="IRA">IRA</option>
      </Select></td>
    <td></td>
    <td><input type="submit" value="Add Account"></td>
  </tr>
</form>
<form id="changePass" name="changePass" action="admin.aspx" method="post" onsubmit="return confirmpass(this);">
  <tr>
    <td colspan="4"><h2>Change user's password.</h2></td>
  </tr>
  <tr>
    <th>
      Users:
    </th>
    <th>
      Password:
    </th>
    <th>
      Confirm:
    </th>
    <th>&nbsp;</th>
  </tr>
  <tr>
    <td>
      <select id="" name="" ><option value="1">1 admin</option><option value="2">2 tuser</option><option value="100116013">100116013
```

```
sjoe</option><option value="100116014">100116014 jsmith</option><option value="100116015">100116015 cclay</option><option
value="100116018">100116018 sspeed</option></select>
    </td>
    <td>
      <input type="--begin_...
```

Issue 5 of 5

TOC

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

```
...

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">

...

...

   </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">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 br 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 br bb"><div id="Header4"><a id="_ctl0__ctl0_Content_LinkHeader4" class="focus" href="../default.aspx?
```

```
content=inside.htm">INSIDE ALTORO MUTUAL</a></div></td>
  </tr>
  <tr>
    <td valign="top" class="cc br bb">
        <br style="line-height: 10px;"/>
        <a id="_ctl0__ctl0_Content_CatLink1" class="subheader" href="../default.aspx?content=personal.htm">PERSONAL</a>
        <ul class="sidebar">

...

...

            <li><a id="_ctl0__ctl0_Content_MenuHyperLink15" href="../cgi.exe">Locations</a></li>
            <li><a id="_ctl0__ctl0_Content_MenuHyperLink16" href="../default.aspx?content=inside_investor.htm">Investor 
Relations</a></li>
            <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">


<h1>Administration Login</h1>

<!-- Password: Altoro1234 -->

<form name="aspnetForm" method="post" action="login.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTY5ODYzNjk3NWRk" />

<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWBAKm/PqICgKaqvKtBQKWuPeSCgL73pWUBA==" />
  <img id="captcha" src="captcha.aspx" /><br />
  <p>
    <strong>Enter the code shown above:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:CodeNumberTextBox" type="text" id="_ctl0__ctl0_Content_Main_CodeNumberTextBox" /><br /><br />

...

...


<form name="aspnetForm" method="post" action="login.aspx" id="aspnetForm">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUKMTY5ODYzNjk3NWRk" />

<input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" value="/wEWBAKm/PqICgKaqvKtBQKWuPeSCgL73pWUBA==" />
  <img id="captcha" src="captcha.aspx" /><br />
  <p>
    <strong>Enter the code shown above:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:CodeNumberTextBox" type="text" id="_ctl0__ctl0_Content_Main_CodeNumberTextBox" /><br /><br />
    <strong>Enter the administrative password:</strong><br />
    <input name="_ctl0:_ctl0:Content:Main:Password" type="password" id="_ctl0__ctl0_Content_Main_Password" /><br /><br />
    <input type="submit" name="_ctl0:_ctl0:Content:Main:SubmitButton" value="Submit" id="_ctl0__ctl0_Content_Main_SubmitButton" /><br />
  </p>
  <p><span id="_ctl0__ctl0_Content_Main_MessageLabel"></span></p>
</form>

<script>
window.onload = document.forms[1].elements[1].focus();
</scrip...
```

I Possible Server Path Disclosure Pattern Found 1

TOC

Issue 1 of 1

TOC

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:

```
...

<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 <br />
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.
  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>

...
```