IFS4103 Lab 1B: Inspecting Web Data on Your Kali Linux Host

Notes:

In our module, you will mainly use **Burp Suite** to perform various web pen-testing tasks. To intercept and inspect web data in HTTP requests and HTTP responses, you can use **Burp Proxy**, which is to be covered in Lab 2.

In this simple lab, **to inspect web data**, you can use **other simpler alternative command/tool/extension** on your Kali Linux machine which you've already set in your Lab 1A.

Objectives:

For this Lab 1B, you will perform the following:

1. To use **cURL** as a CLI-based browsing tool;
2. To inspect **HTML elements** using web browser's **developer tools**;
3. To observe and manipulate web sessions using a **browser extension**.

Task 1: Using cURL as a CLI-based Browsing Tool

cURL is an open-source command-line tool that allows you to **make any web requests**, and **observe the data exchanged** between a web client and server. cURL is available on all platforms from https://curl.se/.

The exercises below describe **some basic usages of cURL**. For more details, you can refer to a free **cURL e-book**, which is downloadable from: https://everything.curl.dev/.

A **documentation** that describes how to use cURL to automate HTTP jobs is also available at: https://curl.se/docs/httpscripting.html. Additionally, its **man page** is additionally available online at: https://curl.se/docs/manpage.html.

Run cURL with the **options given below**. If you want to omit cURL's progress meter, add the `-s` switch. For your easier output inspection, you may want to **direct the output to a file** (e.g. `$ curl -v` *target-URL* `>` *output-file* `2>&1`), or **pipe it to `less`** (e.g. `$ curl -v` *target-URL* `2>&1 | less`).

1. To get **the response body**, run the following basic HTTP request:

   ```
   $ curl http://httpbin.org/html
   ```

2. To inspect **request headers and response headers** in addition to the response body, use the **-v** switch:

   ```
   $ curl -v http://httpbin.org/html
   ```

3. To issue **a POST request** together with submitted parameters:

   ```
   $ curl -X POST target-URL -d
   "param1=value1&param2=value2"
   ```

4. To issue **a POST request** together with parameters **stored in a file**:

   ```
   $ curl -X POST target-URL -d @input-file
   ```

5. To **set a particular header** in a request (e.g. `User-Agent` header):

   ```
   $ curl -H "User-Agent: hahaha/0.0.0"
   http://httpbin.org/user-agent
   ```

6. To **simply set** the `User-Agent` header, you can also use:

```
$ curl -A "hehehe/1.0.0" http://httpbin.org/user-agent
```

7. To store **received cookies** into a file (e.g. `cookiejar.txt`):

```
$ curl -c cookiejar.txt http://www.cnn.com
```

8. To make a **request using cookies** stored in a file (e.g. `edited-cookiejar.txt`):

```
$ curl -b edited-cookiejar.txt http://www.cnn.com
```

Task 2: Inspecting HTML Elements using Web Browser's Developer Tools

You can also use the **developer tools** of your web browser (e.g. Chrome or Firefox) to inspect the **elements of a rendered web page**. One way of accessing the browser developer tools is by right-clicking a shown page and then select "**Inspect**" menu item.

You can do the following tasks while you are in your browser's developer tools:

- Inspect **all HTML elements** of a page;
- Use the **Browser Console**, and run some applicable **commands**;
- Observe **network activities** and performance.

For more details, you can refer to the following **documentations**: https://developer.chrome.com/docs/devtools/ (for Chrome DevTools) and https://firefox-source-docs.mozilla.org/devtools-user (for Firefox DevTools).

Task 3: Observing and Manipulating Web Sessions using Browser Extension

You can also install and use **a browser extension** to view and modify HTTP/HTTPS headers and also post parameters. For Firefox, you can try the following extensions among others:

- HTTP Header Live: https://addons.mozilla.org/en-US/firefox/addon/http-header-live/
- Modify Header Value: https://addons.mozilla.org/en-US/firefox/addon/modify-header-value/
- ModHeader: https://addons.mozilla.org/en-US/firefox/addon/modheader-firefox/

For Chrome, you can try the following extensions among others:

- ModHeader: https://chromewebstore.google.com/detail/empty-title/idgpnmonknjnojddfkpgkljpfnnfcklj
- Request Maker: https://chromewebstore.google.com/detail/request-maker/kajfghlhfkcocafkcjlajldicbikpgnp

Note:

- This task is included in this lab just to illustrate you **how a basic request interception and tampering can be done**. ***Burp Proxy***, which is to be covered in Lab 2, is an integrated and feature-rich interception proxy module/component of Burp Suite.