OWASP SAMM

Getting Started

June 2, 2016

# Environment

## Runtime Environment:

This OWASP SAMM implementation is targeted for a Microsoft stack consisting of IIS and SQL Server. A supported version of Windows Server and SQL Server is recommended. The application may work on older versions.

The site relies on Windows Integrated Authentication, so this should be turned on. There is an option to gather additional user data by utilizing LDAP to get data from Active Directory. See the configuration section for more information.

* HTTP Server: IIS 7x+ (Windows Server 2008 R2 or newer)
* Database Server: SQL Server 2012+, SQL Server Express 2012 or newer

IIS and SQL Server can be on the same box or separate boxes.

## Development Environment:

The solution was developed with a current version of Microsoft Visual Studio. Any current version should work:

* Microsoft Visual Studio 2015 U2
* Microsoft Visual Studio 2015 Community Edition
* Microsoft Visual Studio 2013 U4+

One Visual Studio tool is required – If you are using a 2015 version of Visual Studio, you’ll need: Web Compiler (by Mads Kristensen) – this tool converts the site’s .less file(/Content/Site.less) to css. If you are using the 2013 version of Visual Studio, you’ll need: WebEssentials (by Mads Kristensen). WebEssentials contains the web compiler needed to convert the .less files to .css. The tool can be downloaded from Visual Studio Gallery. Go to Tools | Extensions and Updates…, select Online and enter the tool name in the Search box.

SQL Server Database Tools is recommended, particularly for publishing the database. For Visual Studio 2015, it is usually included when Visual Studio is installed. If you are using Visual Studio 2013, you may have to add it manually via Tools | Extensions and Updates…

Visual Studio can connect to a SQL Server database with the Server Explorer component. This is sufficient for most cases. If you plan to do any database changes, consider installing SQL Server Management Studio.

# Deployment Steps

1. Configure the OWASP SAMM site on the IIS server
2. Identify the instance of SQL Server that will be used to host the database
3. Open the OWASP SAMM solution in Visual Studio
4. Publish the database to the database server. This is a onetime event that will create the database and all the tables for the application. Specify the database name OWASPSAMMOS (case doesn’t matter) in the publish dialog.
5. Run the database script InitialLoadDatabase.sql. This script will populate the empty database tables that were created in the previous step. The first line of the script is a USE statement. Make sure the database name matches the name of the database created in the previous step. The script should only be run once with an empty database.
6. Open the web.config file and configure it for your local environment
   1. LDAP
   2. Database Connection String
   3. Log4net file path
7. Publish the site to the IIS server
8. Open a browser and navigate to the site

# Configuring the Site in IIS

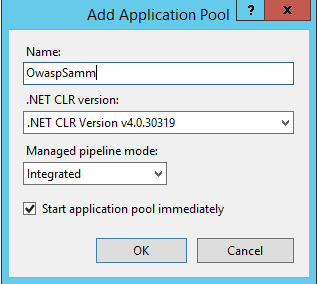
It is assumed that whomever is setting up the site has knowledge of Windows Server and a basic knowledge of IIS and SQL Server. These steps are guidance for setting up the site and configuring IIS and SQL Server

## Service Account

It is recommended that the site have its own application pool and the application pool should run with a service account. If the database is hosted on the same box as IIS, then a local account can be used. If the database and IIS are hosted on separate boxes, a domain account should be used.

