TopDog Hacking Challenge: A Good Offense is the Best Defense

TSIT01, TSIT02 Computer Security

Jonathan Jogenfors
Niklas Johansson
Guilherme Xavier
Information Coding Group
Department of Electrical Engineering, Linköping University

December 3, 2019

Contents

Chapter 1

Introduction

Have you ever taken a computer security course and wanted to learn more? Tired of just listening to the lecturer going on about hacking computers while you dream about actually breaking into stuff? Now is your chance. In this lab course you will be taking on the role as *penetration tester*, or *pentester* for short. This means you'll be shown a selection of vulnerable web applications with the goal of breaking into them and/or make the application perform tasks that it was not designed for.

1.1 Overview

In the LiU TopDog Hacking Challenge you will practice penetration testing. Using a set of increasingly difficult assignments, you will gradually learn the basics of how an adversary might exploit badly designed applications and security systems. The goal is to give you the basics in practical security work and understand some common pitfalls when developing web applications. After the lab you should be well-equipped to avoid these security issues whenever you develop your own web application.

1.2 Lab Organization

This lab will run from the starting date until it closes. The lab system is publicly available and you can work on the assignments in your own time on the lab computers or your personal laptops. As the server is reachable from the Internet, you can also work from home if you so choose. The progress will be stored on the server so you can come back at any time.

There are scheduled sessions where the assistant will be available at his or her office to provide assistance. Plan carefully, because time will be limited for each student. Assistance will be provided at a first come, first served basis. Think drop-in, so no booking is required.

For other questions please see section 1.6.

1.3 Deadlines

The lab starts on Friday, the 22^{nd} of November 2019 at 18:00. The lab must be finished before 17:00 on Friday, the 17^{th} of January 2020. At this time, the assignments will be

disabled an no more progress can be done. If you haven't finished the lab by this date you will have to re-take the lab next year.

There is also another, soft deadline. At 17:00 on Monday the 9^{th} of December 2019 the competitive part of the lab ends and the scoreboard will lock. No more points or medals will be awarded at this point. The next day, just before the guest lecture, there will be a small ceremony for the winners.

1.4 Disciplinary Stuff

Each individual student is expected to perform the lab in order to pass. However, you are allowed (and encouraged!) to cooperate to a reasonable degree. In addition, you are expected to understand and follow the university-wide rules for disciplinary matters, and like any other examination you are not allowed to cheat or disrupt examinations.

1.5 Ethics

This lab and what you learn is for educational purposes only. Do not attempt to use these techniques without authorization. If you are caught engaging in unauthorized hacking, most companies will take legal action. **Claiming that you were doing security research will not protect you.**

1.6 Contact Information

To get in touch with the lab assistant, please send e-mail to the address below corresponding to your course.

1.6.1 TSIT01 Datasäkerhetsmetoder

Course homepage: http://www.icg.isy.liu.se/courses/tsit01/

Lab e-mail: tsit01-lab@isy.liu.se

1.6.2 TSIT02 Computer Security

Course homepage: http://www.icg.isy.liu.se/courses/tsit02/

Lab e-mail: tsit02-lab@isy.liu.se

Chapter 2

Preparing for the lab

Begin by reading through the entirety of these lab instructions. Also note that we are continuously improving these instructions, so be sure to regularly check our gitlab repository for the latest version.

2.1 Logging In

If you are registered for the course, you will automatically have an account. If you are not registered, you need to contact a study counselor. Newly registered students might also have to wait for 24h before the account is available. Note that course registration is compulsory for all examination, not just the lab!

Now go to `https://snickerboa.it.liu.se` and click on "Login via SAML" and login with your LiU-id. In the next step you are free to choose how your name will be displayed on the scoreboard. The scoreboard is publicly available and is also displayed on screens around campus, so it can be a good idea not to use your real name. Note that once this name is set you can not change it again. We reserve the right to ban stupid and/or offensive user names for any reason.

Chapter 3

Performing the Lab

The lab contains a number of modules that cover different topics in web penetration testing.

3.1 Assignments

In order to pass the lab, you are required to finish all 21 assignments (see section 3.6). In order to prepare yourself for the assignments, there are also lessons which give a gentle introduction to the topic at hand. You can solve the assignments in any order you want.

There are also extra challenges, beyond what we require for a passing grade, if you wish to try your skill. Note that the lab assistant will not help you with the challenges, you have to do your own research here.

3.2 Result Keys

For each lesson and challenge your goal is to retrieve the so-called "result key". When you finish a lesson or challenge, the server detects that "it has been hacked" and gives you the key. Paste this key into the box on top, shown in fig. 3.1. Depending on the assignment at hand, the format of the result key can vary, but it might look something like the following:

```
3BSuxx3ORkg3zcq7Y7DOml8a46M23AS97RFtaFqE4HN+14NDiafeYaKiDPwUa/xrrDSfHHRohE6d5Up
PrJAOnAV15bu6ukOU3G5qs5FFEjE=
```

Whenever you receive a result key, paste it to the "Submit Result Key Here" box on the top of the screen. Don't even think of brute-forcing the key, there are detection mechanisms in place and this can be considered cheating. Also, each student will get an individual result key, and sharing keys with your friends is easily detected and not allowed.

3.3 Best Practices

It is a good idea to keep notes of how you pass each challenge. While your progress on the server is backed up frequently we can never be too sure. An important part of computer

