# Setup Instructions

The best way to experience the vulnerable website is to load it directly into your browser from the web at <http://hackyourselffirst.troyhunt.com>

Only the vulnerable website is available online; the attacking website and the secure website are available as code only.

## Installing the vulnerable website

The vulnerable website is an ASP.NET 4.5 MVC 4 solution in the “before” folder. It can be run directly from Visual Studio 2012 but for best results in terms of replicating the behaviour of a real world website when being attacked, run it from a local instance of IIS.

There is a database that will be automatically generated by Entity Framework code first and pre-populated with data. There is a single connection string in the web.config that should be set to an instance of SQL Server and the account will need DBO rights in order to create the objects on first load.

The website also sends email on account creation and password resets so you’ll need to configure the SMTP host in the system.net element of the web.config.

**Warning:** This website is *specifically* designed to illustrate numerous poor security practices and in many places it has required very unusual coding patterns to achieve this. Don’t build real apps this way!

## Installing the attacking website

Identical to the vulnerable website but with no database dependency. For best results, add an entry to your local host file to point attacker.hackyourselffirst.troyhunt.com to 127.0.0.1 so that you can reference the same domain if you want to follow through the course.

## Installing the “secure” website

The secure website can be found in the “after” folder – except you won’t be able to demonstrate everything shown in the video with it. Many of the vulnerabilities I show have dependencies on cascading flaws and fixing earlier ones means that later ones will not be exploitable. Most of the secure patterns will be observable but there will be some that are not.

Give the secure website the same connection string as the vulnerable one as the database schema is identical.

**Warning:** This website is designed to demonstrate specific mitigations against the risks shown in the course. There will be others – it’s not meant to be a production-ready, “secure” site rather a site that demonstrates specific behaviours. As with the vulnerable site, the structure is designed to surface poor security practices in the browser which has meant doing some very odd things in code. By all means, refer to the specific security mitigations covered in the course but don’t assume this is a utopian example of a secure site – it’s not!