## Set up the Application Pool

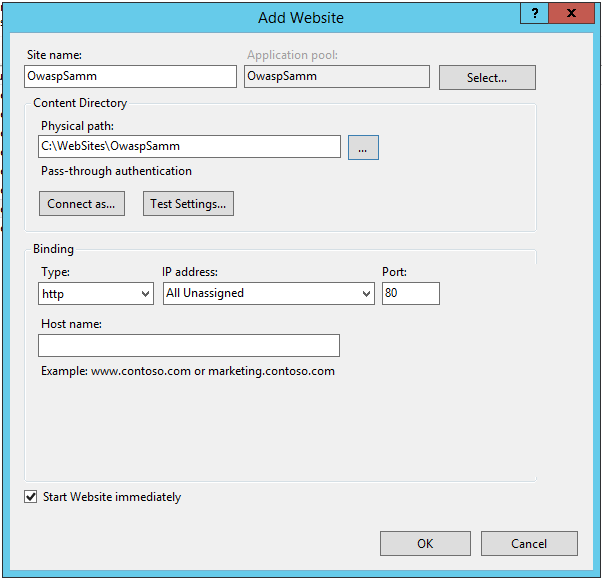
* Obtain an account you can use as a service account. It can be a local account or a domain account. No special privileges are required
* Start Internet Information Services (IIS) Manager
* Click on Application Pools
* Add an application pool with the following values:



* Right click the application pool and select **Advanced Settings**
* Locate the *Identity* item and change the account to the service account
* Save the changes and start the app pool

## Set up the site

* Right click Sites and select **Add Website…**
* Fill in the form and click OK



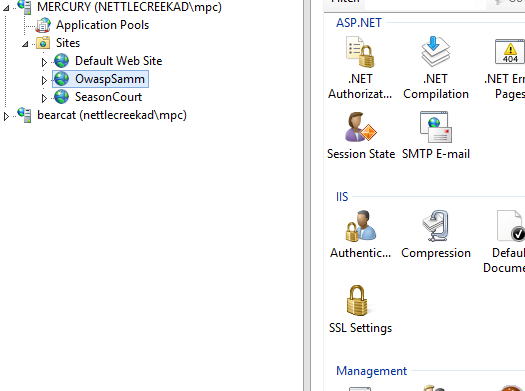
Site name: OwaspSamm

Application Pool: Name of app pool created in the previous step

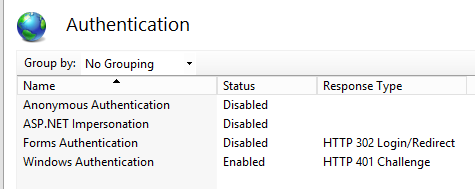
Physical path: A suitable location on the server

*Note: If this site is the default site, you can leave it as port 80. If there is already a site on port 80, you’ll have to choose another port.*

* Select the Site in the Site Tree and double-click the Authentication icon



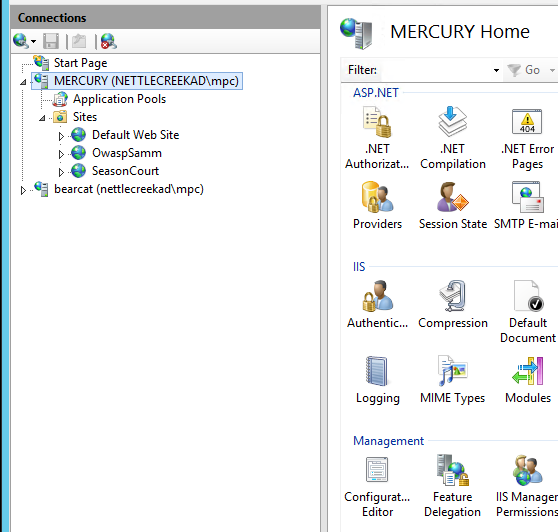
* Set Anonymous Authentication to Disabled
* Set Windows Authentication to Enabled



