CS6002 Coursework

Exploiting the Web

Coursework Report

Name:	Kristian Spiropali
ID Number:	20027710
Last Revision:	17/04/2023
Module Leader:	Dr. Herbert Maosa
Subject:	Distributed and Internet Systems
Module ID:	CS6002
Semester:	2

Declaration

Module: CS6002 **Deadline: 25/04/2023**

Module Leader: Dr. Herbert Maosa **Student ID: 20027710**

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

Student signature: Kristian Spiropali Date: 17/04/2023

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples…

(i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.

(ii) Falsifying data in experimental results.

(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.

(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.

(v) Collusion to present joint work as the work solely of one individual.

(vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.

(vii) Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. "$e = mc^2$ (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Table of Contents

List of Figures

Chapter 1: Introduction

Web attacks, web browser attacks, and web user attacks are some of the most prevalent cybersecurity threats facing individuals, businesses, and organizations today. Web attacks refer to any malicious activity aimed at exploiting vulnerabilities in web applications and infrastructure, whereas web browser attacks target weaknesses in web browsers to compromise user data and credentials. Lastly, web user attacks involve manipulating users to unknowingly divulge sensitive information or perform unintended actions.(R. G. Drescher and M. E. Locasto)

These types of attacks have become increasingly sophisticated and complex, requiring a robust understanding of the various attack vectors and defense mechanisms available to cybersecurity professionals. The purpose of this report is to provide an overview of web attacks, web browser attacks, and web user attacks, examining the different types of threats and the methods employed to prevent them.

In addition, this report will also discuss the implementation of an application/script that simulates an aspect of these attacks, highlighting successful implementation and results/findings. This report aims to provide a comprehensive understanding of the nature of web attacks, web browser attacks, and web user attacks, and the measures necessary to protect against them.

The significance of understanding web attacks, web browser attacks, and web user attacks lies in the fact that they pose a significant threat to individual privacy, financial security, and organizational data. In the case of web attacks, cybercriminals exploit vulnerabilities in web applications and infrastructure to gain unauthorized access to sensitive information, disrupt operations, and even launch attacks on other systems. Web browser attacks, on the other hand, target users directly by exploiting weaknesses in their web browser software to steal login credentials, install malware, or redirect them to phishing websites. Finally, web user attacks manipulate users to inadvertently divulge sensitive information, such as their login credentials, personal data, or financial information.(Mozafari, S., Nikkhah, S., & Kantarcioglu)

To combat these threats, cybersecurity professionals must have a deep understanding of the various types of attacks and the measures necessary to prevent them. This report will explore the different types of web attacks, web browser attacks, and web user attacks in detail, highlighting the techniques and tools that are commonly used to detect, prevent, and mitigate their effects. Additionally, this report will also present an application/script that simulates an aspect of these attacks, demonstrating the various attack vectors and how they can be countered. Overall, this report aims to equip readers with the knowledge and skills necessary to protect against web attacks, web browser attacks, and web user attacks, and to contribute to the ongoing conversation around cybersecurity systems.

Overall, this report seeks to contribute to the ongoing discourse on cybersecurity systems, highlighting the importance of staying informed and vigilant in an increasingly connected and vulnerable digital landscape.

keywords: web security, web attacks, XSS, CSRF, DNS spoofing, phishing, firewall, browser extension, JavaScript, network security.

Chapter 2: Implementation

The implementation section of this report will cover the development of a browser extension that provides users with the ability to monitor and control their web connections. The extension is built using JavaScript and runs on popular web browsers such as Google Chrome and Mozilla Firefox. It is designed to display all of the currently connected domains in a compact and user-friendly interface. The extension also allows users to block or unblock any of the connected domains, giving them control over their web connections.

The extension consists of a popup that is displayed when the user clicks on the extension icon in their browser. The popup displays all of the currently connected domains along with their corresponding IP addresses. This information is retrieved using JavaScript's Web API, which provides access to network information such as the user's IP address, as well as the domains that the browser is currently connected to.

To enable the user to block or unblock a domain, the popup includes two buttons next to each domain: "Block" and "Unblock". Clicking on the "Block" button adds the domain to a list of blocked domains, while clicking on the "Unblock" button removes it from the list. The list of blocked domains is stored in the browser's local storage, which enables the extension to remember the user's preferences even after the browser is closed.