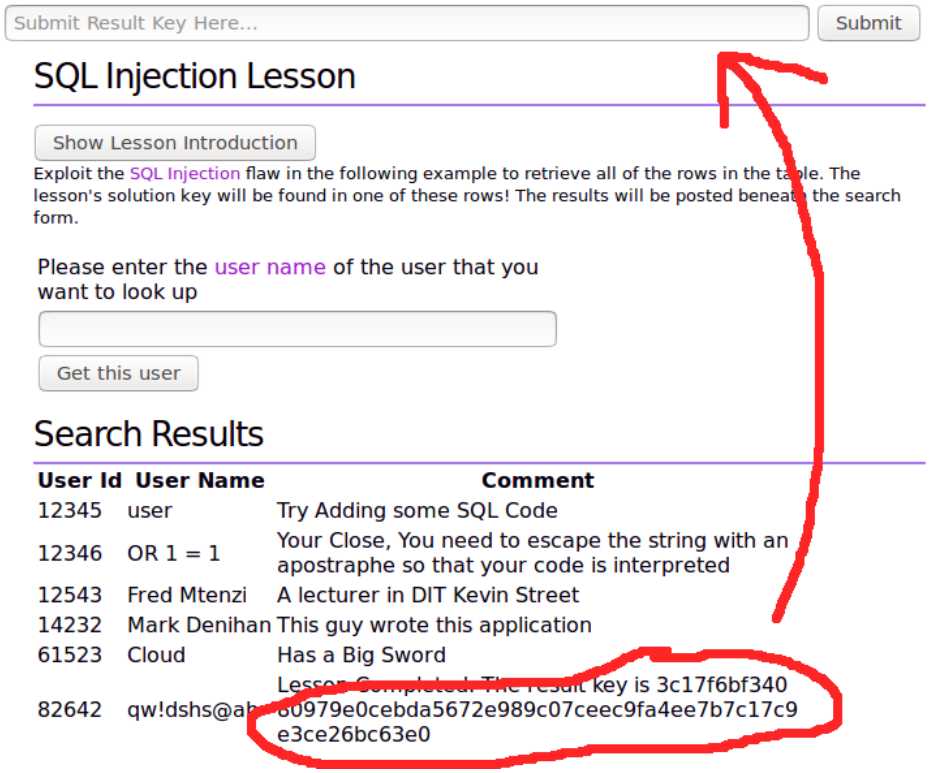

Figure 3.1: Example of result key and where to paste it.

security is to have a disaster recovery plan, and if the database was affected by corruption some written notes can help you recover faster.

3.4 Scoreboard

Whenever you finish a lesson, assignment, or challenge, it will show up on the scoreboard. The scoreboard is public, and anybody can see the progress of the participants. In addition, the scoreboard will be displayed on monitors around campus, especially around Cafe Java in the B-building.

The scoreboard is just for fun, and in order to pass you are only required to finish the assignments. If you have finished the assignments and want more points, you are welcome to try the challenges. Again: the scoreboard has nothing do with your grade! See fig. 3.2 for an example of what the scoreboard looks like. For each completed lesson, assignment, or challenge you will receive points, so the more challenges you finish, the more bragging rights you have. Also, harder challenges give more points.

Your name will not appear on the scoreboard until you have finished your first challenge. Also, there are medals! A gold medal is awarded to the student who finishes a lesson or challenge nobody else has finished yet. A silver medal is given to the second one, and bronze to the third. In the scoreboard there will therefore be users with medals in addition to the normal point score. These medals are not worth any points, but will be used as tiebreakers in the competition. In order to break a tie, we first count the gold medals, then the silver medals, and then the bronze medals. After this, we will draw lots if needed.

But remember, the scoreboard is just for fun. It has nothing to do with actually passing the lab.

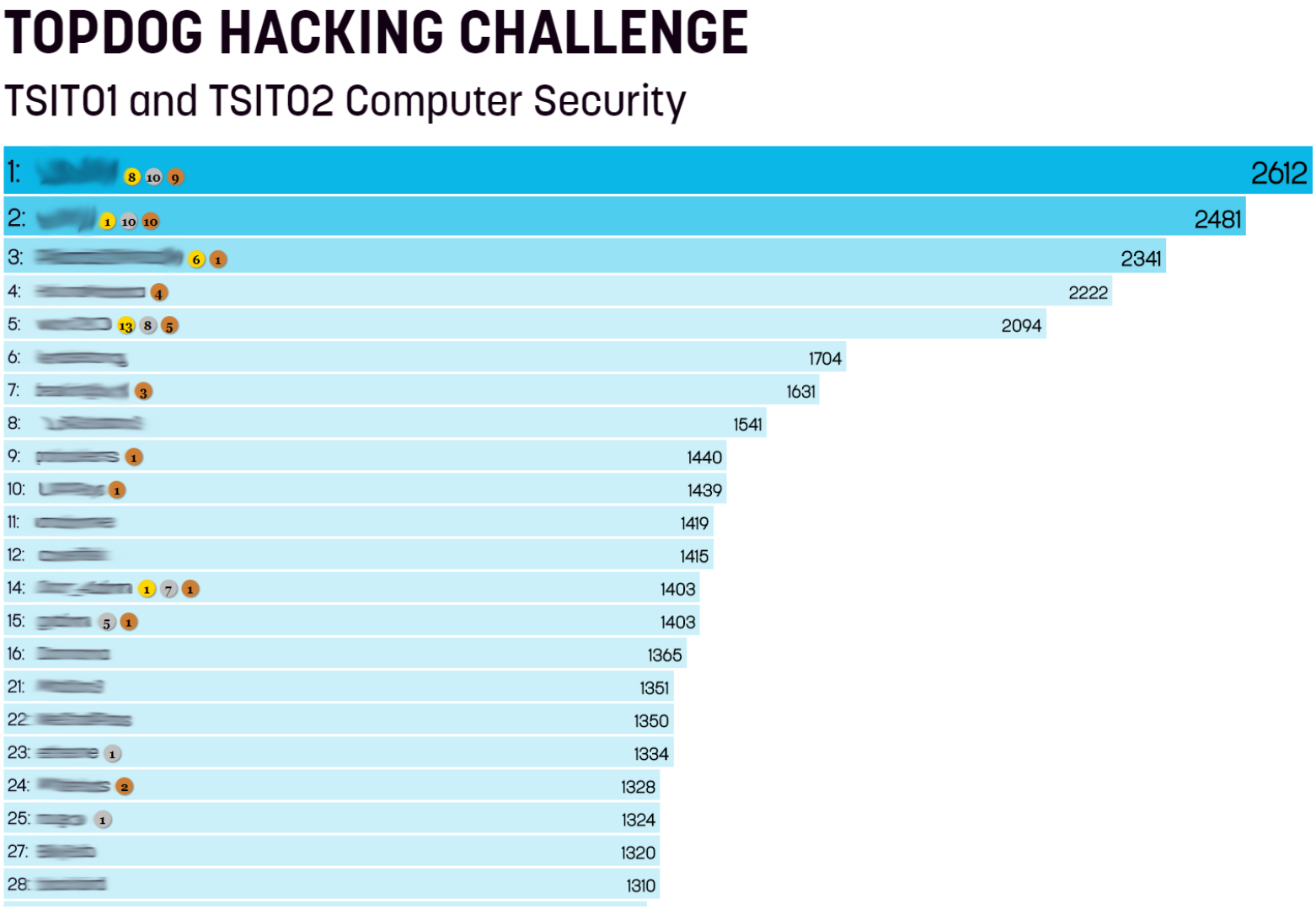

Figure 3.2: The TopDog scoreboard from 2016. Note the medals on some of the usernames.

