**SQL Injection:**

To test enter below string in search box. It will drop UserData table. Below is given query to re-create table

Live';drop table UserData;--

CREATE TABLE [dbo].[UserData](

[UserId] [int] NOT NULL,

[Name] [varchar](50) NULL,

[Address] [varchar](500) NULL,

[EmailId] [varchar](100) NULL

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

INSERT [dbo].[UserData] ([UserId], [Name], [Address], [EmailId]) VALUES (1, N'John', N'Banglore', N'john@gmail.com')

INSERT [dbo].[UserData] ([UserId], [Name], [Address], [EmailId]) VALUES (2, N'Martin', N'Pune', N'martin@gmail.com')

**HttpForbiddenHandler:**

Exclude unnecessary unused file extension. In our example we have excluded xml file. But allowed csv file. Below URL can be used to verify the same..

Allowed

<http://localhost:26641//Data/UserData.csv>

Not Allowed

<http://localhost:26641/Data/UserData.xml>

This configuration need to be specified for all file extension required to be blocked.

<system.web>

<httpHandlers>

<add verb="\*" path="\*.xml" type="System.Web.HttpForbiddenHandler" />

</httpHandlers>

</system.web>

<system.webServer>

<handlers>

<add verb="\*" path="\*.xml" type="System.Web.HttpForbiddenHandler" name="XML"/>

</handlers>

</system.webServer>

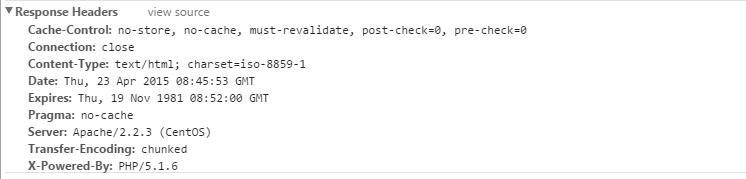
The httpHandlers element is used on sites running on IIS 5 – 6, or IIS 7.x in Classic mode (IIS 6 compatibility mode). The handlers element is used on sites running on IIS 7.x (Integrated mode).

**Cross Site Scripting:**

In write Review text area enter below text and save it. After that whenever page get loaded it will execute script in below text and open new window.

This product is good.Checkout some more<script>window.open('http://www.w3schools.com/', '\_blank');</script>

**Information Disclosure (HTTP Response Headers)**



Above data can be seen in network tab of browser which shows the server details which can help user to predict some of setting and get malicious information.

Below piece of code to be added to Global.aspx to remove some information from header.

protected void Application\_PreSendRequestHeaders(object sender, EventArgs e)

{

// Remove the "Server" HTTP Header from response

var app = sender as HttpApplication;

if (null != app && null != app.Context && null != app.Context.Response)

{

NameValueCollection headers = app.Context.Response.Headers;

if (null != headers)

{

headers.Remove("Server");

headers.Remove("X-AspNet-Version");

headers.Remove("X-AspNetMvc-Version");

headers.Remove("X-Powered-By");

}

}

}

Business Logic Bypassing (Client Side):

Relying only on client side validation can cause malicious users can easily bypass logic (Validation)

Steps To Reproduce:

1. Select an Item from Album.
2. Add item to Cart.
3. Click on **checkout.**
4. Enter Promo code as “FREE”.
5. Click on “**Submit**” button without filling any other details other than Promo code. You will see validation messages.
6. Disable JavaScript (How to disable and enable JavaScript [Click Here](http://enable-javascript.com/)).
7. Refresh the Checkout page.
8. Now click on “**Submit**” button without filling any other details other than Promo code. You can submit data without any validation messages.

Application errors with stack trace:

Steps To Reproduce:

1. Select an Item from Album.
2. When you go to Item Details page you will see the URL something Like <http://localhost:26641/Store/Details/6>
3. Change the Integer value appended at the end of the URL to some characters. (In above example from 6 to “some Characters”).
4. Press Enter key you will get the error message with Stack trace.

Always set a custom error page:

Steps To Reproduce:

1. Select an Item from Album.
2. When you go to Item Details page you will see the URL something Like <http://localhost:26641/Store/Details/6>
3. Change the Integer value appended at the end of the URL to 0 (Zero). (In above example from 6 to 0 (Zero))
4. Press Enter key you will get the source level error message.