האקדמית כנרת
הנדסה, חברה ורוח

Directions

A. Due Date: 16 June 2025 at 8:00am

B. The assignment must be done by individual students.

C. The assignment includes several submission elements, including an in-class presentation. The in-class presentation is required in order to receive a grade for the assignment. Students who do not perform the in class presentation on the days listed below will not receive a grade.

D. The penetration and hacking report must be submitted in PDF or DOCX format via Moodle using the provided the format file.

E. The presentation file must be submitted in PDF or PPTX format via Moodle.

F. The quiz produced must use automatic grading via an external quiz platform that records names, answers, and quiz results.

Web App Penetration and Hacking

In this assignment you will learn about web app penetration and hacking using the Open Worldwide Application Security Project (OWASP) (`https://owasp.org/`) web security training app Juice Shop (JS). Juice Shop is a fully functional JavaScript-based web app that presents an e-commerce storefront. The app allows you to view the catalog of items in the shop and place orders. The app has a login interface and ordering interface, but does not actually have a store behind it, so if you "place an order" you will not actually receive juice in the mail. As a matter of security and privacy, **do not** enter any real private information into the app.

The app is designed to be used by students of penetration testing and web app security, so it has intentional bugs and security flaws baked in. To make the app more interesting, the app is gamified, making the hacking and penetration steps into a challenge with stars (points) you earn along the way. The app is smart enough to detect when you've hacked parts of it, so it will tell you when you succeeded in one of its defined attack "games" by showing a pop-up success message (see examples in Figure 5).

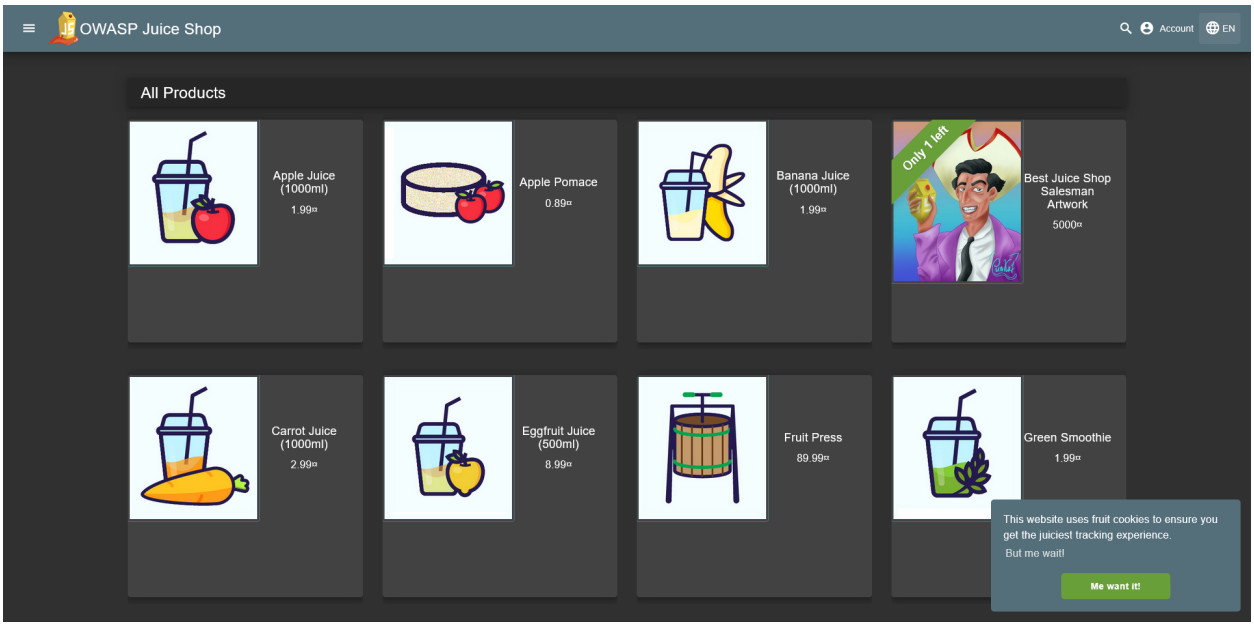

Figure 1: Screen shot of the Juice Shop app

In this unit, you'll attack the Juice Shop app, working out how to achieve security and data breaches. You'll need to earn 12 stars by hacking the Juice Shop, write up a report on how you did the hacks, present two of them to the whole class, and create a quiz to test the others students' understanding of the hacks.

1 About Juice Shop

Juice Shop is a fully functional web app designed to teach penetration testing and hacking. It's free, fully open source, written in JavaScript, and has built-in functions to detect when your hacks have worked. You can read more about the Juice Shop at the following links.