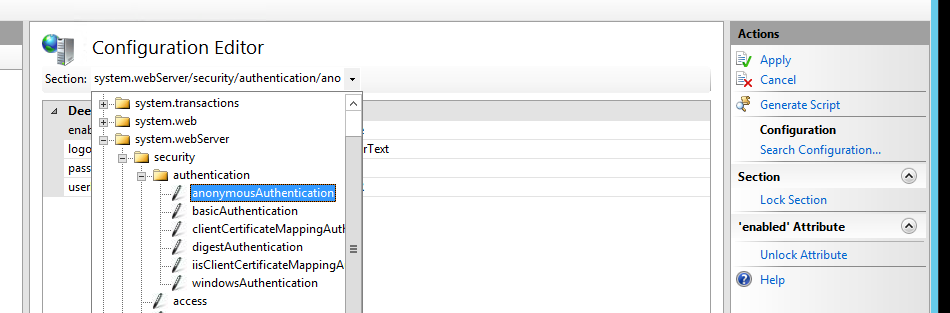
## Configure IIS

In earlier versions of IIS, setting the Authentication for the site was sufficient for running the site. Newer versions of IIS, shipping with Windows Server 2012, require changes to the IIS configuration to allow the site settings to override the IIS Authentication settings. Follow these steps to update the IIS configuration to allow the site to override the default IIS settings.

* From IIS Manager select the server in the Connection tree
* Double-click the Connection Editor icon



* The Configuration Editor will open in the middle section of the console.
* Use the Section dropdown and select **system.webServer** | **security** | **authentication** | **anonymousAuthentication**



* Click the **Unlock Attribute** link in the Actions section on the right, then the **Apply** link
* Select the **windowsAuthentication** attribute and do the same thing, **Unlock** and **Apply**

# SQL Server

The OWASP SAMM application uses SQL Server to store the assessment data. An instance of SQL Server is required to run the application. The instance can reside on the same box as IIS or on its own box. Any supported version of SQL Server from Express to Enterprise is acceptable.

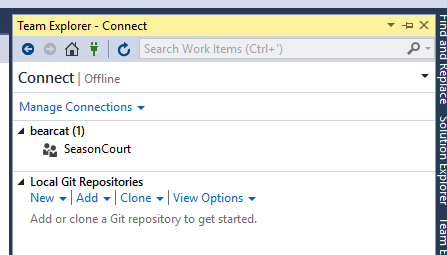
# Opening the Solution

The OWASP SAMM solution is a .net project. Visual Studio is recommended for opening the application and deploying the various pieces of the project. If you have a current version of Visual Studio (2015 or 2013) you should be fine. If you don’t have Visual Studio, the 2015 Community edition works fine and is free.

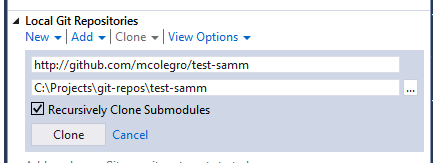
The project is hosted on GitHub. VS 2015 ships with some support for GitHub. If you have an older version, you may have to add some GitHub extensions to get the functionality needed to access a GitHub repository.

The following instructions are based on VS 2015

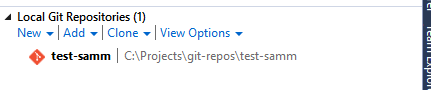
* Start Visual Studio
* Open Team Explorer
* Click the Manage Connections link



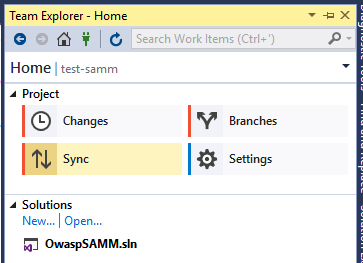
* Clone the GitHub repository. Click the **Clone** link and fill in the form with the appropriate path to the GitHub repository then click the **Clone** button. For the purpose of this document, a sample repository is being used.



* Visual Studio will set up the local repository and copy the files to it. It will also add the repository to the list of repositories.



* Double-click the repository. Team Explorer will switch to the Home view. The solution file will be listed under solutions.



* Double-click the solution file to open the solution.

# The Solution

Open Solution Explorer to view the solution. The solution contains three projects.

|  |  |
| --- | --- |
| OwaspSAMM.Web | Main project – MVC 5 Web site |
| OwaspSAMM.DAL | Data Access Layer – contains all the code to access the database |
| OwaspSAMM.Database | Database project |

## Configuring the Site

Most configuration settings are controlled via the web.config file. The configuration items that are most likely to be changed are:

LDAP – can be used to lookup user information automatically

Log4net – specify a location for the log file on the server

Connection String – specify the connection string for connecting to the database

### LDAP

The application uses Windows Integrated Authentication (WIA) to identify the user. The use of LDAP is optional and can be used to gather additional information about the user, such as Name, Manager, Business Unit and Organization. Of course the information that can be collected is limited to what is available in the local Directory. If you chose to utilize LDAP to get additional user data, you will want to examine the LdapProcessing class that is located in the /Common folder.