3.5 List of Lessons

The following lessons are available:

Broken Session Management

Cross Site Request Forgery (CSRF)

Cross Site Scripting (XSS)

Failure to Restrict URL Access

Insecure Cryptographic Storage

Insecure Direct Object References

Poor Data Validation

Security Misconfiguration

SQL Injection

Unvalidated Redirects and Forwards

3.6 List of Assignments

Below are the required assignments (there are hidden hints!):

Session Management Challenge 1

Poor Data Validation 1

Cross Site Scripting 1

Session Management Challenge 2

Session Management Challenge 3

SQL Injection 1

SQL Injection 2

Insecure Cryptographic Storage Challenge 1

Insecure Cryptographic Storage Challenge 2

Insecure Direct Object Reference Challenge 1

Insecure Direct Object Reference Challenge 2

Poor Data Validation 2

Failure to Restrict URL Access 1

CSRF 1

Cross Site Scripting 2

Session Management Challenge 4

Failure to Restrict URL Access 2

Cross Site Scripting 3

Insecure Cryptographic Storage Challenge 3

SQL Injection 3

Insecure Direct Object Reference Bank Observe that the account name needs to be unique, otherwise you will just get the message *"An Error Occurred! You must be getting funky! Could not create account!"*)

5.8

Chapter 4

Contributing to the lab

We think that this lab is a great way to teach important concepts in information security, and want to encourage you to submit your ideas and feedback to us. Do visit our internal project page, where you can help us out! This is the LiU GitLab server, and there are several ways of helping out.

4.1 Submit a Bug Report

If you run into a bug, you are encouraged either to contact us directly, or file a bug report on gitlab. Click the link shown in fig. 4.1 and fill out the details. The more specific you can be, the more you will be able to help us out!

4.2 Ideas for Enhancements

You can file more than just bug reports! If you feel something is missing, or something could be done better, you can also submit enhancement ideas.

4.3 Fork It!

Are you familiar with git? If yes, then you can contribute even more! Fork the repository, commit your own changes, and then send us a merge request (sometimes called a pull request). If you need help, please see the official documentation.

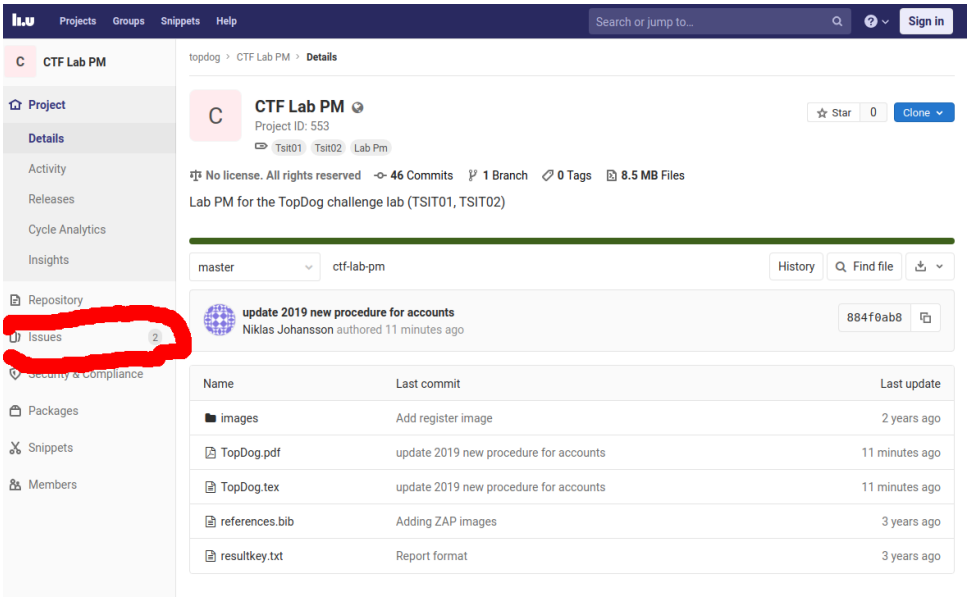

Figure 4.1: Click on Issues to file an issue report.

Chapter 5

Frequently Asked Questions (FAQ)

This section will be updated with frequently asked questions about the lab.

5.1 I am Stuck, What Should I Do?

First make sure you have read through the whole lab instructions. Secondly, consult and discuss with a friend (this is the best way of getting new ideas). Lastly, use the assistances drop-in time slot (see section 1.2).

5.2 There is Something Wrong With the Server!

First check that your Internet connection is working and that your attack proxy isn't giving you problems. If the TopDog server is unavailable, or if there's some *technical* issue with it that has nothing to do with the lab itself, first wait a few minutes. If it doesn't come back it might be an outage (planned or unplanned). If we are doing some planned work on the server this will be posted on Lisam.

If the server is still down and there's nothing on Lisam saying it's a planned outage, the server might be down. Please send an e-mail to the lab assistant, see section 1.6.

5.3 I Can't access the network after configuring ZAP

If this happens or if you can't proceed by "accepting the risk" untrusted certificate. Then you should export the certificate from ZAP and import it into Firefox.

Start by going to

ZAP > Tools > Options > Dynamic SSL Certificates,

and Save the certificate (see fig. 5.1).

Now you have to import the certificate in Firefox. Go to

Preferences > Privacy and Security > Certificates > View Certificates...

and hit "Import" (see fig. 5.2).

