**Debugging .NET Applications**

**Introduction**

Sometimes we want to be able to see how the target application handles data in real-time. For example, imagine we have identified a potential deserialization vulnerability, but the payload we are using doesn't work and we aren't sure why. By debugging the application, we can step through the relevant code line-by-line until we realize why the payload is not working.

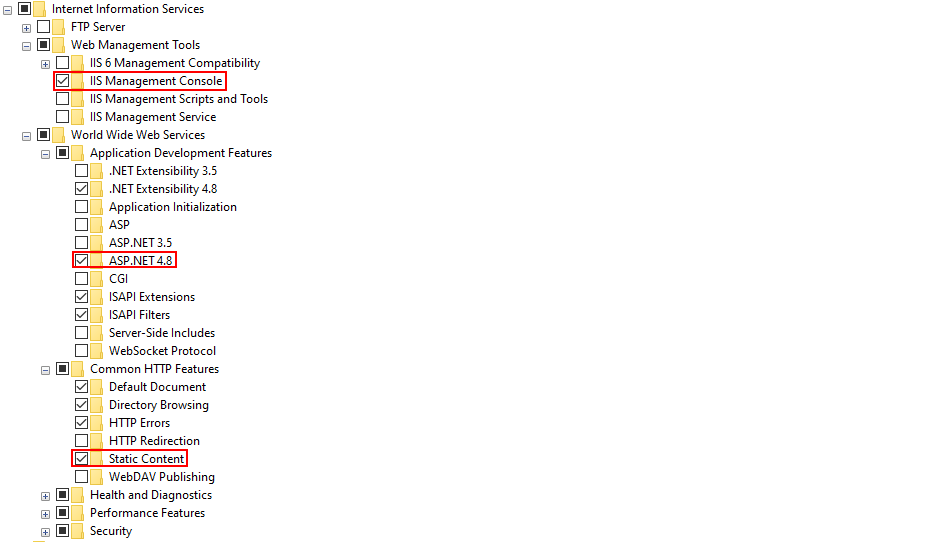
Typically, debugging requires the source code of an application. However, when it comes to .NET, we can use another open-source tool called [dnSpy](https://github.com/dnSpyEx/dnSpy) to do the same with decompiled code.

**Running TeeTrove**

**Installing Internet Information Services (IIS)**

The deployment files that we were provided for TeeTrove are not standalone, as we need another program to run the application. In this case, we will use IIS to serve the web application locally, so that we can debug it.

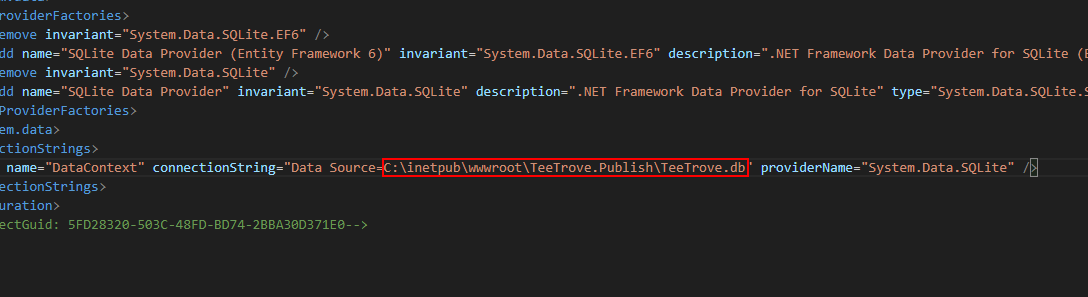
IIS comes by default with Windows, however, it may not be enabled by default on your installation. To enable IIS, open the Start Menu and search for Turn Windows Features on or off. Inside the window, we want to click on Internet Information Services. Next, expand the dropdown and ensure the following features are enabled, paying special attention to the ones highlighted in red:



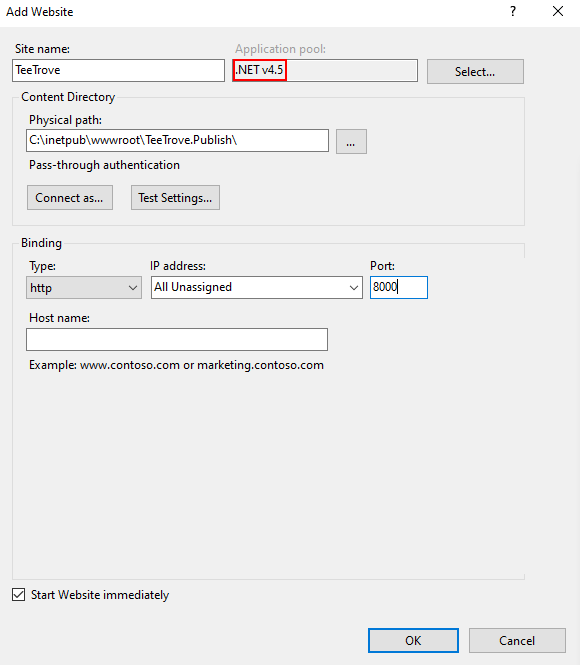
Once the appropriate options are checked, we can click OK and Windows will automatically download any missing files.