To enable LDAP use, set the following key in the appSettings section of the web.config

<add key=”LDAPEnabled” value=”true” />

To disable LDAP use:  
 <add key=”LDAPEnabled” value=”false” />

If LDAP is enabled, the following keys are required:

* LDAPAdminGroup: Name of an LDAP group. System administrators should be members of this group. Eg. “OWASP-SAMM-Admins”
* LDAPBUOwnerGroup: Name of an LDAP group. Business Unit Owners should be members of this group. Eg. “OWASP-SAMM-BU-Owners”
* LDAPServerDNS: DNS name of LDAP server, including port number. Eg. “ldap.company.com:1234”
* LDAPUserPath: Path to people area of LDAP. Eg. “ou=people,o=company.com”
* LDAPGroupPath: Path to group area of LDAP. Eg. “ou=groups,o=company.com”

### Logging

The application uses log4net for logging. The web.config file contains a path for the log file on the server. Make sure the path is set to a valid location.

### Database Connection String

There is connection string information located in two places in the solution. The main location is in the web.config. This is used by the application at runtime.

The other connection string is in the app.config file located in the DAL project. This connection string is used by entity framework when updating the entity model.

Both connections should point to the location of the database.

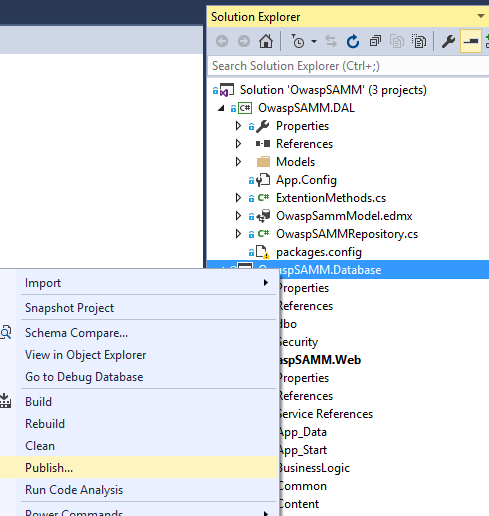
# Deploying the database

You should have a SQL Server instance running. The account you use for publishing the database will need to have “Sysadmin” permissions on the SQL Server instance in order to create the database.

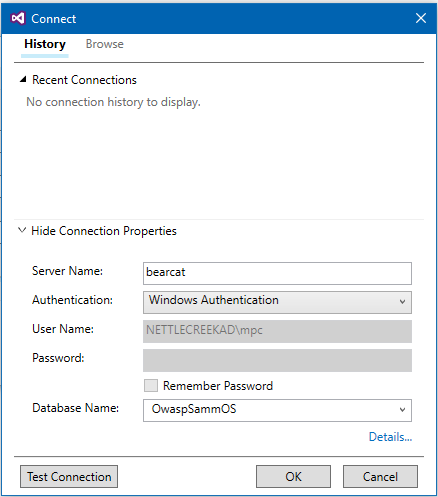
## Publish the Database

Steps for deploying the database.

* Open the solution
* In solution explorer, right click the database project, select Publish from the context menu



* Fill in the Publish Database form. The simplest way to fill in the form is to click the edit button and fill in the Connect dialog box.