- About the Juice Shop Project `https://owasp.org/www-project-juice-shop/`
- The Hacking Instructor with step-by-step guides to some of the hacks on the shop: `https://pwning.owasp-juice.shop/companion-guide/latest/part1/challenges.html#_hacking_instructor`
- General Guide to Hacking Juice Shop: `https://pwning.owasp-juice.shop/companion-guide/latest/index.html` (cover is in Figure 2)

Figure 2: Cover of Pwning Juice Shop book

- Cloud-based Demo of the Juice Shop: `https://juice-shop.herokuapp.com/#/`

The app is **self-healing**, so every time you start it up it wipes its database and gives you a clean slate. That means that if you broke something by accident, restarting will help. That also means that restarting will make you lose your point progress. Therefore, you must save your progress manually every so often to protect your progress and show in your hacking report submission.

You can read more about these issues at the following links:

- About self-healing: `https://pwning.owasp-juice.shop/companion-guide/latest/part1/running.html#_self_healing_feature`
- How to save your progress: `https://pwning.owasp-juice.shop/companion-guide/latest/part1/challenges.html#_manual_progress_and_settings_backup`

2 Installing Juice Shop

Juice Shop app can be run in the cloud at the link above, but that's **not** where you'll be running it for this assignment. There are a few reasons for that. First, to ensure that the results are your own (everybody using the cloud version sees everyone else's progress). Second, some hacks are disabled on the cloud version for security reasons. Therefore, you'll need to install your own local copy of Juice Shop on your computer and attack it.

There are a few ways to install Juice Shop. Links to the installation guides follow.

- Docker installation (**recommended**): `https://pwning.owasp-juice.shop/companion-guide/latest/part1/running.html#_docker_image`

  Note that if you're running Docker on Linux it from within VirtualBox, you must open a port to forward Windows unless you want to use the browser inside VirtualBox.

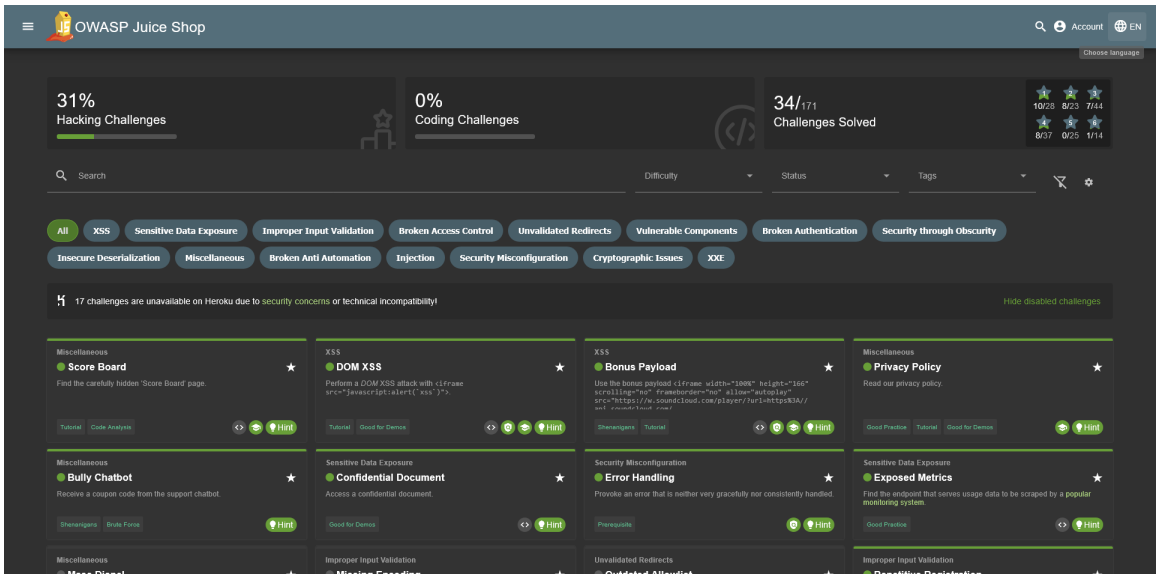

Figure 3: Juice Shop score board

- Local installation (requires Node.js): `https://pwning.owasp-juice.shop/companion-guide/latest/part1/running.html#_local_installation`

3 Performing the Hacks

For this assignment you'll need to earn 12 stars in the Juice Shop hacking chart. You can track your performance and achievements on the Score Board of the app (Figure 3). You'll notice that the score board itself isn't linked from anywhere on the app's page. You'll need to find it yourself. It's not a hard challenge to find, but it's a first step along the way.