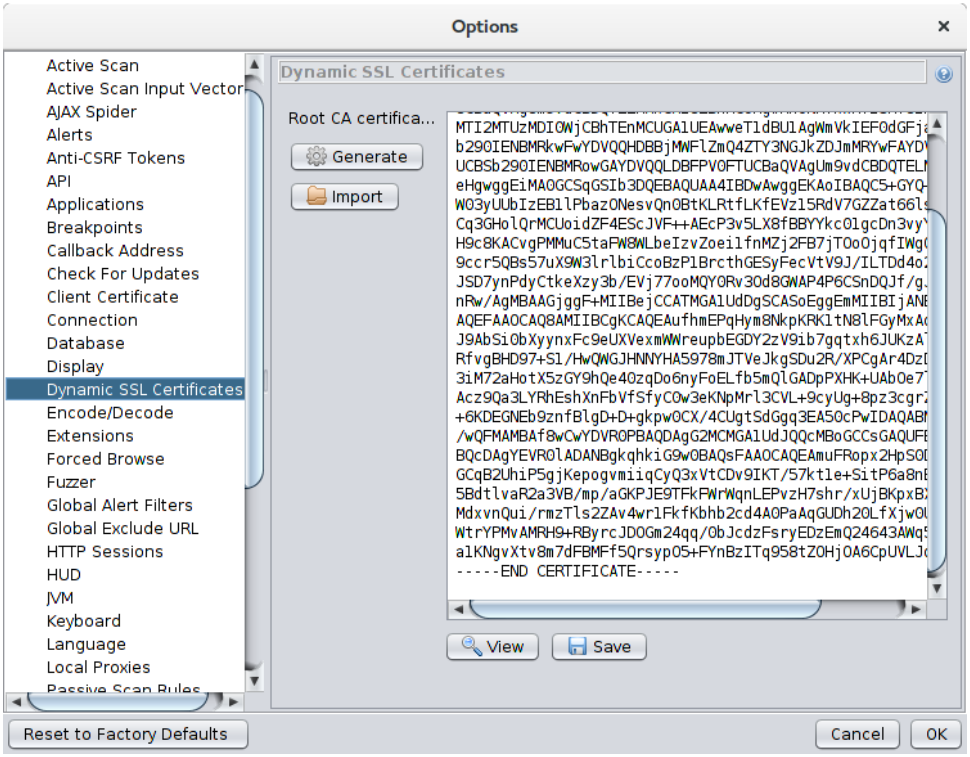

Figure 5.1: Click on save to export the certificate.

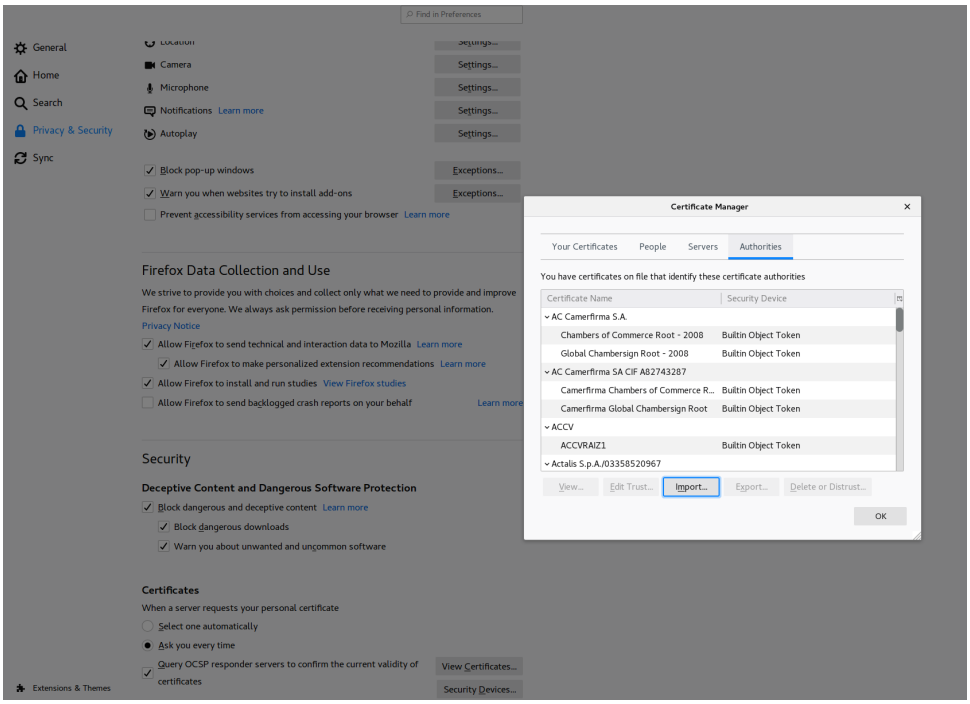

Figure 5.2: Click import and point to the ZAP certificate.

Figure 5.3: Click import and point to the ZAP certificate.

You will now get a pop-up where you verify that you trust the certificate (see fig. 5.3). Fill in the two boxes, hit OK, and you are done.

5.4 Can I Get Bonus Points For the Exam?

The scoreboard and its points, bonus points, and medals is for fun only. They have absolutely nothing to do with passing the lab or with the examination of the course. The lab assistant can see how many assignments you have finished, independently of the scoreboard.

5.5 I Finished the Lab and Want Something More Challenging!

Try your skills on the challenges! If this is still not enough, check out appendix B!

5.6 I Didn't Get a Result Key, Only "Key Should be here! Please refresh the home page and try again! If that doesn't work, sign in and out again!"

This is a bug has happened in older versions. We *hope* that this issue has now been solved, but if it does happen, please contact us (section 1.6) and we'll help you.

5.7 The Result Key in the Insecure Crypto Challenges Isn't Working!

Make sure you check that you've got UPPERCASE/lowercase correctly. Some online calculators will mess this up. Also make sure it handles spaces correctly.

5.8 In the Insecure Direct Object Reference Bank Challenge, There's no Money Left

It can happen that the total amount of money is too small to pass the lab. In this case, contact us at section 1.6 and we'll fill up bank with some more money to steal!

5.9 I Love Computer Security and I Want To Learn More!

We think so, too! Check out some of our other courses, for instance TSIT03 Cryptology that is given every first half of the fall semester. Also, check out the courses TDDD17 Information Security, Second Course and TDDC90 Software Security given at the Department of Computer Science. If you want more challenges, we've added some information in appendix B.

Appendix A

Tools

Penetration testing requires you to have a large and diverse toolbox. In this lab, you will mostly use online tools (that you'll have to find yourself) and one offline tool: ZAP. The online tools can be things such as online calculators, hex-to-dec-converters, decryption tools for cryptographic algorithms etc. Use Google! Also, the slides from the lab preparation lecture will be of use to you.

A.1 Viewing the Source Code

The first step in most web attacks is usually to look at the source code. This will show you the raw HTML/CSS/JavaScript that builds up the page. For a quick reference on what the HTML tags do, check out the W3 HTML Reference[1]. Figure A.1 shows the source code of one of the modules.