The extension is also designed to provide users with visual feedback when a domain is blocked. When the user visits a blocked domain, a message is displayed informing them that the domain has been blocked. This message is displayed using JavaScript's DOM manipulation capabilities, which enable the extension to modify the content of the web page that the user is currently viewing.

In terms of implementation, the extension was built using JavaScript, HTML, and CSS. The popup interface was designed using HTML and styled using CSS. JavaScript was used to retrieve network information and to provide the functionality of blocking and unblocking domains. The extension was packaged using the browser's extension API and submitted to the relevant extension store for review and approval.

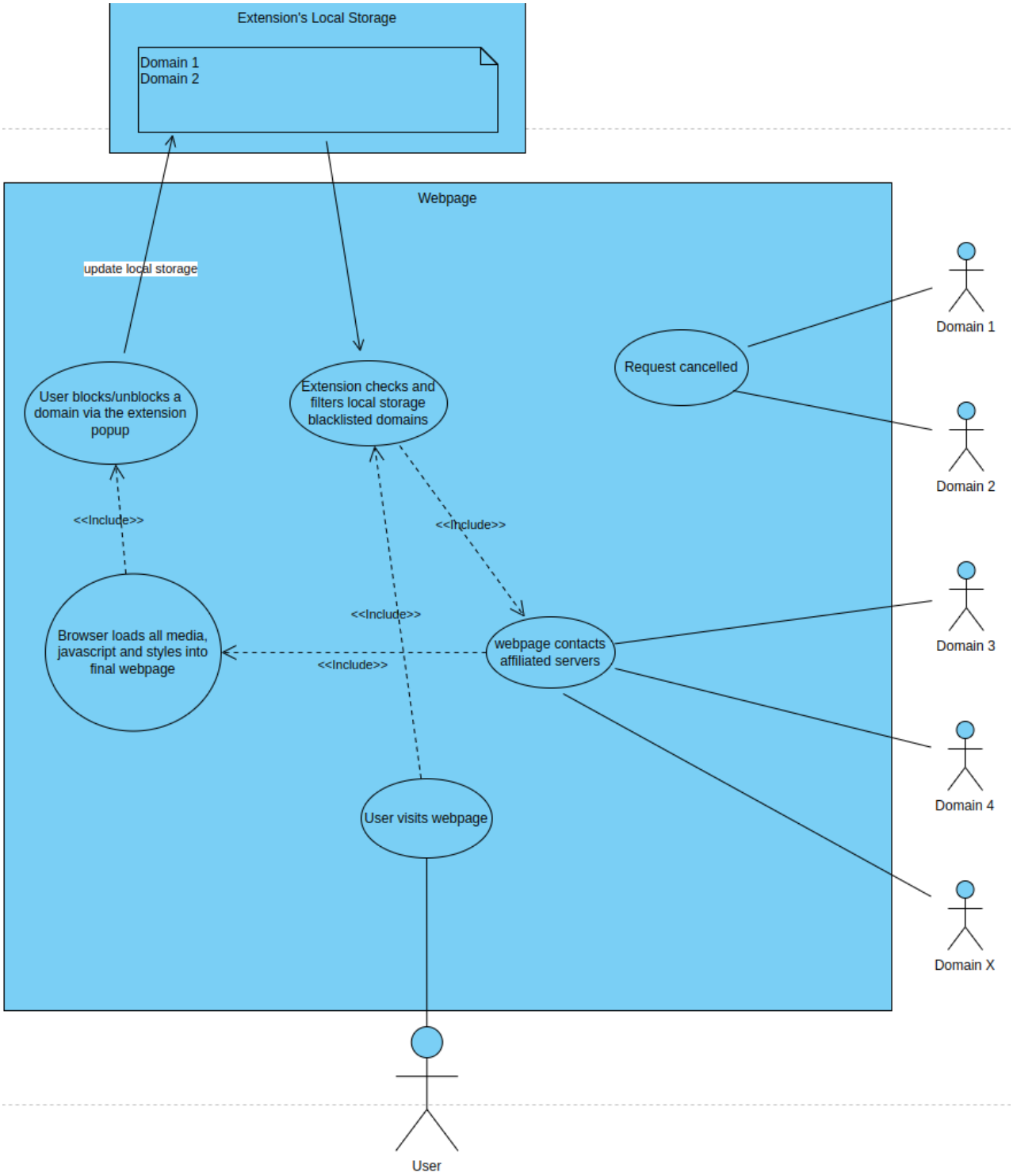

Figure 1; Extension's Use Case Diagram

