






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

Process:

1. Using who-is tool find out the who-is record for demo.testfire.net

Name Servers	ASIA3.AKAM.NET (has 52,647 domains) EUR2.AKAM.NET (has 52,647 domains) EUR5.AKAM.NET (has 52,647 domains) NS1-206.AKAM.NET (has 52,647 domains) NS1-99.AKAM.NET (has 52,647 domains) USC2.AKAM.NET (has 52,647 domains) USC3.AKAM.NET (has 52,647 domains) USW2.AKAM.NET (has 52,647 domains)
IP Address	65.61.137.117 - 2 other sites hosted on this server
IP Location	 - Texas - Windcrest - Rackspace Inc.
ASN	 AS33070 RMH-14 - Rackspace Hosting, US (registered Sep 24, 2004)
Domain Status	Registered And Active Website

2. Also identify the webserver and version.

Wappalyzer	
Web Framework	Web Server
 Microsoft ASP.NET 2.0.50727	IIS IIS
	Operating System
	 Windows Server
Website Title	 Altoro Mutual
Server Type	Microsoft-IIS/8.0
Response Code	200
SEO Score	92%
Terms	365 (Unique: 214, Linked: 79)
Images	6 (Alt tags missing: 5)
Links	39 (Internal: 38, Outbound: 1)

3. Use net craft to get the technologies.

Site Technology

Fetchd on 1st July 2018

Server-Side

Includes all the main technologies that Netcraft detects as running on the server such as PHP.

Technology	Description	Popular sites using this technology
Using ASP.NET ↗	ASP.NET is running on the server	www.microsoft.com , www.wordreference.com , www.t-online.de

Doctype

A Document Type Declaration, or DOCTYPE, is an instruction that associates a particular SGML or XML document (for example, a webpage) with a Document Type Definition (DTD).

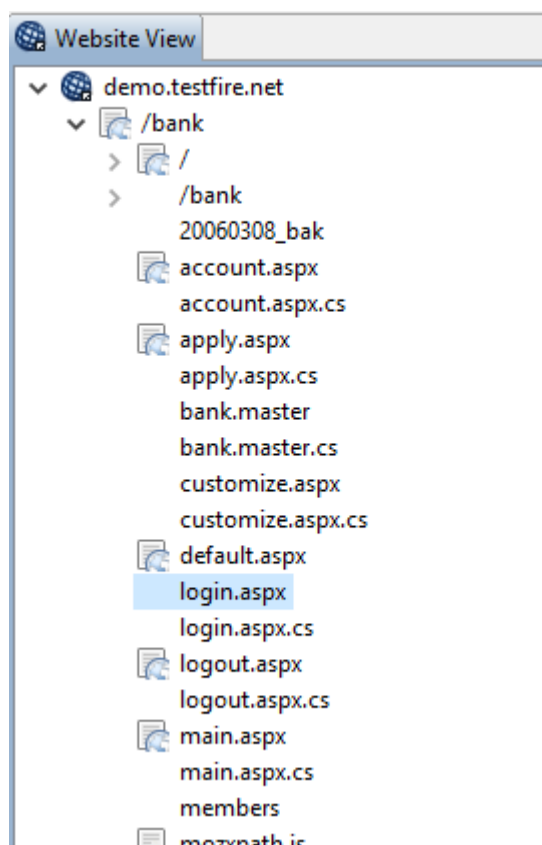
Technology	Description	Popular sites using this technology
XHTML ↗	Extended version of the Hypertext Markup Language	www.dailymail.co.uk , www.repubblica.it , www.nk.ca

CSS Usage

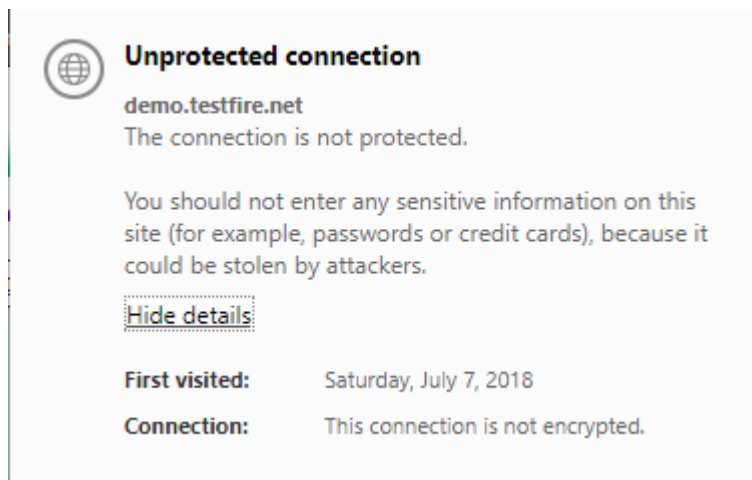
Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language (such as XHTML).

