14-828/18-636: Homework 2

Released: Tuesday, February 18, 2020

Due: Thursday, March 5, 2020 by 1:29 PM Eastern Time

Name:

Andrew ID:

Scores

Part 1 (50 pts max):

Part 2 (50 pts max + 20 BONUS points):

Total (100 pts max + 20 BONUS points):

Guidelines

- Be neat and concise in your explanations.
- For Q1, you must use at most one page for your explanation for each problem (1.1, 1.2, ... each on separate pages) (code you wrote MUST be submitted to Canvas as explained below). Start each problem on a new page. You will need to map the sections of your PDF to problems in Gradescope.
- For questions where you are required to submit the code for the extensions, please label the folders for which the extension is for (e.g. Q1.1, Q1.2...Q2.1 etc.) and zip up all the folders into one zip file. Submit the code on Canvas.
- For CTF problems, **you must use the following format in your explanation**:
  - CTF Username
  - Flag
  - Explain the vulnerability in the program, and explain conceptually how that vulnerability can be exploited to get the flag.
  - How did you exploit the vulnerability? List the steps taken and the reasoning behind each step. The TA grading should be able to replicate the exploit following the steps. Feel free to make references to your code!
  - Append your source code in the same write-up. Your source code should be readable from the write-up PDF itself. Note that this does not count towards the page count above.

  Omitting any of the above sections would result in points being deducted.
- For Q2, no page restriction is in place, but we ask that you be concise, professional and neat in your writing.
- Please check your English. You won't be penalized for using incorrect grammar, but you will get penalized if we can't understand what you are writing.
- Proofs (including mathematical proofs) get full credit. Statements without proof or argumentation get no credit.
- There is an old saying from one of my math teachers in college: "In math, anything that is only partially right is totally wrong." While I am not as loathe to give partial credit, please check your derivations.

- **This is not a group assignment. Feel free to discuss the assignment in general terms with other people, but the answers must be your own.** Our academic integrity policy strictly follows the current INI Student Handbook `http://www.ini.cmu.edu/current_students/handbook/`, section IV-C.

- Write a report using your favorite editor. **Only PDF submissions will be graded.**

- Submit to Gradescope a PDF file containing your explanations and your code files before 1:30pm Eastern on the due date. Late submissions incur penalties as described on the syllabus (first you use up grace credits, then you lose points).

- Post any clarifications or questions regarding this homework in Piazza.

- Good luck!

1 Baby's First Chrome Extension (50 points)

In this section, we will explore the different types of Chrome extensions and how they can behave, and potentially be exploited. We recommend that you take a look at the development guide at `https://developer.chrome.com/extensions` and reference some of the sample extensions provided by Chrome (available at `https://developer.chrome.com/extensions/samples`).

You might also want to install your new creations outside of your daily browser. We recommend installing Chromium (`https://chromium.woolyss.com/`) and doing your experiments there.

1.1 Hello World Browser Action Extension (10 points)

Write a simple browser action extension that upon clicking on the icon, displays a popup which has an input form to key in your name. Upon typing your name in the input element, it should display "Hello {name}!". Your browser extension should include an HTML file (containing the pop-up), a JavaScript file (to enable the interactive element described above), and a CSS file to style the form. Note that the JS and CSS should not be inlined within the HTML file. Document on the write-up what you did and submit your code on Canvas under the folder Q1.1 in the final zip file submission.

Hint: the following repository might help you: `https://developer.chrome.com/extensions/examples/api/browserAction/set_page_color.zip`

1.2 Content Scripts (7 points)

Modify the extension to create a content script that replaces all occurrences of the word "Chrome" with "Firefox" on the the pages in the Chrome developer extensions page, i.e. pages under the following directory `https://developer.chrome.com/extensions/`. In addition, change the chrome icon (where the id is "logo") to the Firefox icon. For full credit, you would have to include and use external libraries like jQuery in your content script. Document on the write-up what you did and submit your code on Canvas under the folder Q1.2 in the final zip file submission.

1.3 Content Scripts Part II (8 points)

Modify the above extension again, such that the browser action pop-up from Q1.1 now takes two strings: a string to match against, and a string that replaces the matched strings. Include a form handler that sends a message to a content script for the pages in `https://developer.chrome.com/extensions/`, which then handles the request and replaces all occurrences of the matches with the string to replaced by. Document on the write-up what you did and submit your code on Canvas under the folder Q1.3 in the final zip file submission.

1.4 Background Extension (10 points)

In a separate extension project, implement a browser extension that runs in the background and sends the geolocation to an external server every minute. You do not have to set up an external server, *but you should make sure the code works*.