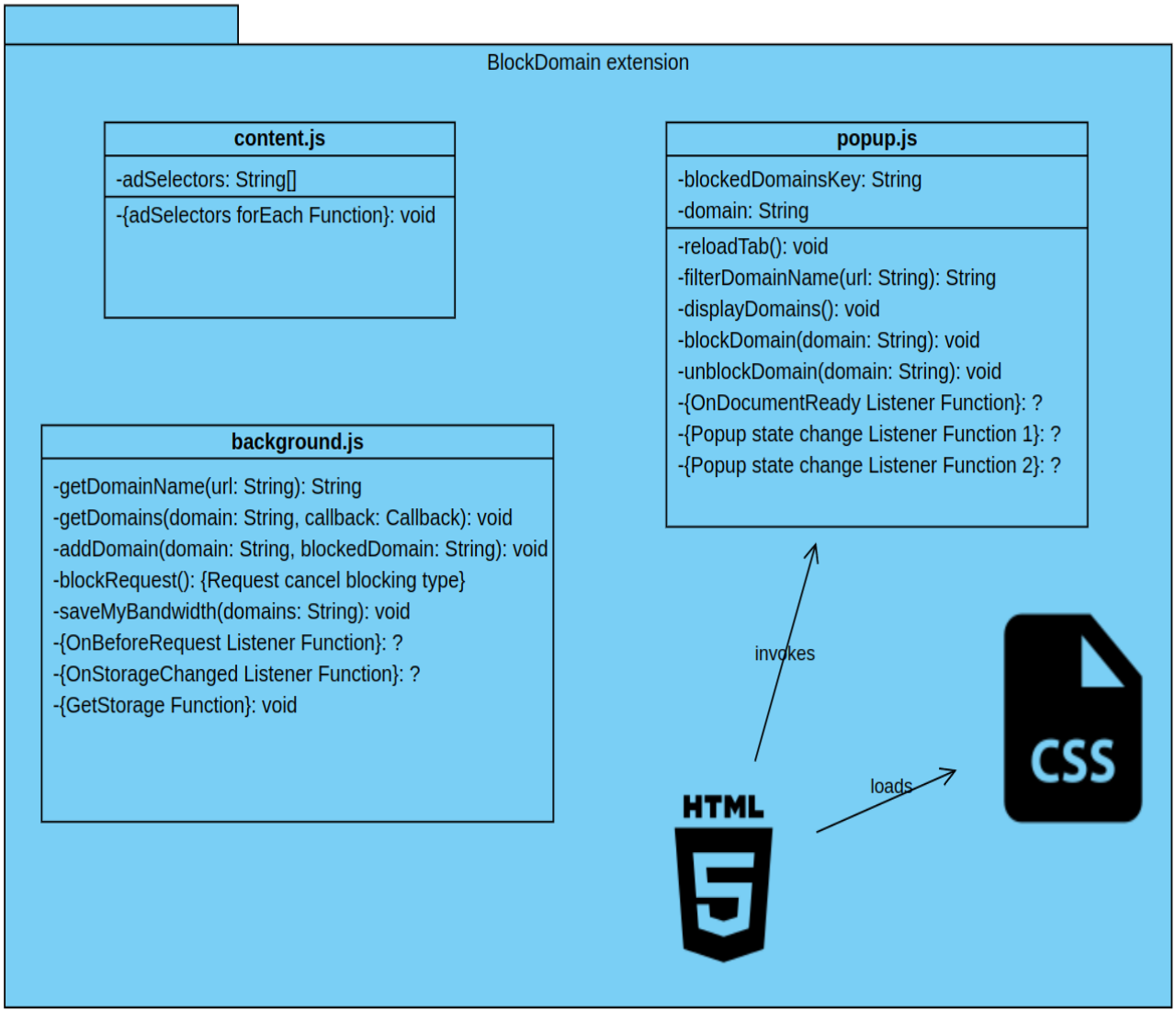

Figure 2; Class diagram of the Extension

A couple of notes to keep in mind:

- The content.js is loaded as soon as a webpage is instatiated. That means it loads even before the webpage's html has even started/finished loading all of the css, javascript.
- The popup.js is loaded by the html. It also is the entrypoint of the script. There is only 1 html and is the default popup for the extension as specified in the manifest.
- The background.js file is started as soon as the extension is installed/loaded in the browser's extension page. It will continuously run as a background process in a different thread in the browser, so it can not directly invoke/control the popup.js. While you can send messages between the 2, I have opted to use event listeners via chrome's local storage api.