There are two main ways to view source code. The "traditional" way is to use the `View Source` feature found in most web browsers. Right-click the page and select "View Source". You can also use the "inspect" feature to view the source related to a particular item on a page by right-clicking it and selecting "Inspect" or "Inspect Element", depending on your web browser.

In the TopDog challenge you must remember that the web modules are located in an `iframe` in the web page. You must therefore click *inside* the module itself and select "View Frame Source", or "This Frame" followed by "View Source" as shown in fig. A.2 (this example assumes you are using Firefox). Otherwise, you will be reading the source code of TopDog itself and not the module.

A.2 The Zed Attack Proxy (ZAP)

ZAP is the Zed Attack Proxy by OWASP[2]. You will use this tool to modify HTTP packets sent between your web browser and the web server. An attack proxy is the most important tool for pentesting web applications. There are of course other attack proxies for you to use, but ZAP is the tool we can help you with during the coaching sessions. ZAP runs on Windows, Linux and OSX and requires Java 7 or higher.

[1] http://www.w3schools.com/tags/

[2] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

```
<h2 class="title">Failure To Restrict URL Access Challenge 1</h2>
<p>
    To recover the result key for this challenge you need to obtain the current server status message from an administr
    <br/>
    <br/>
    Use this form to view the status of the server <!-- from the point of view of a peasant or guest  -->
    <br/>
    <br/>
    <form id="leForm" action="javascript:;">
        <table>
        <tr><td>
            <div id="submitButton">
            <input type="submit" value="Get Server Status"/></div>
            <p style="display: none;" id="loadingSign">Loading</p>
            <div style="display: none;" id="hintButton"><input type="button" value="Would you like a hint?" id="theHint
        </td></tr>
        </table>
    </form>

    <div id="resultsDiv"></div>
</p>
```

Figure A.1: An example of a source code of a module.

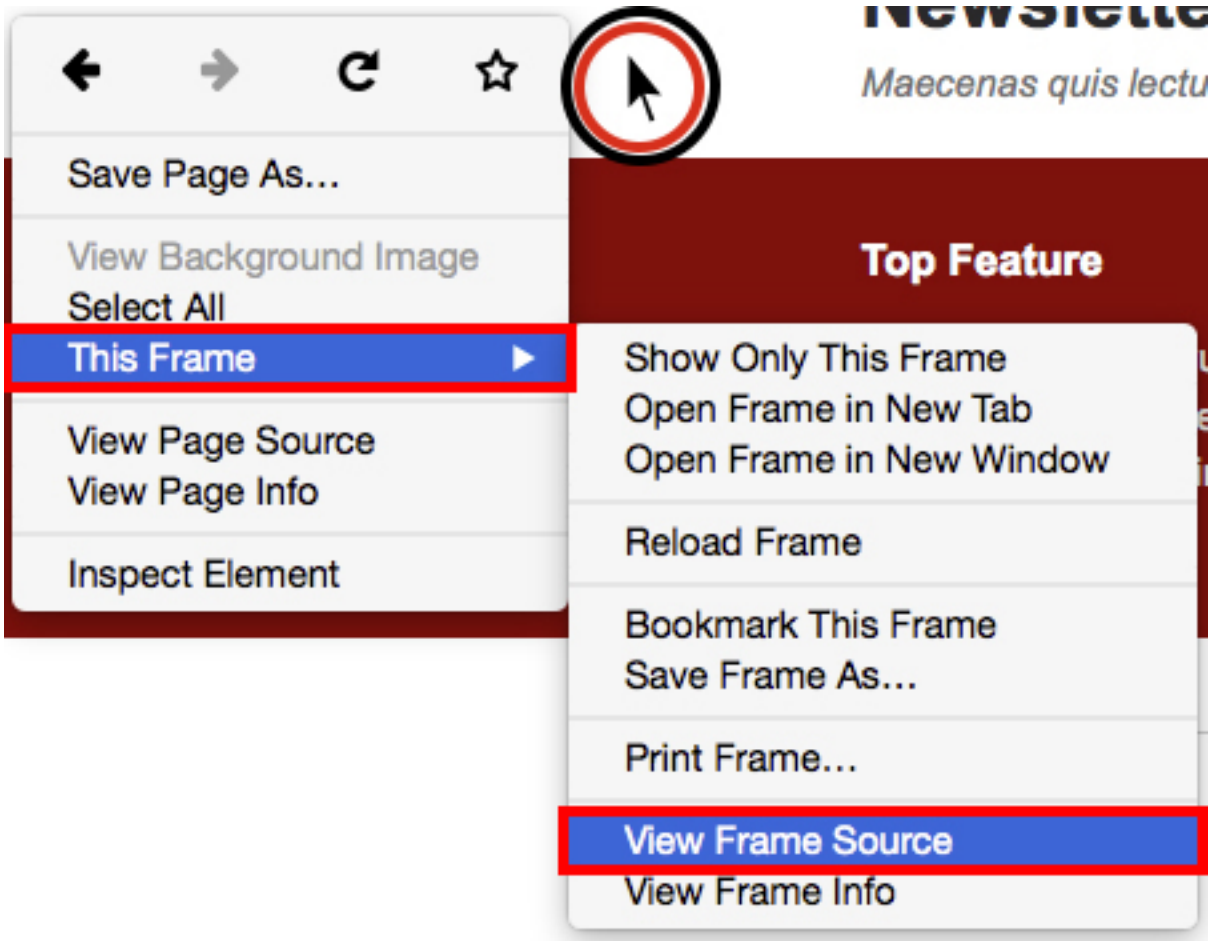

Figure A.2: How to view only the source of the `iframe` containing the module.