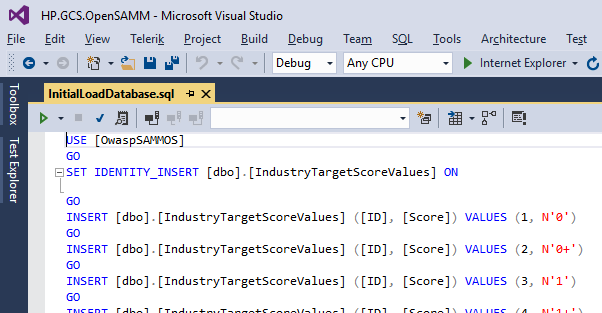
* Enter your server name. Enter the Database Name: **OwaspSammOS** and click **OK**. This will drop the appropriate connection string information into the Target database connection and Database Name boxes.
* Click the Publish button to publish the database to SQL Server.
* Check to see that the database has been created and contains tables.

## Load the database tables

The next step is to load the tables with data. There is a script in the database project that will load all the tables with the data needed to start using the application.

There are many ways to run the script. The following is a simple process for running the script directly from Visual Studio.

* Open the solution and expand the database project
* Locate the script file InitialLoadDatabase.sql which is located under /dbo/Scripts. Double-click the file to open it. The file will open and there will be a tool ribbon just above the file.



* Click the Connect icon to connect to a database. Fill out the Connect form with the same server and database name that was used above. If the Connect icon connects you to the database without prompting with the Connect dialog box, you can click the Change Connection icon to get the Connect dialog. 
* Click the Execute icon to run the script



* The script will run and populate the tables with data. You can check a few of the tables to ensure they contain data.

## Set Database Access

The service account used in the IIS site application pool needs access to the database and needs to be configured in SQL Server. It can be done with a script. If you have SQL Server Management Studio (SSMS), it’s fairly simple to configure.

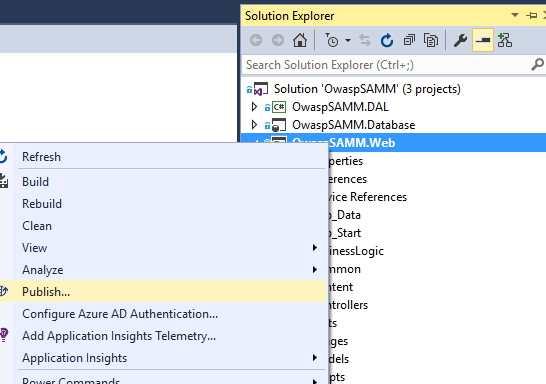
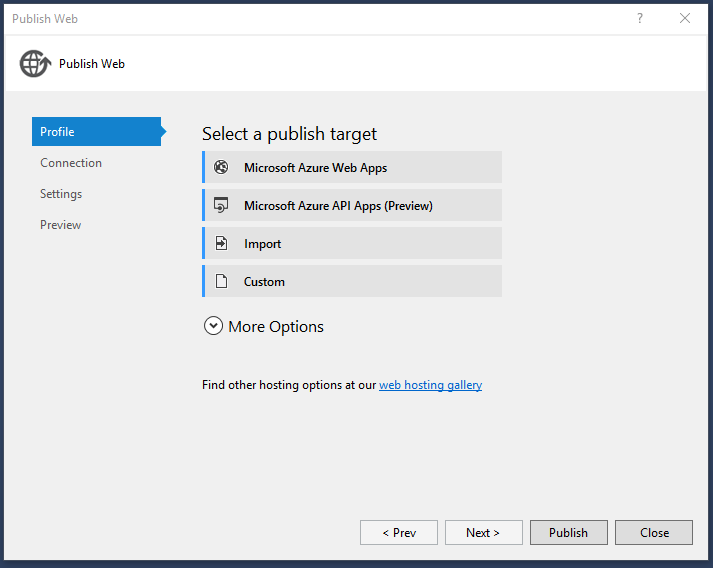
* Open SSMS and connect to the database.
* Expand the **Security** tree
* Right-click the **Logins** node and select **New Login**
* In the New Login dialog box, enter the name of the service account in the **Login Name** field. If it’s a domain account, remember to include the domain name.
* Select the **User Mapping** page
* Find the OwaspSammOS database in the list of databases, check the **Map** box in front of the name
* The lower pane contains the database roles for the selected database. Check the **db\_owner** role
* Click the **OK** button

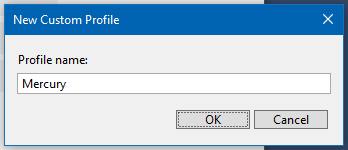
# Deploying the web site

There are many ways to build and deploy the web site. If you have a preferred method, go ahead and use it.