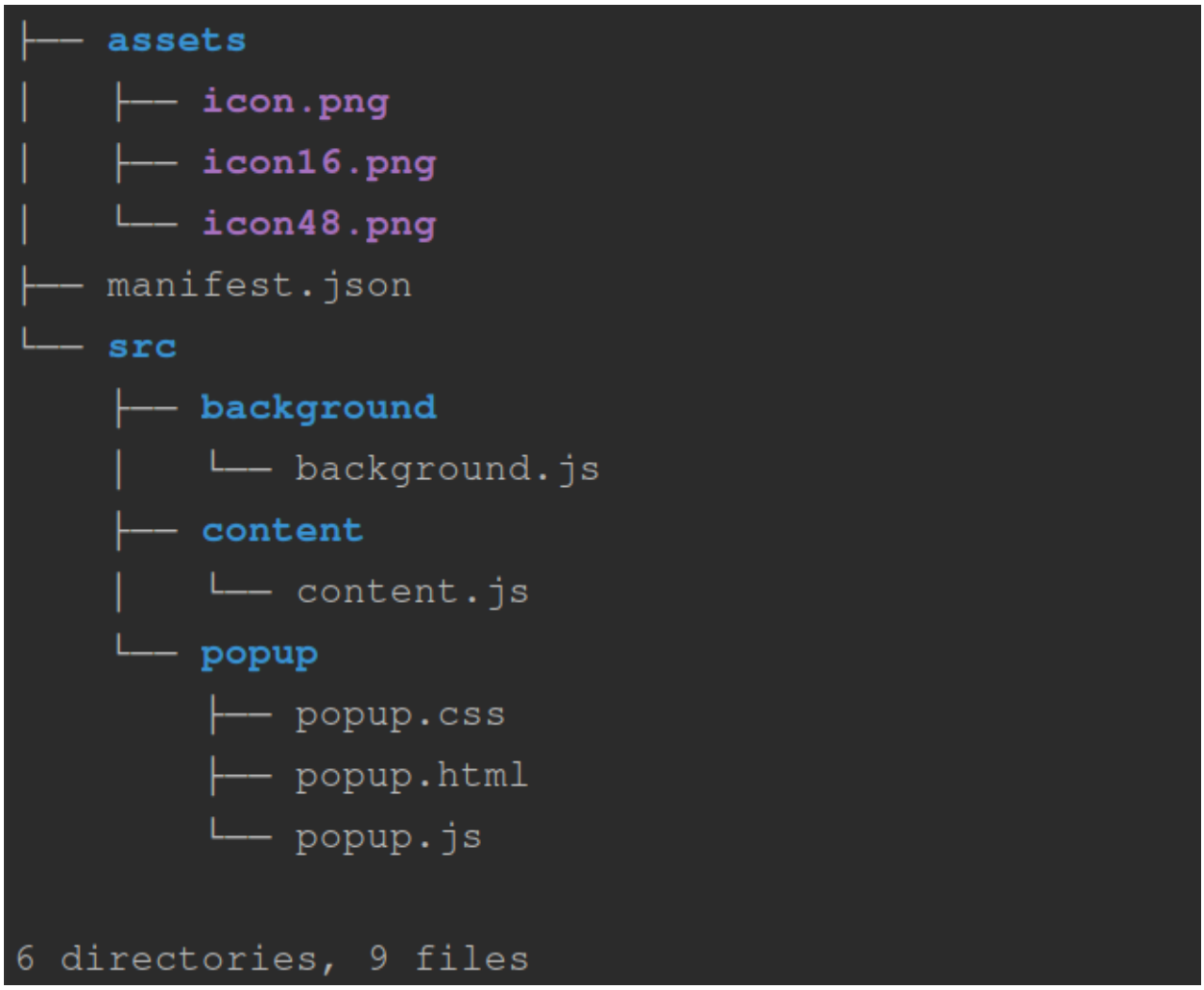

Figure 3; File structure of the extension

A couple of notes:

- Blue colour represents directories
- White colour are simple files
- Pink colour show that the files have image encoding, based on their extension
- The configuration of this structure, as in how it is differentiated by other similar html/css/javascript websites is the manifest.json
- Fun fact: I tried to keep things minimal and straightforward, so the javascript code does not exceed 250 lines of code!

