Lab 0: Lab Setup Guide

Date Due: September 14, 2021, by 11:59 PM EST.

This setup guide assumes you are running a Windows machine that can run Visual Studio 2019 Community Edition.

IMPORTANT: You can either follow the manual setup steps (but *no support* will be provided by the instructor) **OR** the provided Virtual Machine for the class.

Manual Setup

Please proceed to Phase 1 below if using the Virtual Machine. Please install the following specific software:

1. Visual Studio 2019 Community Edition (https://visualstudio.microsoft.com/vs/community/).
2. MySQL Server Community version 5.7.33 (https://downloads.mysql.com/archives/installer/).
3. MySQL Workbench Community 6.3.10 (https://downloads.mysql.com/archives/workbench/).
4. MySQL DotNet Connector 6.4.4 (https://downloads.mysql.com/archives/c-net/).
5. Git 2.32.0 or higher (https://git-scm.com/).
6. Burp Suite Community Edition 2021.8 (https://portswigger.net/burp/communitydownload).
7. Firefox version 32.0.3 (http://ftp.mozilla.org/pub/firefox/releases/).
8. (Optional) Sourcetree (GIT GUI at https://www.sourcetreeapp.com/) or other Git based GUI.

Go to https://code.umd.edu/users/sign_in

Using Git, clone the following repository to your local machine: https://code.umd.edu/gkini/webgoat.git

Import the project into Visual Studio 2019 Community Edition.

You will find that a lot of the references in the .NET project are showing yellow exclamation mark triangle icons indicating it cannot find the assemblies referenced by the project. You will have to manually add the MySQL DotNet connector assemblies (installed above) in Visual Studio References for the project.

You also may need to manually run the SQL scripts for the WebGoat.NET application to get the database setup along with username and password for the admin account. Once you have the WebGoat.NET application running, continue on to Phase 1 below.

Phase 1: Setup the lab environment

1. The virtual machine is configured to run with 2 cores, 8 GB RAM, and 128 GB hard disk with Windows 10 Enterprise Edition installed and provided by UMD IT. It has no password set for the 'Student' user account, an administrative account. You may wish to change:
    a. The key to use your UMD provided Windows 10 Student edition key or a newly purchased Windows 10 key.
    b. The password for the Student account so that it is secured.

2. Ensure you can open Visual Studio 2019 Community Edition. You may need an email sign in to continue using it for free.
3. Ensure you can run the WebGoat.NET application after importing it. On the virtual machine, this is already done for you. You can directly open the WebGoat.NET application in Visual Studio 2019 Community Edition.
4. Run the following commands in GIT to change the remote to point to your own student account on https://code.umd.edu/<UMD CAS ID>/WebGoat.NET:
    a. First login to https://code.umd.edu using CAS authentication
    b. Create a blank **Private** project called 'WebGoat' without the single quotes. It is important you keep the same exact name and case for grading purposes. Otherwise you risk not getting a good grade.
    c. Right-click the folder where you cloned the WebGoat folder and choose **Git Bash here**. On the virtual machine this is located at: C:\Users\student\Workspace\WebGoat.
    d. This will open a terminal prompt that allows you to run Git commands.
    e. Running the following command will show you the remote repository that you will push to and pull from to save your work or restore from respectively:
        i. `git remote -v`
    f. If there are no remotes setup, it will not show anything.
    g. Now setup Git global configuration options and replace the text in double quotes appropriately:
        i. `git config --global user.name "Your Full Name"`
        ii. `git config --global user.email "Your UMD Email address"`
        iii. `git init --initial-branch=main`
    h. The last command above uses the branch name "main" to refer to the initial and main branch. Development in Git is usually performed using "feature" branches that branch off of the main branch. To add a feature, you branch from the main branch creating a "feature" branch, develop and test your feature in that branch then merge back into the "main" branch when its ready for merging. Please read https://nvie.com/posts/a-successful-git-branching-model/ for more information.
    i. Now you want to create a secure SSH key to associate with your account:
        i. `ssh-keygen -t ed25519 -C "Your UMD email address"`
        ii. Accept the defaults and provide a passphrase to protect your private key file.
    j. Now to use the key, you have to login to https://code.umd.edu , go to your Profile Settings, SSH Keys and add the id_ed25519.pub file contents as a key. For more information see https://docs.gitlab.com/ee/ssh/. **IMPORTANT:** Don't copy your id_ed25519 (private key) file contents, rather the **.pub** extension file (public key) file contents:
        i. `cat ~/.ssh/id_ed25519.pub`
        ii. Copy the output to a SSH Key entry in your profile on https://code.umd.edu .
    k. Now to use the Git Bash command line to do all your git commands, you can enter the following commands before you start doing git commands. For Git GUI, refer to its documentation on how to use the SSH key for authentication with the remote repository.
        i. eval `ssh-agent -s`
        ii. ssh-add ~/.ssh/id_ed25519