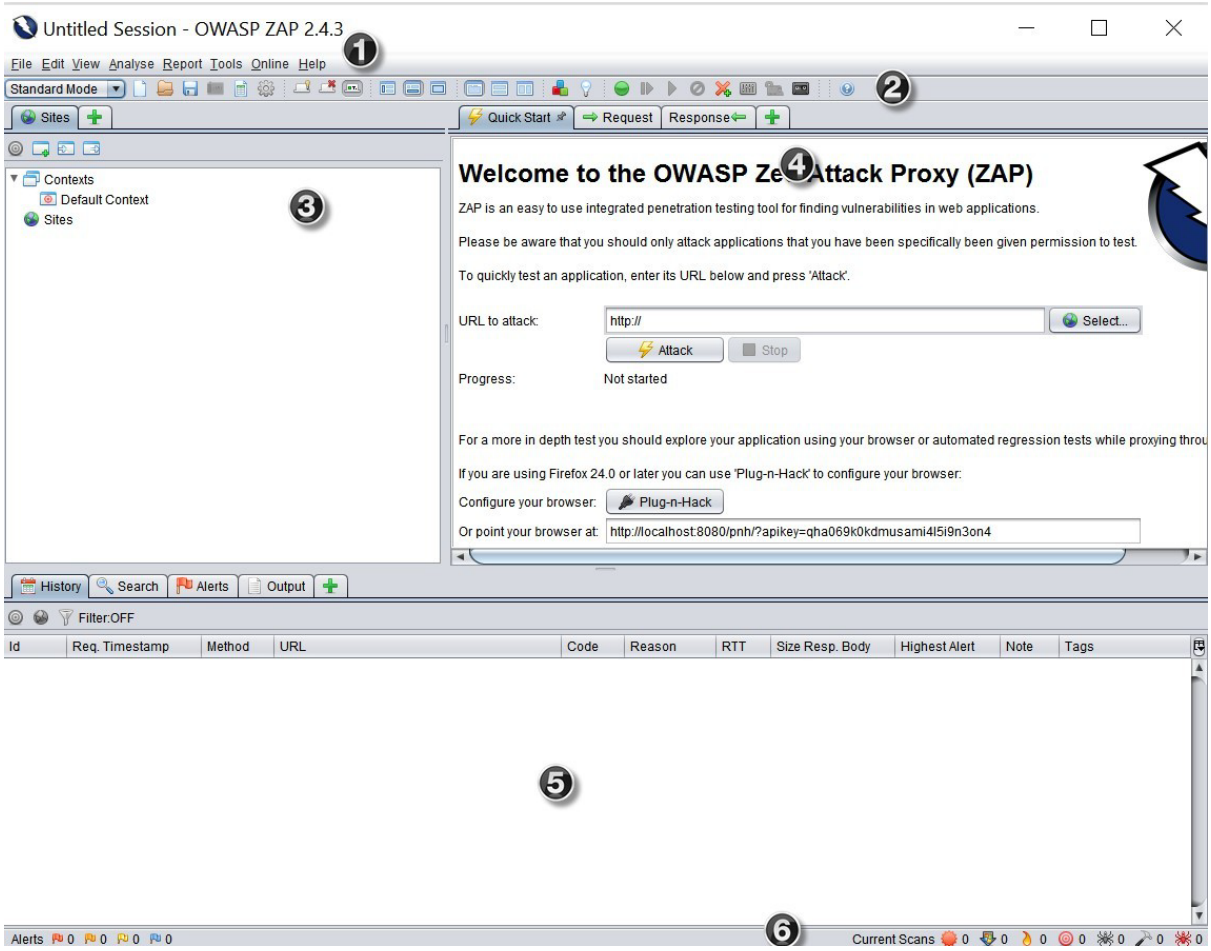

Figure A.3: Overview of the ZAP interface. 1. Menu Bar – Provides access to many of the automated and manual tools. 2. Toolbar – Includes buttons which provide easy access to most commonly used features. 3. Tree Window – Displays the Sites tree and the Scripts tree. 4. Workspace Window – Displays requests, responses, and scripts and allows you to edit them. 5. Information Window – Displays details of the automated and manual tools. 6. Footer – Displays a summary of the alerts found and the status of the main automated tools.

A.2.1 Configuring ZAP

Installing ZAP is easy. If you don't have Java, the installer will help you download and install it. If you have any trouble, check the ZAP Quick Start Guide[3] or the ZAP Wiki[4]. Upon first startup, ZAP will ask you if you want to persist the session. It's safe to say yes. After starting up, you will see the ZAP interface as shown in fig. A.3.

In order to use the attack proxy, you will need to configure your web browser to connect through it. Here, it is recommended that you download and install a secondary web browser to your computer, so that you have one normal browser (for googling and general browsing) and one "attack browser" for use with TopDog. Otherwise, ZAP will

[3] https://github.com/zaproxy/zaproxy/releases/download/2.5.0/ZAPGettingStartedGuide-2.5.pdf

[4] https://github.com/zaproxy/zaproxy/wiki/Introduction

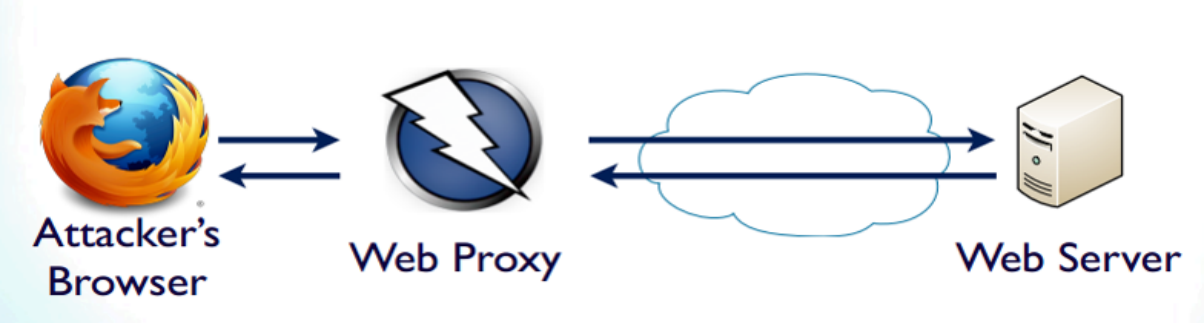

Figure A.4: Schematic of the attack proxy

intercept all your HTTPS sessions (i.e. also your general web browsing), which is very annoying.

By default, ZAP listens to connections on port 8080. Therefore, configure the attack browser (the one you use for TopDog) to use `localhost:8080` as the proxy configuration for HTTP and HTTPS protocols.

For instance, the configuration[5] for Firefox is shown in fig. A.5. Instructions for configuring proxy settings for Chrome can be found here: https://support.google.com/chrome/answer/96815.

Now, using the attack browser, go to the TopDog page: https://snickerboa.it.liu.se. It might be that your browser gives a warning about your connection being insecure since ZAP decrypts and re-encrypts HTTPS traffic (see fig. A.6). Remember that we talked about this in the lecture. You will have to accept the ZAP certificate and add it as an exception to the attack browser.

