

System Security Group Project #2

(10% of class grade, due 2/13 at midnight)

Go through interactive broken web application tutorials. Explain what you learn! This project is worth 10% of your class grade.

What you will need:

You will need access to the class vSphere environment. You will be using the following virtual machines:

- OWASP-BWA
- Kali Linux

Directions:

Setup

Follow the project videos for accessing and running the OWASP-BWA and the Kali Linux virtual machines.

Start the OWASP-BWA image (if not already started) and log in with the given username/password combination. Open the Firefox browser in Kali Linux and navigate to the IP address given.

Select the OWASP WebGoat Link and log in with user/user to get going. (However, you should note the other user/pass combinations.)

!!! This VM has many serious security issues. We strongly recommend that you run it only on the "host only" or "NAT" network in the virtual machine settings !!!

TRAINING APPLICATIONS	
OWASP WebGoat Version: 5.4+SVN Language: Java User Credentials (username/password): guest/guest User Credentials (username/password): user/user User Credentials (username/password): basic/basic Admin Credentials (username/password): webgoat/webgoat Link: home page	OWASP WebGoat.NET
OWASP ESAPI Java SwingSet Interactive	OWASP Mutillidae II
OWASP RailsGoat	OWASP Bricks

“WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons. In each lesson, users must demonstrate their understanding of a security issue by exploiting a real vulnerability in the WebGoat applications. For example, in one of the lessons the user must use SQL injection to steal fake credit card numbers. The application aims to provide a realistic teaching environment, providing users with hints and code to further explain the lesson.” – WebGoat wiki at https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project

On the left, click on the Introduction link. It will expand. Click on and read “How to work with WebGoat” and “Useful Tools”. You can skip the part on Tomcat Configuration since it will already be set up for you. Be sure you are familiar with the basics of webscarab and Firefox dev tools (discussed in the project videos).

Lessons in WebGoat

Start with these first two lessons:

- General (Yes, you should also do the HTTP Splitting sub-lesson)
- Insecure Storage

[For groups of 3]: Then complete (as far as possible) at least 9 lessons out of the following lessons.

[For groups of 4]: Then complete (as far as possible) at least 12 lessons out of the following lessons.

- Access Control Flaws
- AJAX Security
- Authentication Flaws
- Buffer Overflows
- Code Quality
- Concurrency
- Cross-Site Scripting (XSS)
- Improper Error Handling
- Injection Flaws
- Denial of Service
- Insecure Communication
- Insecure Configuration
- Malicious Execution
- Parameter Tampering
- Session Management Flaws
- Web Services

The deliverable

I am expecting 2 (optionally 3) parts to the deliverable.

1. **Paragraphs:** For each *page of each module you complete*, type up at least one paragraph on what you learned. Don't just copy and paste text from the lesson plans (I will be checking). In your paragraph, let me know what was easy, what was hard, and what you couldn't get to work quite right. We will use class periods as work periods to ask questions about things you can't quite get and get advice from other teams. (All advice from other teams should get a citation/shout out in your paragraph.)
2. **Score cards:** In addition, go to Admin Functions -> Report Card and submit your score card(s) to me. Each group member should be participating. For example, if you split the project up into three equal pieces, each member should submit a score card. If you all worked together on all the activities, just submit one score card. If you worked together on some things *and*

individually other things, you will need to make clear who did what and submit all the score cards. Some modules contain much more work than other modules, so keep that in mind when breaking up your project.

Here is an example of the top of a mostly-empty score card:

 Concurrency
 Cross-Site Scripting (XSS)
 Improper Error Handling
 Injection Flaws
 Denial of Service
 Insecure Communication
 Insecure Configuration
 Insecure Storage
 Malicious Execution
 Parameter Tampering
 Session Management Flaws
 Web Services
 Admin Functions
[Report Card](#)
 Challenge

Results for: user

Lesson	Complete	Visits	Hints
Normal user lessons			
How to work with WebGoat	Y	419	0
Tomcat Configuration	N	0	0
Useful Tools	Y	0	0
How to create a Lesson	Y	0	0
Http Basics	Y	43	2
HTTP Splitting	N	6	2
Using an Access Control Matrix	N	0	0
Bypass a Path Based Access Control Scheme	N	0	0
LAB: Role Based Access Control	N	0	0
Remote Admin Access	N	0	0
Same Origin Policy Protection	N	0	0
LAB: DOM-Based cross-site scripting	N	0	0
LAB: Client Side Filtering	N	0	0
DOM Injection	N	0	0

3. **[Optional] How could this project be made better in the future?** If you have some ideas on how to make this assignment better or anything more helpful in the future, let me know!

Your deliverable should be in your team google folder. Create a new google doc called "WebGoat" directly under your main team folder and place your paragraphs and score cards in there.

In addition, you will be required to fill out an internal peer review before you receive a grade. (You cannot skip this part.)

Other Things

- I do not have all the answers. (I know, I look like I do...)
- I'm expecting things to sometimes be difficult, even with the hints and solution videos available. Sometimes the difficulty will be in getting things set up. Other times it might be in figuring out what you should be watching or modifying.
- If you don't know what something is, Google it to find out!
- I expect you to have to use Google to figure out features of webscarab or Firefox dev tools.
- I know other walkthroughs exist on the Internet. If you use another source, cite it!
- Make sure you are taking the time to learn. If your paragraphs don't convince me you really understand what's going on, I will award less points.