iii. Enter your passphrase for the private key file

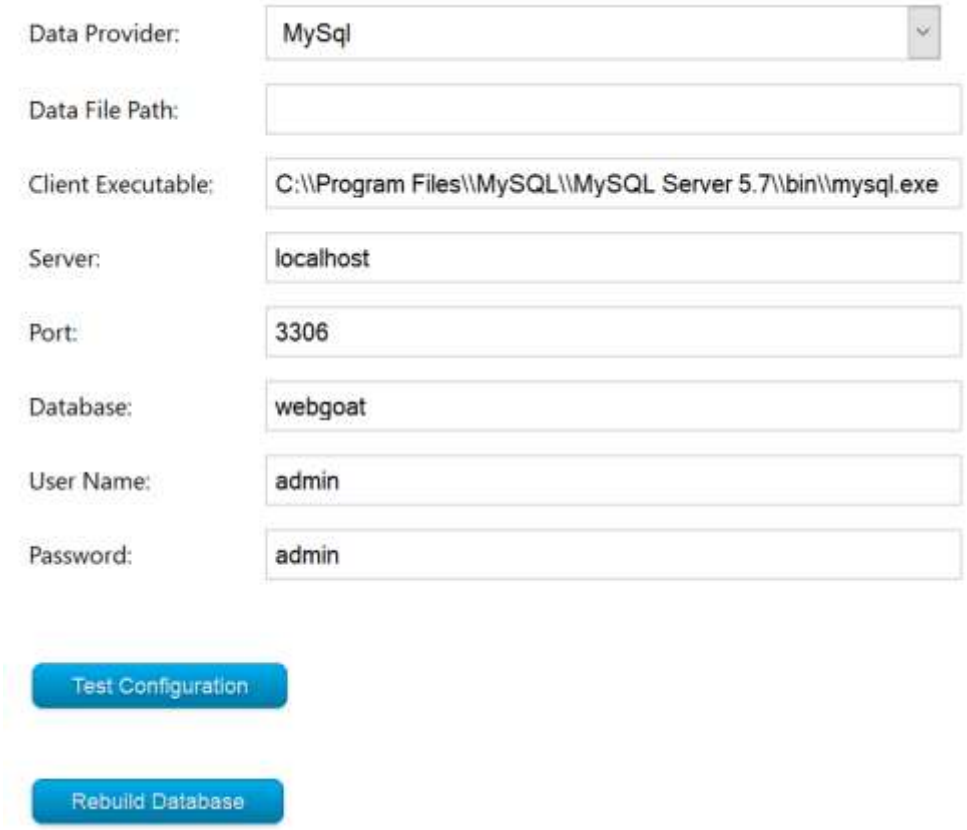
l. You now want to add your newly created project on https://code.umd.edu as the remote by running the following command:

```
i.   git remote add origin git@code.umd.edu:<Your UMD
     ID>/webgoat.git
ii.  git add .
iii. git commit -m "Initial commit"
iv.  git push -u origin main
```

m. Now running the command from 4e above shows the new remote repository. This remote repository will be used to perform auto grading of code submissions. You will use git commands to push answers to various lab exercises under different branches to this repository and submit for auto grading on ELMS.

Phase 2: Run the WebGoat.NET application and utilities

1. Now run the Burp Suite Community Edition proxy, without which the web application will not run in the configured Firefox browser. For the community edition you can only create a temporary project and load project settings each time you start it. After starting a temporary project, please choose:
    a. Project > Project Options > Load Project Options
    b. Go to C:\Users\student folder and choose the **burp_suite_project_settings.json** file.
    c. Then go to Proxy tab in the interface below and click the 'Intercept is On' button. It should show 'Intercept is off' in the text.
2. Run the WebGoat.NET application by pressing the Green Play button at the top in Visual Studio 2019 Community Edition with the text "IIS Express (Firefox)" in the virtual machine.
3. After running the WebGoat.NET application in Visual Studio 2019 Community Edition, you will see the WebGoat.NET Application default home page showing 'Welcome to WebGoat.NET' screen. Now click the 'Setup Database' button.
4. Now you will see a screen like the following:



5. The above screen values should work for the Virtual Machine which is already setup with the pre-requisite Database and settings.
6. The username and password for the root user for MySQL (when using MySQL Workbench application) to administer the WebGoat database is:
   Username: root
   Password: student
7. The username and password for the admin user for MySQL WebGoat Database is:
   Username: admin
   Password: admin
8. Please submit the following in a writeup:
   a. Screenshot of the running WebGoat.NET application Welcome Page in the browser.
   b. Screenshot after running Rebuild the Database button.
9. **REMEMBER: If any of the attacks you perform causes the application to enter a bad state, just go to the Rebuild Database link at the top right and click Rebuild Database button.**