A.2.2 Intercepting HTTP(S) Traffic With ZAP

Now you can browse around in TopDog and see that the traffic appears in ZAP. In the left-hand pane you see `Sites`. Expand it and you see the site https://snickerboa.it.liu.se. Inside, you see the different requests (mainly `GET` and `POST`) that were made to the server.

On the main pane (the window that says "Welcome to OWASP...") there are three tabs on top: `Quick Start`, `Request`, and `Response`. The `Request` tab shows the request sent by the web browser and the `Response` tab shows the response by the web server. See fig. A.7 for an example of a request package. Figure A.8 is an example of a response package.

Now we want to capture a HTTP response for ourselves. Begin by pressing the green circle (see fig. A.9) so that it turns red. ZAP will now capture all requests that pass through it. Then, in your attack browser, perform some action that sends a HTTP request, like clicking a link or a button. ZAP will then pop up, showing you the packet it captured (just like in fig. A.7). You can now modify the packet before sending it on, or just forward it without modification. Either way, press the `step` button to pass the request on to the server and capture the next packet.

[5] http://www.wikihow.com/Enter-Proxy-Settings-in-Firefox

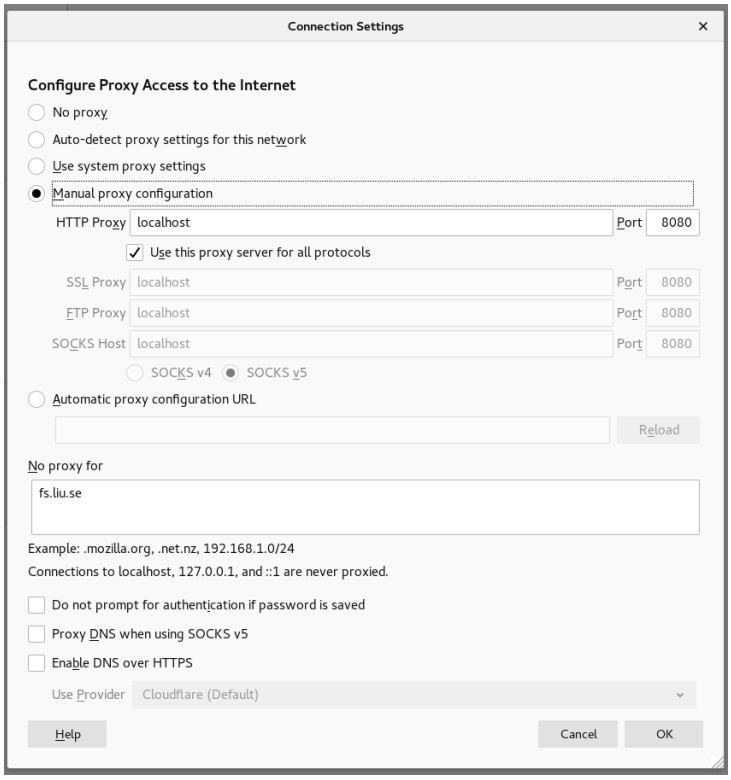

Figure A.5: Firefox proxy configuration (Preferences -> General -> Network Settings -> Settings...).

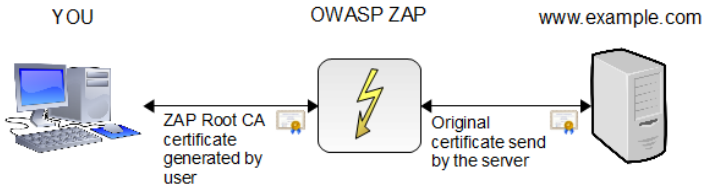

Figure A.6: Schematic diagram of ZAP when dealing with HTTPS traffic.

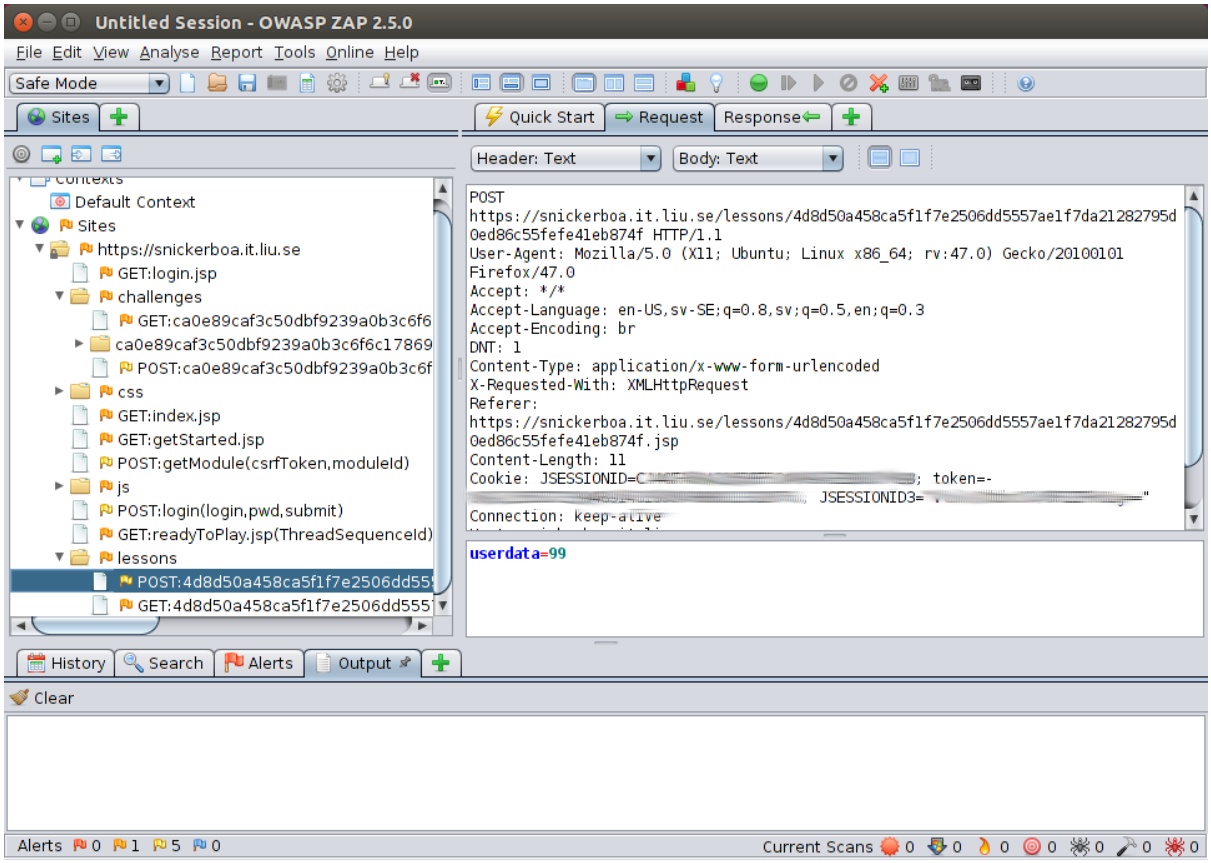

Figure A.7: Screenshot of ZAP showing a HTTP request package sent from the web browser to the web server. The request belongs to the lesson module for cross-site scripting. In the lower right side you can see `userdata=99`, which means that the request contains POST data from a form with the varaible `userdata` set to the value 99.