In addition, something worth noting is that you are not able to hide your extension icon by default. Explain why that is a good policy to enforce from the standpoint of the common user. Document on the write-up what you did and submit your code on Canvas under the folder Q1.4 in the final zip file submission.

Hint: the following repository might help you: `https://developer.chrome.com/extensions/examples/api/browserAction/make_page_red.zip`

1.5 Universal XSS (15 points)

Use the CTF server from HW 1 to solve this problem. We have created our very own extension, that informs users when the titles of other tabs that they have become updated! However, there is a tiny vulnerability in the code. Can you use it to exploit visitors of your site that are running that extension? In particular, the visitor will have the extension and a tab which displays the flag. Read the content from the other tabs to win! Solve the Universal XSS problem on the 14828 CTF Server. Submit a write-up containing your CTF username following the guidelines explained earlier.

2 Privacy Destroying Extension (50 points + 20 BONUS points)

2.1 The Basics (50 Points)

In this section, we will be using knowledge from above to write a privacy destroying extension. In particular, we will write an extension that tracks the user's browsing behavior, and extract as much personal information as possible from the user (just from this extension alone).

The user agent *Chromium 80.0.3987.0 (r722234)* will install the extension (much like how you installed the extension while developing) and will enable all permissions that are required. The user agent is then guaranteed to perform the following actions:

1. Browse the web normally through the navigation bar (e.g. Reddit, YouTube, etc.)
2. Make web searches on Google
3. Log in to Facebook
4. Make payments at undisclosed vendor websites
5. Disclose personal information on registration web pages
6. Chat with friends on various web chat platforms

Note that you do not have to implement an extension that captures all of the above for full points. **To get full credit for this section, you extension must minimally meet the criteria below:**

1. Be able to record all browsing activity (URLs visited by the User Agent)
2. Be able to exfiltrate data to an external server (where the server writes to a comma separated value file containing the URLs of the site visited)

Deliverables

In your Canvas submission: You have to submit a *Dockerfile* OR *node.js* application for us to set up as the listening server for the exfiltration. You should also include a folder containing the code for your extension. You should also submit a README.md containing step by step instructions on how to run your setup. Finally, you MUST include a video showcasing the capabilities of your extension - in particular, you must show that your extension works for the base case (where the user browses a site and it is recorded in your server) and for the bonus cases, if you choose to do so. *(Windows users can use the Game bar to record the video, and OS X users can use Quicktime for screen recording.).* Submit all the above on Canvas under the folder Q2.1 in the final zip file submission.

In your Gradescope submission: Please submit a write-up detailing the method of tracking you have chosen, why you have chosen such a method, and how your extension interacts with the server to exfiltrate data.

Testing

One week before the deadline, we will release a test environment for you to test the extension on a subset of test cases we will run. Please use this test harness to verify that your extension works as expected!

2.2 Bonus (Up to 20 points)

For additional points, you may wish to expand on the capabilities of the extension from the minimal criteria. Here are some suggestions on the areas in which you may pursue in expanding your Chrome extension:

1. Exfiltrating sensitive form data, e.g. username, passwords
2. Keylogging/Mouse tracking
3. Clickjacking
4. Reading configuration
5. Stealing cookies
6. Exfiltrating OS information
7. Stealing data from other open tabs/windows
8. Accessing and exfiltrating localStorage
9. Taking screenshots of current activity
10. Stealing previous browser history
11. Accessing local files
12. Any other interesting things not listed here

You may also explore other areas of improvement, including reducing the set of permissions required to do the tracking, or alternate ways of performing the same exfiltration of data, and weighing the trade-offs of each method.

Grading

In your Canvas submission: Submit the code under the folder Q2.1. For the code that you wrote for Q2.1, commenting in the code where you made the bonus modifications would be helpful. Include a BONUS.md describing briefly what you did. Also include a screen recording of your extension improvements in action (if they extract more data than in Q2.1). *(note that if your changes break the functionality in Q2.1, you may include your new code in a separate folder Q2.2)*

In your Gradescope submission: Please submit a write-up detailing the modifications that you made, why you made those modifications, and how they improved over the original extension, and any other interesting technical details that you wish to share with us.

The amount of bonus points awarded will be based on effort, coverage, creativity, and technical difficulty. For example, a bonus project which just integrates the geolocation script from Q1 will be awarded less points compared to a project that explores something new.

However, do note that since this is a bonus, we have strict requirements on what you can use. In particular, you are **NOT ALLOWED** to use code from other existing Chrome browser exploitation projects that exists on the web. **You MUST write all code yourself for the bonus section**. We will be using MOSS at the end of the semester to verify that no plagiarism occurred, so make sure you take this seriously!