Lab 1: Web Same-Origin-Policy

Dr. Lotfi Ben Othmane Lisa Nguyen Quang Do

23 October 2015

Rules and Grading

This lab is due on **05.11.2015**, at **23:59**.
Your dedicated repository for the labs is: `https://repository.st.informatik.tu-darmstadt.de/sse/secdev/2015/students/labs/grp-X/`. You are to submit a report named `Lab1-Submission.pdf` at this location. We will *only* grade this document.

Your report should describe what you have done and observed. Explain how you found the answer, the tools you used, etc. This can include screenshots and code snippets. You are encouraged to pursue further investigations. When possible, describe and explain interesting observations. An original answer may receive bonus points (at the discretion of your instructor). At the same time, your answers should be concise.

Cheating will be sanctioned with a 0 grade. Here is what we define as cheating:

- getting solutions from other students IS cheating
- actively hindering other students from completing the lab (this includes (un)-intended DoS on shared resources) IS cheating
- exploiting an unforeseen vulnerability in one of the exercises is NOT cheating
- exploiting a vulnerability in our grading system is NOT cheating (responsible disclosure will be rewarded)

Work environment setup

The labs will take place inside a VirtualBox VM. The virtual machine image is available on the course's SVN: `https://repository.st.informatik.tu-darmstadt.de/sse/secdev/2015/public/labs/SSDLabs.ova`. To run it, you will need to download and install VirtualBox, freely available at `https://www.virtualbox.org/wiki/Downloads` (or installable from your package manager if you are using Linux).

The image we provide contains several machines. Some of the labs will require you to use them all, others will only need one of them. You are free to log into any of the machines from the `student` account, using the password `SSD15Stud`. The `root` password is `SSD2015`. Feel free to create yourself other users if needed.

Remember to snapshot your VMs, in case you accidentally destroy something important.

Network setup

For some exercices, the VMs will need to run on the same network. Perform the following steps to set up a NAT network in VirtualBox:

- Create a NAT network (File>Preferences>Network). Call the network `SSDNetwork` and leave the default values.
- For each VM, go to Settings>Network. Enable the first network adapter, attach it to the NAT Network with the name `SSDNetwork`.

By default, your VMs are connected to the outside world. If you think your experimentations might be dangerous to others, disable networking beforehand (`/etc/init.d/net.ethX stop`). You may start an SSH server in the main container of the VM. You can also mount a shared folder between your host and your VM, to transfer files back and forth.

Warmup (3 points)

Now that the VM is running, answer the following questions:

(a) How many machines are there? What are their names? (1 point)

(b) What is the name of the distribution running in the VM? (1 point)

(c) What is the subnet used by the machines? (1 point)

Exercise 1: Web Same-Origin Policy (87 points)

In this exercise, you will explore the Same-Origin Policy in web applications. The security model of existing web browsers is based on that policy, which provides basic protection features to web applications.

You are allowed to use `Wireshark` or any other tool if you feel the need for it.

1 Lab environment setup (9 points)

For this lab, we will need three things: (1) the Firefox web browser, (2) an Apache web server, and (3) the `Collaborative` open-source project management web application. We will use the `LiveHTTPHeaders` and `Edit Cookies` extension of Firefox.

1.1 Start the Apache server

```
sudo service apache2 start
```

1.2 DNS

The DNS was pre-configured for this lab.

(a) What are the IP addresses of the DNSs? How did you find them? (2 points)

(b) Open Firefox and type in the address `http://www.soplab.com`. Where is the webpage hosted? What about `http://www.soplabattacker.com` and `http://www.soplabcollabtive.com`? What happens when you try to access those three addresses outside of the VM? (2 points)

(c) Look at the files `/etc/hosts` and `/etc/apache2/sites-available/default`. Explain your observations from the previous question. (4 points)

1.3 The `Collaborative` web application

Access the login page for the `Collaborative` application for the SOP lab (`http://www.soplabcollabtive.com`). You can log in with the following credentials:

```
username: admin
password: admin
```

(a) How many users are currently registered? What are their usernames? (1 point)

1.4 Disabling cache

The lab will require you to make some modifications to the web applications while you are using them. To ensure that the web browser always fetches the page from the modified web application and not from its cache, you can disable the cache. Type `about:config` in Firefox's address bar, and setup the following preferences:

```
browser.cache.memory.enable // set to false, default = true
browser.cache.disk.enable //set to false, default = true
browser.cache.check_doc_frequency
// 1 = everytime, 3 = as needed (default)
```

2 Understanding DOM and Cookies (28 points)

(a) Briefly explain what the Document Object Model (DOM) of an HTML page is. Draw the DOM of the code shown in Figure 1. (2 points)

(b) Write a JavaScript function that traverses and displays the entire DOM tree for Figure 1. The function should show the `h1` heading and paragraph added to the document by the `appendp` function. Include your code in your report. (3 points)

(c) What are cookies? What are they typically used for? In particular, describe how cookies are created and which role they play in the interaction between the web browser and the web application. What do the following optional cookie attributes correspond to: `expires`, `max-age`, `path`, `domain`, `secure`? (2 points)

(d) What is a cookie-based session management scheme? What is a session cookie? (1 point)

(e) Read the source code of `http://www.soplab.com/cookie.html`. How does one store, read and process cookies in JavaScript? (2 points)

(f) Write your own HTML page that, on click on a button, displays how many times the web page has been visited by the current user. Include your code in your report. (4 points)

(g) The `Collaborative` web application (`http://www.soplabcollabtive.com`) uses a cookie-based session management scheme. Identify the name of its session cookie. How did you find it? (2 points)