**Configuring IIS**

Before we can configure IIS, we need to make sure the supplied deployment files are extracted somewhere the server can access, like C:\inetpub\wwwroot. Next, we need to modify Web.config so that the application can access the database file correctly; open Web.config in the text editor of your choice, scroll to the bottom of the file, and update the value of Data Source to the full path of the TeeTrove.db file in the same folder.



Now we are ready to configure IIS. Open the Start Menu and search for Internet Information Services (IIS) Manager. Inside, right-click Sites and select Add Website. Fill out the popup window like shown below, and make sure that the Application Pool is set to .NET v4.5, otherwise, it will not serve the application correctly!

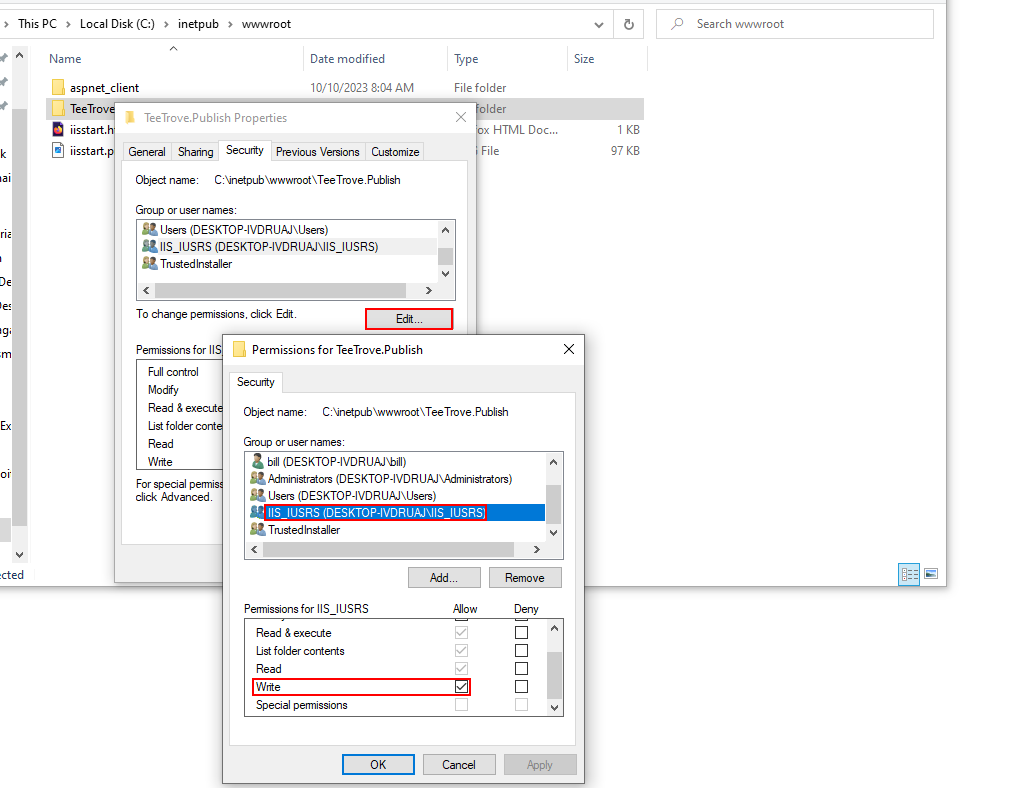


Hit OK and now TeeTrove should be accessible at <http://localhost:8000>.

A screenshot of a group of t-shirts

Description automatically generated

And now there is just one final step to make sure we can write to the database. Browse to the location where the deployment files are, right-click the folder, and modify the permissions so that the IIS\_IUSRS user has write permissions on the folder.



**Debugging TeeTrove**

**Preparing the DLL Files for Debugging**

Before we can get into debugging, we need to "prep" the files. By default, IIS makes debugging complicated by optimizing the assemblies. To prevent this from happening, we can use a PowerShell script to disable optimization.

Download the following [PowerShell Module](https://gist.github.com/richardszalay/59664cd302e66511618f51eaaa77db26), and run the following commands (replacing the last path with wherever you placed the application):

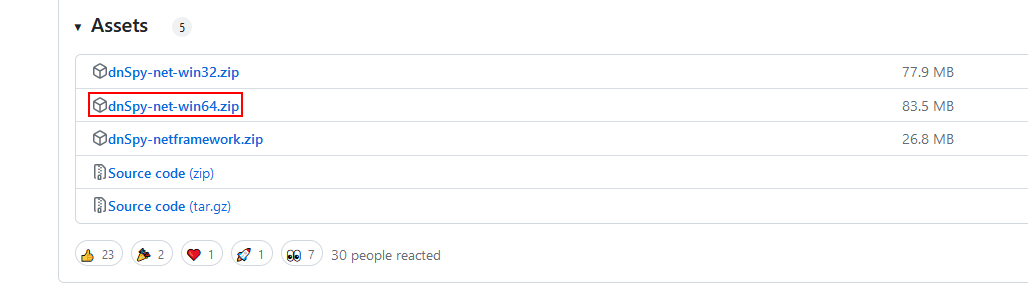
Debugging .NET Applications

PS C:\> Import-Module .\IISAssemblyDebugging.psm1

PS C:\> Enable-IISAssemblyDebugging C:\inetpub\wwwroot\TeeTrove.Publish\

**Installing dnSpy**

Now that we have TeeTrove running, and the application files are prepped for debugging, let's work on getting our debugging environment set up. For this, we will need to install [dnSpy](https://github.com/dnSpyEx/dnSpy). Head to the GitHub repository's Releases page, and then download the latest -win64.zip archive.