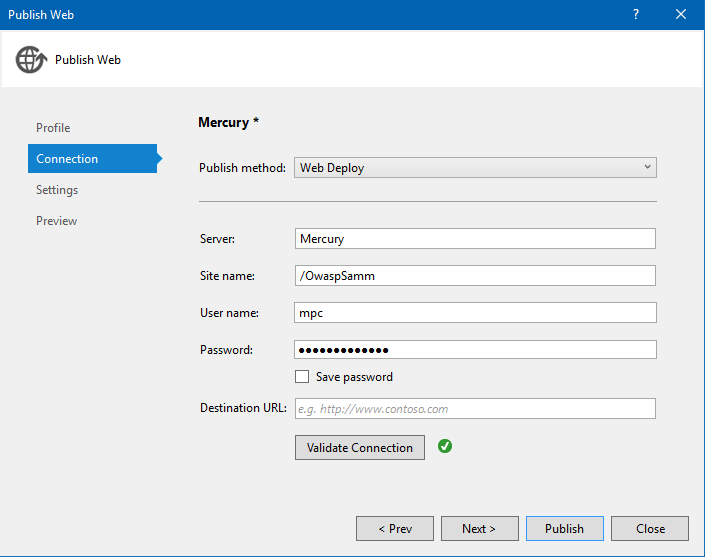
One way of building and deploying the site is to use the Publishing feature of Visual Studio. It requires you to have the Web Deployment Agent running on your server. The Web Deployment Agent is an optional component that is available in IIS. The Web Platform Installer is used to install it.

The following describes the steps to publish the web site using the publishing functionality of Visual Studio.

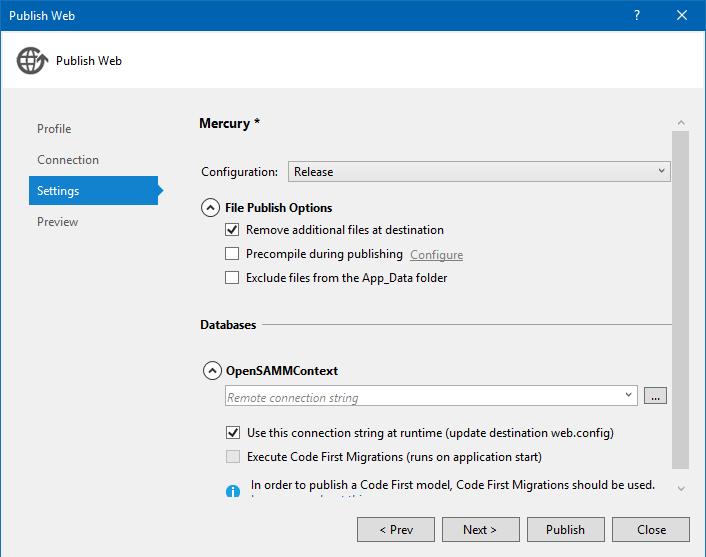
* Open the solution
* Right-click the web project (HP.GCS.OpenSAMM.MVC5) and select Publish from the context menu. 
* The Publish Web dialog box appears. The VS 2015 screen is displayed. VS 2013 has a slightly different appearance. Start by selecting **Custom**, enter a name for the profile and click **Next**. 