Technology	Description	Popular sites using this technology
External ↗	Styles defined within an external CSS file	www.bbc.co.uk , www.bbc.com , www.facebook.com

4. Use tools like httrack to get the website view in a hierarchical structure it's used to identify the layout login page in the website.



5. The Login page is not encrypted and hence the passwords are sent in clear text and is vulnerable to mitm.



The screenshot shows a web browser window with the "Mutual" logo and a navigation bar with "PERSONAL" and "SMALL BUSINESS" tabs. The main heading is "Online Banking Login". Below it, a red error message states: "Login Failed: We're sorry, but username was not found in our database. Please try again." The login form includes fields for "Username:" (containing "admin123") and "Password:" (containing "admin"), and a "Login" button.

Overlaid on the right side of the browser window is a Wireshark packet capture window. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The packet list shows several TCP and HTTP packets. The selected packet (Frame 6) is an HTTP POST request to "/bank/login.aspx". The packet details pane shows the following structure:

- Frame 6: 769 bytes on wire (6152 bits), 769 bytes captured (6152 bits) on interface 0
- Ethernet II, Src: IntelCor_0f:c2:60 (f8:94:c2:0f:c2:60), Dst: Cisco_9e:7a:1d (70:7d:b9:9e:7a:1d)
- Internet Protocol Version 4, Src: 192.168.51.204, Dst: 65.61.137.117
- Transmission Control Protocol, Src Port: 1048, Dst Port: 80, Seq: 1, Ack: 1, Len: 715
- Hypertext Transfer Protocol
- HTML Form URL Encoded: application/x-www-form-urlencoded
 - Form item: "uid" = "admin123"
 - Form item: "passw" = "admin"
 - Form item: "btnSubmit" = "Login"

Vulnerability: Clear text Password over HTTP

Description: A form with a password input field that submits to an insecure (HTTP) target. Password values should never be sent in the clear across insecure channels. This vulnerability could result in unauthorized disclosure of passwords to passive network attackers.

Risk: High

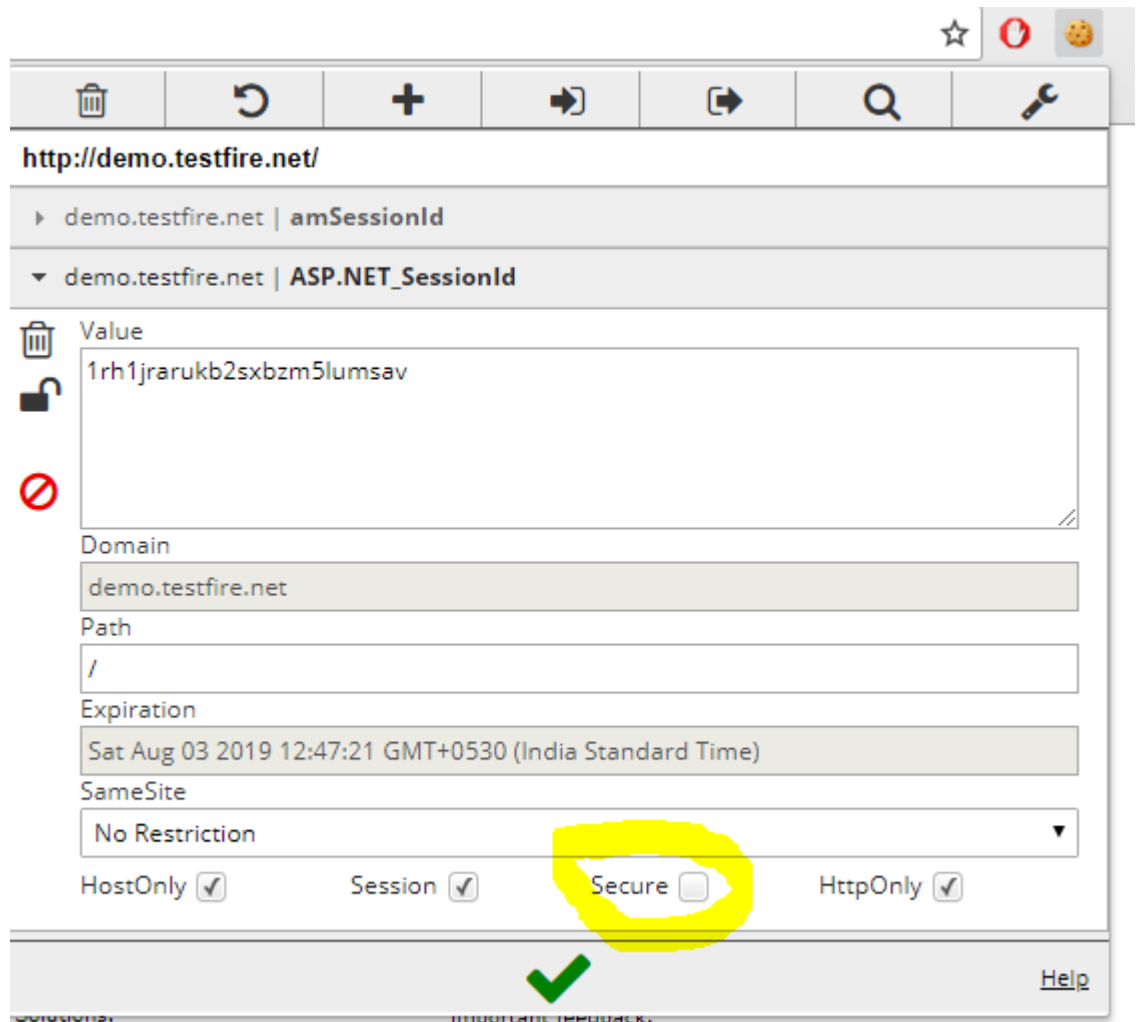
Impact:

- Detected a form that can cause a password submission over an insecure channel.
- This could result in disclosure of passwords to network eavesdroppers.

Remediation:

Passwords should never be sent over cleartext. The form should submit to an HTTPS target.

6. Use cookie editor to view the cookies that are used by the website we can see that the cookie's secure flag is not set.



Vulnerability: Session Cookie without Secure Flag

Description: The secure flag is an option that can be set by the application server when sending a new cookie to the user within an HTTP Response. The purpose of the secure flag is to prevent cookies from being observed by unauthorized parties due to the transmission of the cookie in clear text.

Risk: High

Impact:

- Cookies can be exposed to network eavesdroppers.
- Session cookies are authentication credentials; attackers who obtain them can get unauthorized access to affected web applications.

Remediation:

When creating the cookie in the code, set the secure flag to true.

7. Perform static code analysis of the login page we can see that the post action is `confirminput()` js function.

Which do not have any input validation thus exposing to vulnerabilities such as SQL injection.

```
<div class="+1" style="width: 99%;">
  <h1>Online Banking Login</h1>
  <!-- To get the latest admin login, please contact SiteOps at 415-555-6159 -->
  <p>...</p>
  <form action="login.aspx" method="post" name="login" id="login" onsubmit="return (confirminput(login));">...</form>
</div>
```

```
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);
  }
}
```

8. Enter the ' or 1=1—in the text fields to bypass the authentication.

PERSONAL	SMALL BUSINESS
<h2>Online Banking Login</h2> <p>Username: <input type="text" value="' or 1=1--"/></p> <p>Password: <input type="password" value="....."/></p> <p><input type="button" value="Login"/></p> <h2>Hello Admin User</h2> <p>Welcome to Altoro Mutual Online.</p> <p>View Account Details: <input type="button" value="▼"/> <input type="button" value="GO"/></p>	

Vulnerability: SQL Injection

Description: SQL Injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements (also commonly referred to as a malicious payload) that control a web application's database server (also commonly referred to as a Relational Database Management System – RDBMS).

Risk: High

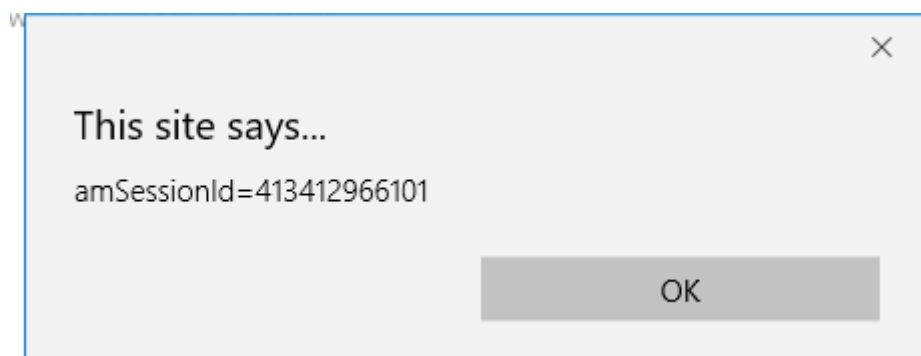
Impact

- These vulnerabilities can be exploited by remote attackers to gain unauthorized read or write access to the underlying database.
- Exploitation of SQL injection vulnerabilities can also allow for attacks against the logic of the application.
- Attackers may be able to obtain unauthorized access to the server hosting the database.

Remediation

- The developer should review the request and response against the code to manually verify whether or not a vulnerability is present.
- The best defence against SQL injection vulnerabilities is to use parameterized statements.
- Sanitizing input can prevent these vulnerabilities. Variables of string types should be filtered for escape characters, and numeric types should be checked to ensure that they are valid.
- Configuring database access controls can limit the impact of exploited vulnerabilities. This is a mitigating strategy that can be employed in environments where the code is not modifiable.