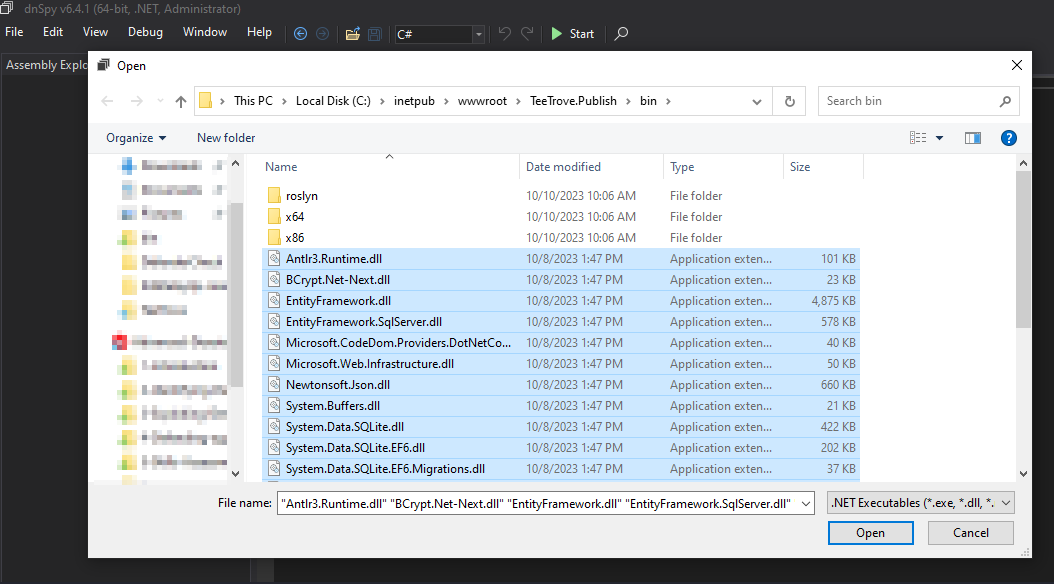


Once downloaded, simply extract the archive and the tool is ready to be used!

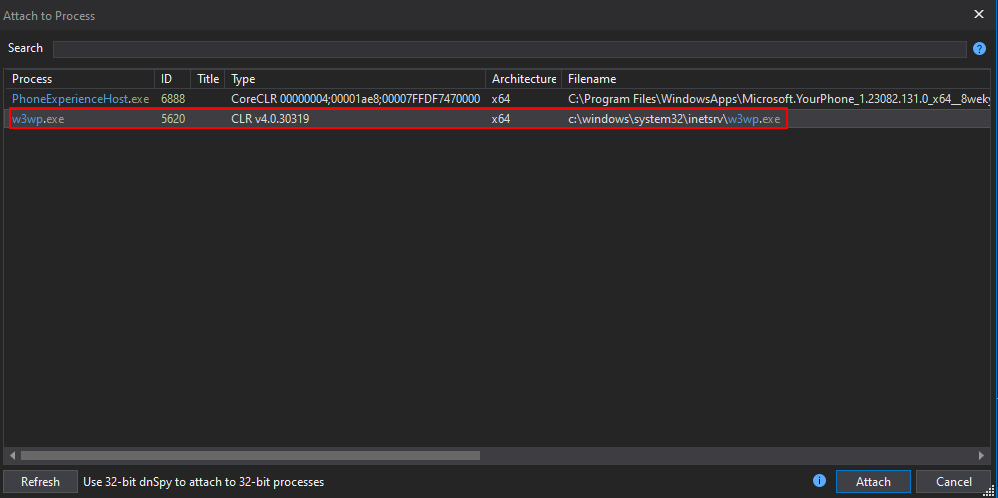
**Debugging TeeTrove with dnSpy**

Finally, open up dnSpy as Administrator. The layout will be similar to both dotPeek and ILSpy; there is an Assembly List on the left-hand side, and the main window pane is where decompiled code will be displayed.

From the File menu, select Open and select all the DLL files in the application folder.



Next, select Debug > Attach to Process and look for w3wp.exe. If it does not appear in the list, send any request to the web application and click Refresh, it should show up.



At this point, if everything was done correctly, debugging should be working. We can test this by opening TeeTrove.Controllers.AuthController and setting a breakpoint on line 18. We can try to load <http://localhost:8000/Auth/Login> in the browser, and the application should break, allowing us to step through lines and view the values of variables.

