WebGoat 8: A deliberately insecure Web Application

Introduction

WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons.

This program is a demonstration of common server-side application flaws. The exercises are intended to be used by people to learn about application security and penetration testing techniques.

WARNING 1: *While running this program your machine will be extremely vulnerable to attack. You should disconnect from the Internet while using this program*. WebGoat's default configuration binds to localhost to minimize the exposure.

WARNING 2: *This program is for educational purposes only. If you attempt these techniques without authorization, you are very likely to get caught. If you are caught engaging in unauthorized hacking, most companies will fire you. Claiming that you were doing security research will not work as that is the first thing that all hackers claim.*

Installation Instructions:

1. Run using Docker

Every release is also published on DockerHub.

Using docker run

The easiest way to start WebGoat as a Docker container is to use the all-in-one docker container. This is a docker image that has WebGoat and WebWolf running inside.

```
docker run -p 8080:8080 -p 9090:9090 -e TZ=Europe/Amsterdam
webgoat/goatandwolf
```

WebGoat will be located at: http://127.0.0.1:8080/WebGoat WebWolf will be located at: http://127.0.0.1:9090/WebWolf

Important: Choose the correct timezone, so that the docker container and your host are in the same timezone. As it important for the validity of JWT tokens used in certain exercises.

Using docker stack deploy

Another way to deply WebGoat and WebWolf in a more advanced way is to use a compose-file in a docker stack deploy. You can define which containers should run in which combinations and define all of this in a yaml file. An example of such a file is: [goat-with-reverseproxy.yaml]()

This sets up an nginx webserver as reverse proxy to WebGoat and WebWolf. You can change the timezone by adjusting the value in the yaml file.

```
docker stack init
docker stack deploy --compose-file goat-with-reverseproxy.yaml webgoatdemo
```

Add the following entries in your local hosts file:

```
127.0.0.1 www.webgoat.local www.webwolf.localhost
```

You can use the overall start page: http://www.webgoat.local or:

WebGoat will be located at: http://www.webgoat.local/WebGoat

WebWolf will be located at: http://www.webwolf.local/WebWolf

Important: the current directory on your host will be mapped into the container for keeping state.

2. Standalone

Download the latest WebGoat and WebWolf release from https://github.com/WebGoat/WebGoat/releases

```
java -jar webgoat-server-8.1.0.jar [--server.port=8080] [--server.address=localhost]
java -jar webwolf-8.1.0.jar [--server.port=9090] [--server.address=localhost]
```

The latest version of WebGoat needs Java 11 or above. By default WebGoat and WebWolf start on port 8080 and 9090 with `--server.port` you can specify a different port. With `server.address` you can bind it to a different address (default localhost)

3. Run from the sources

Prerequisites:

- Java 11
- Maven > 3.2.1
- Your favorite IDE
- Git, or Git support in your IDE

Open a command shell/window:

```
git clone git@github.com:WebGoat/WebGoat.git
```

Now let's start by compiling the project.

```
cd WebGoat
git checkout <<branch_name>>
mvn clean install
```

Now we are ready to run the project. WebGoat 8.x is using Spring-Boot.

```
mvn -pl webgoat-server spring-boot:run
```

... you should be running webgoat on localhost:8080/WebGoat momentarily

To change IP address add the following variable to WebGoat/webgoat-container/src/main/resources/application.properties file

```
server.address=x.x.x.x
```