* Connection Page:



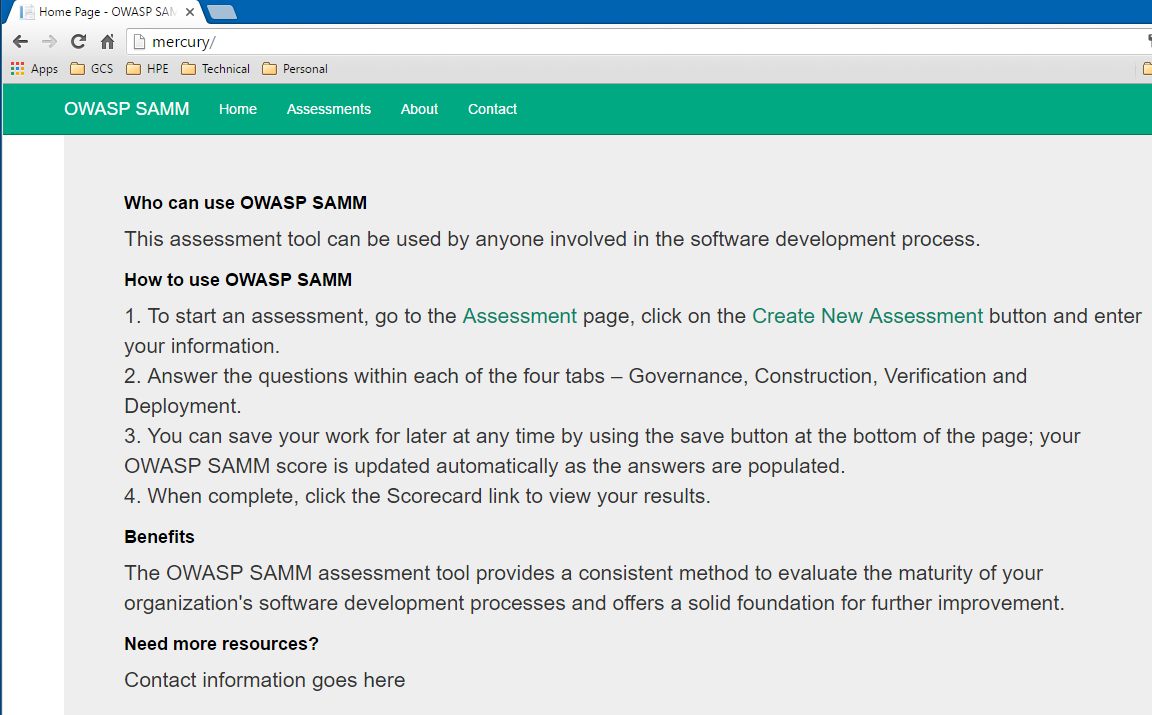
* Set the following values
  + Publish Method: Web Deploy
  + Server: The fully qualified name of your server
  + Site Name: Name of the site you created in IIS
  + User name/Password: These are the credentials for an account that has access to IIS
  + Destination URL: Optional – if specified, the URL will open in a browser upon completion of publish operation.
* Click the **Validate Connection** button. This will verify that VS can communicate with the server. If there is a problem, you’ll need to resolve it before continuing.
* Click **Next** for the **Settings** Page



* Settings
  + Configuration: Release – can be changed to Debug if needed
  + Check Remove additional files at destination
  + Configure the OwaspSAMMContext value by clicking the button with the ellipses. A database configuration dialog box will appear. Enter the database server name and database name and click OK. The appropriate connection string will be dropped into the box.
* The preview page is optional. At this point, you are ready to publish the site. Click the **Publish** button.
* Visual Studio will build the application and publish it to the IIS server

# Browse the site

Once the site has been published, you should be able to open the site in a browser. Open the browser, type the URL to the IIS server.



## Site Administrator

The site has a couple of functions that are accessible to “Administrators”. The designation of administrators is handled differently, depending on whether LDAP is enable or not.

If LDAP is enabled, the system will look at the LDAPAdminGroup application key in the web.config and check to see if the user is a member of that directory group.

If LDAP is not enabled, you will need to manually designate the first administrator. Once the administrator role is assigned, the administrator can manage user roles in the application. To assign the Administrator role to a user, use a tool like SSMS. Locate the user in the UserData table and set the Administrator field to 1.