(h) Use the `LiveHTTPHeaders` extension to find out when the web application creates the session cookie in the web browser (Tools>Live HTTP Headers). Describe the interactions between the web application and the web browser. (4 points)

(i) Turn on a second VM and make sure both machines are connected to the same network. On both machines, open Firefox and load the `Collaborative` web application login page. *Ensure that both machines are accessing the same page, and not their local ones independantly.*

Assume the following scenario: On the first machine, Barnaby, the admin of the `Collaborative` web application, is logged in with his account, `admin` (password:

```
<html>
  <head>
  <title>Self-modifying HTML</title>
  <script>
  function appendp() {
    var h1_node = document.createElement("h1");
    h1_node.innerHTML = "Self-modifying HTML Document";
    document.childNodes[0].childNodes[2].appendChild(h1_node);
    var p_node = document.createElement("p");
    p_node.innerHTML = "This web page illustrates how
          DOM API can be used to modify a web page";
    document.childNodes[0].childNodes[2].appendChild(p_node);
  }
  function gethtmlchildren() {
    var entiredoc = document.childNodes[0];
    var docnodes = entiredoc.childNodes;
    for(i=0; i<docnodes.length; i++)
      alert(docnodes[i].nodeName);
  }
  </script>
  </head>

  <body name="bodybody" >
  <h1>SOP Lab</h1>
  <script> appendp(); </script>
  <input type="button" value="Display children of HTML tag"
        onclick=gethtmlchildren() >
  </body>
</html>
```

Figure 1: A web page with a JavaScript program updating the web page dynamically

`admin`. Kotetsu, on the second machine, does not have any valid credentials to access the application. Suppose that Kotetsu somehow obtained Barnaby's session cookie. From that information, can Kotetsu log in the admin account? Detail the procedure. You can provide screenshots to illustrate your explanations. (8 points)

3 SOP for DOM and Cookies (16 points)

In this section, you will explore how web browsers identify the origin of web applications and how access restrictions are applied on DOM objects and cookies. For this section, you will use the following web page: `http://www.soplab.com/index.html`. It contains two frames: the main frame and the navigation frame.

(a) Look at the code in `http://www.soplab.com/navigation.html`. What is displayed in the main frame? In the navigation frame? Which resources are accessed when the "Read cookie" button is clicked? The "View source" button? (2 points)

(b) Provide the following URL to the navigation frame: `http://www.google.com`. *From the navigation frame*, are you able to access the cookies and DOM objects of the web page? Why? (3 points)

(c) Provide the following URL to the navigation frame: `www.soplab.com/navigation.html`. *From the navigation frame*, can you access the cookies and DOM objects of the whole web page? Why? (3 points)

(d) With the same URL (`www.soplab.com/navigation.html`), can you access the cookies and DOM objects of the whole web page *from the main frame*? Why? (3 points)

(e) Note that the cookies and DOM objects are not the only objects restricted by the SOP. The `history` object and URL of the frame are too. Test them and report your observations. (5 points)

4 SOP for `XMLHttpRequest` (13 points)

(a) What are XMLHttpRequest? What are they commonly used for? (1 point)

(b) What does the following code snippet do?

```
<script>
        xhr = new XMLHttpRequest();
        xhr.open(POST,"http://www.soplabcollabtive.com/
                admin.php?action=addpro",true);
        xhr.send(null);
</script>
```

Does the SOP apply to the target URL? (1 point)

(c) A more complex example of XMLHttpRequest is contained in `www.soplab.com/navigation.html`. Is the SOP extended to the target URL of the HTTP requests? (1 point)

(d) What are the dangers of not extending the SOP to the XMLHttpRequest requests? Describe a few possible attacks. (10 points)

5 Exceptions from the SOP (8 points)

(a) Some HTML tags can also trigger an HTTP request within a web page. For example, the `img` tag triggers an HTTP GET request. Investigate if the SOP applies for `img`, `frame`, `iframe`, and `a`. Describe your experiments and report your observations. You can craft a web page in `www.soplabattacker.com/` to make requests to `www.soplab.com/`. Code snippets are welcome in your report.

6 Conclusion (13 points)

Summarize what you learnt about the SOP. Include the following observations:

- What is it? How does it work?
- Why is it used? What could happen without such a policy?
- What should a web programmer be especially wary about concerning SOP and security?

Exercise 2 (10 points)

This exercise is given in the form of open challenges. There are purposely harder to solve than the guided first exercise and you will need to look for information yourself. There are several ways to solve the exercise, you are free to submit any working solution, as long as it does not invlove cheating.

A malevolent employee of your company has stolen a confidential technology, but he was caught by an IDS. You are responsible for investigating the case.

(a) **Network - secure channels: (3 points)** The network trace can be found here: `https://repository.st.informatik.tu-darmstadt.de/sse/secdev/2015/public/labs/Lab1/capture.pcap`. Find the username of the thief and the location of the stolen file. Briefly describe your reasoning.

(b) **Cryptography - confidentiality: (4 points)** The thief was careful enough to encrypt the file containing the stolen technology. Find the name of the technology. Briefly describe how you found it.

Note: For those who haven't answered the first question, the file is available here: `https://repository.st.informatik.tu-darmstadt.de/sse/secdev/2015/public/labs/Lab1/secret_file`.

(c) **Steganography - protection mechanisms: (3 points)** You strongly believe that there is more to the file than it appears. It is possible that the thief has hidden a secret message in the file itself. Find the secret message. Briefly describe how you found it.

Note: For those who haven't answered the second question, another file containing the same message is available here: `https://repository.st.informatik.tu-darmstadt.de/sse/secdev/2015/public/labs/Lab1/image.png`.