3.1 Tutorials

Some of the hacking challenges have step-by-step tutorials associated with them. For instance, in Figure 4 you can see a selection of 3 star challenges, including Forged Feedback, Login Jim, Login Bender, API-only XSS, Admin Registration, and Bjoern's Favorite Pet. Three of the challenges (Forged Feedback, Login Jim, Login Bender) have the Tutorial label on them in the lower left hand corner and a green hat icon that takes you through a step-by-step how-to guide for the hack. You are certainly encouraged to try out the hacks shown and learn how they work, but the hacks with tutorials will not count toward your required star count.

3.2 Success Messages

You'll know that a hack was successful when you see a pop-up banner about the challenge as in Figure 5. The pop-up will stay there until you close it with an X click.

3.3 Hacking Requirements

You must hack challenges to give you 12 stars. The hacks must consist of the following:

1. (At least) one ⋆⋆⋆ (3 star) attack that doesn't have a tutorial provided (There are 21 of them)
2. (At least) one ⋆⋆⋆⋆ (4 star) attack that doesn't have a tutorial provided (There are 25 of them)
3. Other challenges without tutorials that give you at least 5 more stars. You can do a few easy ones or one hard one. It's up to you.

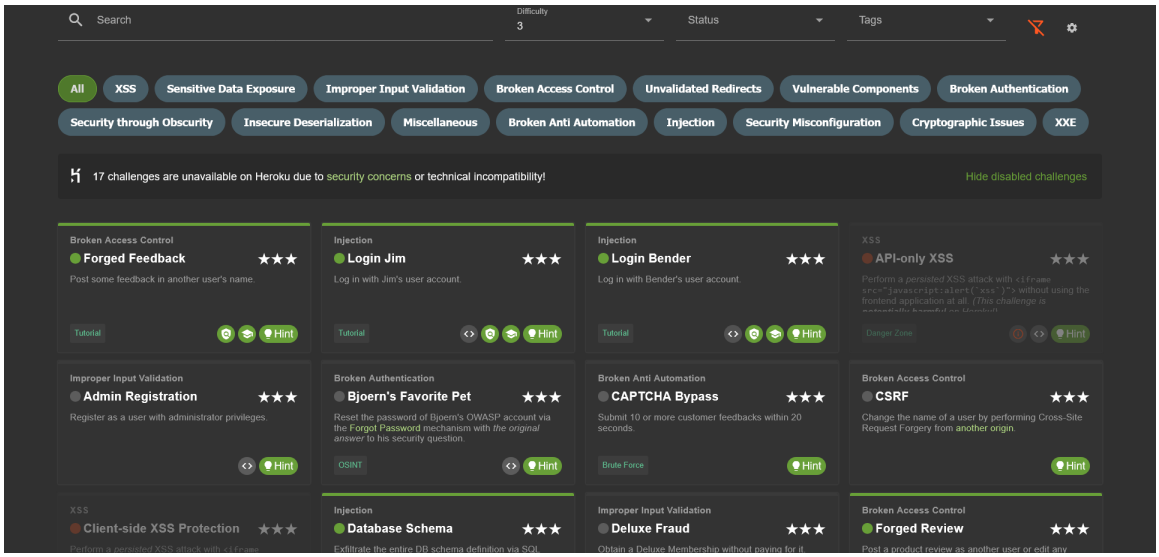

Figure 4: Challenges with and without tutorials

Each student will present 2 hacks during a live demo (the 2 with the highest star ratings done), so you'll need to register which hacks you plan on presenting to the class on the Moodle web assignment.

4 Hacking Report

You will write a hacking report about your work to document how you did the hacks. The report must follow the format in the report template found on Moodle. Each successful hack must be put in its own chapter. If there are not enough chapters in the report, you can add some more of your own. Keep in mind that the challenges you solved must be carefully documented so that they can be reproduced by the reader based on your report alone.

5 In Class Presentation

Each student will present their hacking explanation demo over the course of 4 two-hour class sessions, with each session containing 4 presentations. If you know you can not make it to class on a particular day, do not sign up for a lab that must be presented that day!

Based on the schedule, the presentation dates will be: 16 June, 22 June, 23 June, 29 June.

You will present your two highest star hacks to the whole class. You will teach the class about what you learned from doing the hacks, sharing your knowledge with everyone. Your presentation should be planned for 20 minutes and include the following elements:

1. A 10 minute explanation and demo of the hack you performed with the **highest** star value.
   - What is the goal of the hack?
   - How do you know if the hack succeeded?
   - What part of the app are you attacking?
   - What support tools do you need to do the attack?