9. Enter the string `<script>alert(document.cookie)</script>` in the search box as there is no input validation.



Vulnerability: Cross-site scripting

Description: Cross-site scripting (XSS) is a class of vulnerabilities affecting web applications that can result in security controls implemented in browsers being circumvented. When a browser visits a page on a website, script code originating in the website domain can access and manipulate the DOM (document object model), a representation of the page and its properties in the browser.

Risk: High

Impact:

- XSS is generally a threat to web applications which have authenticated users or are otherwise security sensitive.
- Malicious code may be able to manipulate the content of the site, changing its appearance and/or function for another user.
- This includes modifying the behaviour of the web application (such as redirecting forms, etc).
- The code may also be able to perform actions within the application without user knowledge.
- Script code can also obtain and retransmit cookie values if they haven't been set HttpOnly.

Remediation:

- The developer must identify how the untrustworthy data is being output to the client without adequate filtering.
- There are various language/platform specific techniques for filtering untrustworthy data.
- General rules for preventing XSS can be found in the recommended OWASP XSS Prevention Cheat Sheet (see references).

10. Brute force the login with well-known username and passwords like admin, password, 12345 e.t.c.

Online Banking Login

Username:

Password:

Here we try with admin, admin which is the correct login and automatically logs us in.

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details:

Vulnerability: Predictable Login Credentials

Description: Nowadays web applications often make use of popular open source or commercial software that can be installed on servers with minimal configuration or customization by the server administrator. Often these applications, once installed, are not properly configured and the default credentials provided for initial authentication and configuration are never changed. These default credentials are well known by penetration testers and, unfortunately, also by malicious attackers, who can use them to gain access to various types of applications.

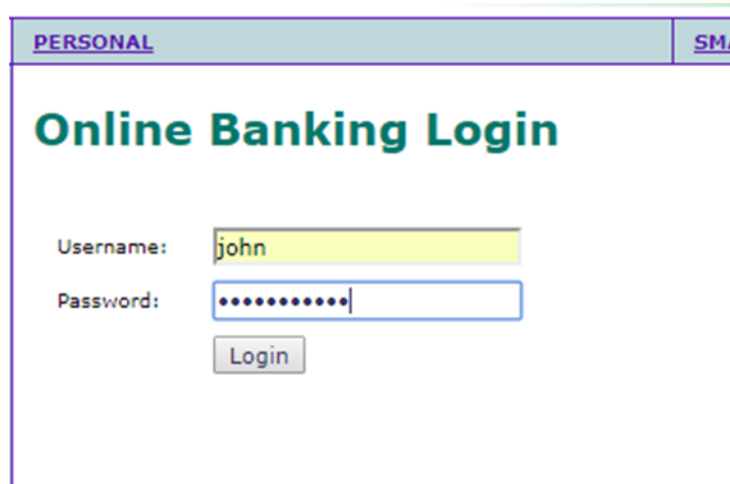
Risk: High

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

Remediation:

- Change the login credentials to a stronger combination

11. Navigate to the login page and enter some special characters to print the error message details like stack trace.



PERSONAL SM

Online Banking Login

Username: john

Password:

Login

An Error Has Occurred

Summary:

Syntax error (missing operator) in query expression 'username = 'john' AND password = '' or 1=1â€¢'.

Error Message:

System.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression 'username = 'john' AND password = '' or 1=1â€¢'. 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.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) at System.Data.Common.DbDataAdapter.Fill(DataSet dataSet, String srcTable) at Altoro.Authentication.ValidateUser(String userName, String password) in c:\downloads\AltoroMutual_v6\website\bank\login.aspx.cs:line 68 at Altoro.Authentication.Page_Load(Object sender, EventArgs e) in c:\downloads\AltoroMutual_v6\website\bank\login.aspx.cs:line 33 at System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) at System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) at System.Web.UI.Control.OnLoad(EventArgs e) at System.Web.UI.Control.LoadRecursive() at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)

Vulnerability: Improper Error Handling

Description: Improper handling of errors can introduce a variety of security problems for a web site. The most common problem is when detailed internal error messages such as stack traces, database dumps, and error codes are displayed to the user (hacker). These messages reveal implementation details that should never be revealed. Such details can provide hackers important clues on potential flaws in the site and such messages are also disturbing to normal users.

Severity: Medium

Impact:

- Internal implementation of critical server side scripts.
- Versions, services and libraries that are being used.

Remediation:

- A specific policy for how to handle errors should be documented, including the types of errors to be handled and for each, what information is going to be reported back to the user, and what information is going to be logged.
- In the implementation, ensure that the site is built to gracefully handle all possible errors.