Challenge 1: Implementing a privacy-enhancing proxy

1 Challenge

Design and implement a privacy-enhancing web-proxy in Java. All data[1] exchanged between your web browser and any web-server will be passing by this proxy. By properly filtering and manipulating the data being exchanged, your program will as much as possible protect your privacy, while maintaining the functionality of the websites visited.

2 Details

You will be provided by a program called `PEPFramework.java`, which already is a fully functional proxy, in which you yourself can add lines of programming code to process outgoing requests and incoming replies.

By default, it simply displays the first lines (i.e., the *request headers*) of each HTTP request received from your web browser, and the first lines of each HTTP response received from a server (i.e., *response headers*). You will have to modify this proxy server such that it does not just copy its input to its output, but instead takes actions that enhance the user's privacy, e.g. preventing some HTTP actions from completing, or removing privacy-sensitive data from the request to the server.

To make this easy, there are only two places in the code which you have to change: a part which is invoked once for every request, and a part which is invoked once for every response.

3 Getting started

3.1 Installing Java

If you do not yet have Java installed on your computer, do the following:

1. Install the Java Development Kit (JDK) from `http://www.oracle.com/technetwork/java/javase/downloads/index.html`. Choose the left option, showing the Java logo, not the NetBeans package.
2. Install an IDE, e.g. Eclipse from `http://www.eclipse.org/downloads/packages/eclipse-ide-java-deve` `lunasr1a`. Eclipse does not come with an installer, simply unpack the zip-file.
3. Start Eclipse via `eclipse.exe`, and configure a Workspace directory on the first run.

3.2 Compiling and starting the proxy

1. Get the framework code for the Privacy Enhancing Proxy from BlackBoard and unpack it.
2. In Eclipse, create a new Java Project and point it to the directory where you just unpacked the framework code. You can directly click on the 'Finish' button. **NB:** when chosing the location, don't traverse down into the `src` directory, but select that directory.
3. Close the Welcome tab, so you new project workspace becomes visible.
4. Create a new 'Run Configuration' via Run, Run Configurations in the menu.
   (a) In the Run Configurations window, double click 'Java Application'
   (b) Set 'Main class' to `proxy.PEPFramework`
   (c) Click 'Apply', and 'Run'. The console now shows the output of the program.

[1] HTTPS traffic will not pass your proxy, and for good reasons. Think of why this is the case.

5. Verify that you can compile and run the proxy using the 'Play'-button or CTRL+F11. **NB:** you can only run one instance at a time, so always stop the proxy before starting a new one.

3.3 Configuring a web browser to use the proxy

You will need to configure a browser to use the proxy: basically, telling it to not send the requests directly to the webserver, but to the proxy which will forward them. It is advised to use a secondary browser for this, so you can still use your favourite browser to search for information while doing this challenge.

Configuring Firefox

1. Go to Preferences via the menu button at the top-right.
2. Choose 'Advanced', then 'Network', then 'Settings' (under 'connection')
3. Select 'Manual proxy configuration' and fill in '127.0.0.1' and '8080' for 'HTTP Proxy' and 'Port' respectively. (Note that 127.0.0.1 is a special IP address by which a host can talk to itself, in this case the web browser talking to the proxy, both running on the same machine.)

Configuring Internet Explorer

- Go to Settings, 'Internet Options'
- Go to the 'Connections' tab
- Click on 'LAN Settings'
- Enable 'Use a proxy server '
- Use address '127.0.0.1', port '8080'
- Go to the advanced settings, and make sure to only use the proxy for normal HTTP traffic

Most other browsers can be configured in a similar fashion. Make sure that plugins like NoScript and AdBlock are disabled for this challenge. Optionally (in the case of Firefox) create a fresh profile.

3.4 Test the proxy

To test the proxy, browse to a simple website, like the one used for last Monday's observation session: `http://netsys.ewi.utwente.nl/wireshark/` . If all is well, the page should appear. Also, the proxy should print some request and reply headers. Check that!

4 MyProxy.java structure

Within `MyProxy.java` you see two methods that you may need to modify:

onRequest: This method (think "function" if you're used to C/C++) is executed whenever an HTTP request is sent through the proxy. Code can be put here to print the contents of the request (handy for inspecting what is happening), to modify, add or delete one or more of the request headers, or even to drop the request entirely.

The headers of the request are contained in a HashMap data structure, which allows for easy manipulation; see `https://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html`.

By default, the method contains a few lines of code which simply print all of the headers in the request.

onResponse: This method is executed whenever an HTTP response is handled by the proxy server. Again, code can be put here to print the contents of the response, and to modify it or its headers.

Inside this method, the response is available in two forms: again as a *HashMap* containing the response headers, and as a *byte array* containing the entire response (including the data). You can use whichever representation is most convenient for the task at hand.

If you want to modify the response, that can only be done through the byte array; changes made to the HashMap are discarded. As a hint, you may want to convert the byte array to a String and manipulate using for example `replaceAll`[2]; an example of this is given in comments in the code. If your modification changes the length of the response, then be sure to update the variable `inOctets` accordingly, otherwise the rest of the code of the proxy server will not send the right number of bytes to the client.

5 Exercises

In this series of exercises, we will be using the following websites. Before you start, have a quick look at each of them.

- http://thismachine.info
- http://www.whatbrowser.org
- http://www.whatsmyip.org/more-info-about-you/
- http://www.utwente.nl

5.1 Who are we talking to?

As you have browsed to the sites listed above, a lot of output was produced by the proxy. Along with every request, all of the headers (also for the returning response) are printed. As a first step, we want to know what hosts we have contacted. Alter the helper methods in such a way, that only the contacted hosts are outputted.

Which hosts are contacted when accessing our university's webpage?

5.2 Hiding your browser's identity

On the first website, you are presented with the name and version of your browser, along with your IP address. Discuss what you want to hide, and how you can do it. Can you hide every piece of information shown on that site? Why (not)?

Implement your solution in `MyProxy.java` and verify that you gained some anonymity.

5.3 There it is again

Browse to the second website. You'll notice that it is still possible to determine the browser you are using. Find out how this website is doing that, and how you can prevent it.

5.4 Utwente and Facebook

When accessing the www.utwente.nl site, also some request is sent to Facebook. Have a closer look at this request. What's in it? Do you really want Facebook to know this?

Enhance your proxy in such a way, that the tracking Facebook does, is prevented.

[2] https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#replaceAll(java.lang.String,java.lang.String)

5.5 How far can you go?

What more can you do to improve your online privacy? How far can you go without destroying normal browsing experiences? Do you encounter any ethical questions?

From here on, your proxy is limited to your creativity. While enhancing it, regularly check other websites that you normally use. To they break when using your proxy? Or do they gain in usability?

6 Submission and grading

For the final test, go to http://netsys.ewi.utwente.nl/challenge1/, and discuss the results with one of the assistants. Once your proxy has met the requirements, be sure to upload your code to Blackboard. Furthermore, every group should submit a brief report (one page maximum) describing their findings and approach.

7 Learning objectives

- Understand how HTTP works.
- Insight into how the web works nowadays, and in particular the privacy issues associated with that.
- Understand how solutions to privacy issues can be implemented, and the drawbacks associated.