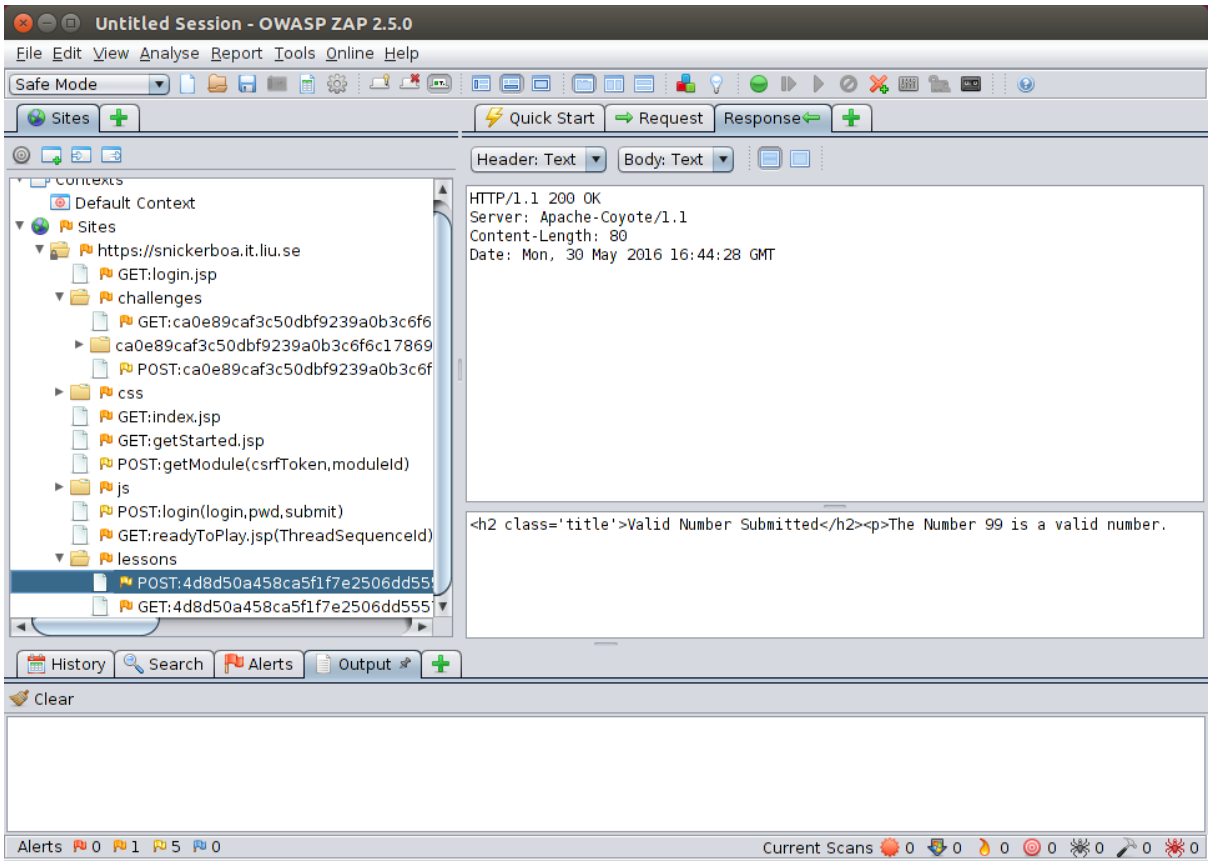

Figure A.8: ZAP showing a response packet from the web server to the web browser. The body of the response shows a HTML-encoded text saying that “the number 99 is a valid number”.

Figure A.9: The important Break, Step, and Play buttons in ZAP (the three leftmost buttons)

The web server will then process the request and reply with a HTTP response. Since you clicked `step`, ZAP will capture this request as well and show it to you, just like in fig. A.8. Now, click the `play` button which will deactivate the break point entirely and send the response back to the attack browser. We recommend using Firefox as your attack browser.

ZAP has many more features, but for this lab you only need to concern yourself with the break/step/play features. Note that the web browser will be stuck on "loading" until ZAP has sent both the request and response to their destinations.

Appendix B

Capturing The Flag

TopDog is what the hacking community calls a CTF, or Capture The Flag. CTF:s are a good way of practicing one's skills in order to become better at pentesting, reverse-engineering, cracking, etc. It is common for security conferences to have CTF competitions where teams try to solve a number of challenges set up by the organizers. Prizes are usually awarded for the teams who finish first.

If you found this lab course interesting and want more CTF challenges, check out this list: https://captf.com/practice-ctf/.

There is also a team of LiU students called LiUHack who regularly participate in CTF challenges.

Appendix C

About This Document

This lab memo is intended for students of the computer security courses `TSIT01` and `TSIT02` at Linköping University.

C.1 Changelog

2019 Lab has now been integrated with LiU-ID login.

2017 Revised for the 2017 course.

2016 Initial version.

C.2 Acknowledgements

This lab owes its existence to Anders Märak Leffler who brought this software to my attention back in 2015. I also want to thank the OWASP Foundation and the OWASP chapter in Gothenburg for help with getting started. Thanks to the LiU IT department who was willing to set up and support a web application server that, contrary to all common sense and in violation of probably a dozen IT policies, contains all kinds of web vulnerabilities. Also thanks to Niklas Johansson for helping me get all the lab details straight and, of course, prof. Jan-Åke Larsson, who gave us the go-ahead to build what is probably going to be a very interesting lab course.

Linköping, November 2016
Jonathan Jogenfors