   Then perform a step-by-step walk through of the hack in front of the class.
2. A 10 minute explanation and demo of the hack you performed with the **second-highest** star value.

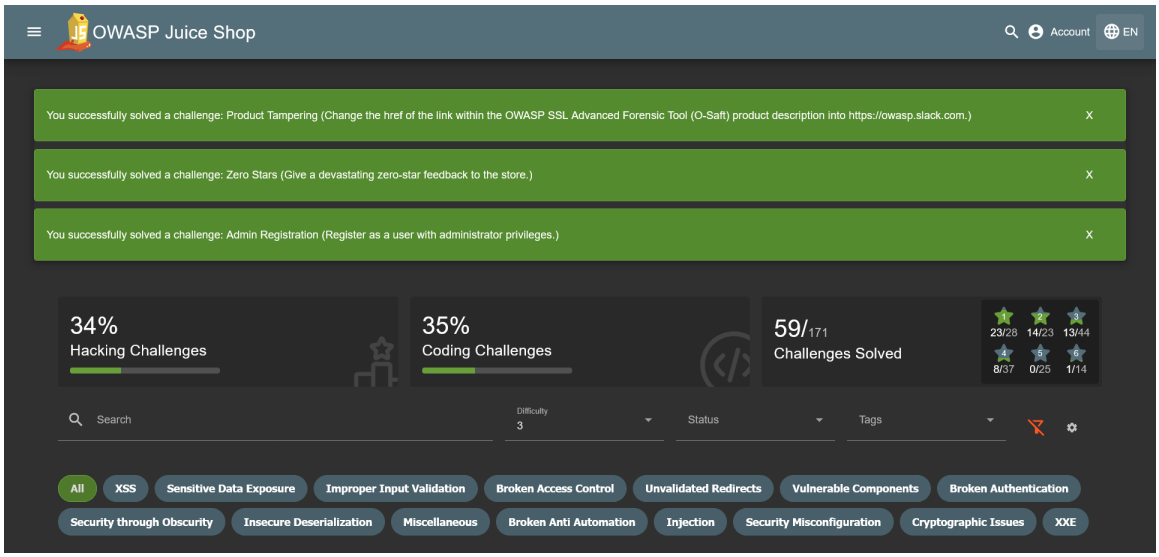

Figure 5: Hack success pop-up messages.

- What is the goal of the hack?
- How do you know if the hack succeeded?
- What part of the app are you attacking?
- What support tools do you need to do the attack?

Then perform a step-by-step walk through of the hack in front of the class.

Create a slide presentation file (PPTX or PDF) to accompany your presentation. Ensure that it's clear and visually engaging.

6 In-Class Quiz

To test how well the class understood your presentation and demonstration, you will create a quiz to give to everyone. The quiz must be multiple choice and run on a platform that enables you to ask questions, automatically grade the responses, and record who answered what. Potential platforms include Google Forms and Quizlet. Try out the quiz tool before you actually give the quiz to ensure it meets the requirements.

The quiz must include 3-5 multiple choice questions that check for understanding of the attacks you presented. The quiz questions must be clearly written and be challenging. They should force the students to recall your presentation and the results you presented.

The questions may be in English or Hebrew, but must be clearly written and presented. Use spell check and grammar check on all questions and answers.

The quiz will take place after your presentation. The quiz will be 10 minutes long at most, so ensure the questions can be answered within that time frame.

7 What to turn in

Submit the following elements via Moodle.

1. Presentation Slide Deck: Turn in your slide deck in PDF or PPTX format.
2. Hacking report: Turn in your lab report in DOCX or PDF format.

3. Quiz grades: Turn in the grades from the quiz along with the answers from each student. Use the quiz's export feature to submit the grades in Excel or CSV format.

8 Grading Rubric

8.1 Hacking Results Report (45%)

Completeness: 20% : All steps are documented with clear screenshots and descriptions

Analysis: 15% : Insightful analysis of the results and understanding of the setup or attack mechanisms

Clarity: 10% : Well-organized report with clear, concise language

8.2 Class Presentation (40%)

Content: 20% : Accurate and thorough background information, clear demonstration of hacks

Delivery: 10% : Clear and confident speaking, good pacing, and engagement with the audience

Visual Aids: 10% : Slide deck is well-designed with relevant visuals and easy-to-follow flow

8.3 In-Class Quiz (15%)

Relevance: 5% : Questions are directly related to the presentation content

Difficulty: 5% : Questions are challenging but fair, reflecting an understanding of the hacks

Clarity: 5% : Questions are clearly worded without ambiguity