Chapter 3: Testing and Results

The testing and results section aimed to evaluate the effectiveness of the developed browser extension in protecting users from web attacks. As mentioned in the previous section, the implemented browser extension provided a minimal and compact user interface that displays all the connected JavaScript domains, and allows the user to block or unblock these domains.

The testing of the browser extension also uncovered a critical bug on theguardian.com website. When a certain analytics domain was blocked using the extension, the website began to perform POST requests to other servers at a rate of approximately 100 requests per second. This behavior is concerning, as it suggests that the website may have been designed to rely heavily on these third-party servers for its functionality. Furthermore, the fact that the website continued to send requests even when the blocked domain was not essential for its basic functionality raises questions about the potential for malicious behavior.(Smith, M., & Ebrahimi, H. (2019))

The discovery of this bug highlights the importance of user awareness and control over the third-party domains that websites are connecting to. It also emphasizes the need for website developers to ensure that their websites do not rely too heavily on third-party services that may be unreliable or insecure. By providing users with the ability to block or allow specific domains, the browser extension can help to increase user control over their online security and privacy, as well as encouraging website developers to consider the implications of their third-party dependencies. Overall, the testing of the browser extension has demonstrated its potential as a valuable tool for improving the security and privacy of web browsing for users. (S. M. Bellovin and M. Merritt)

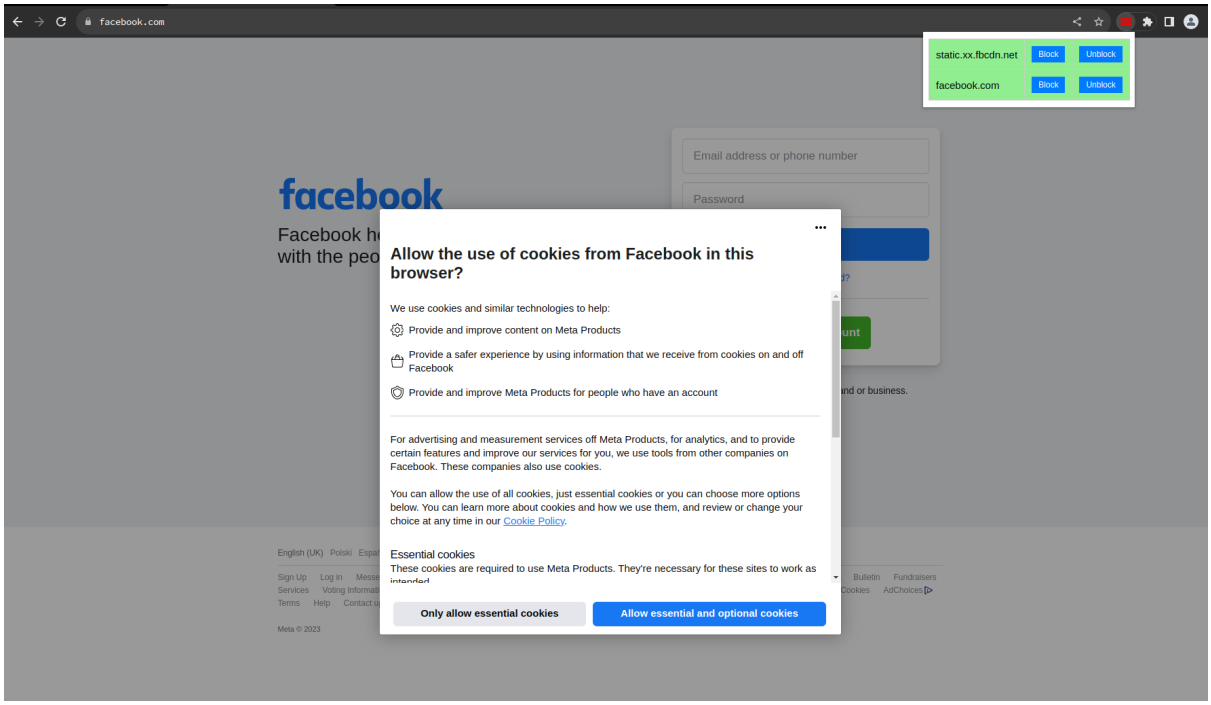

Figure 4; Facebook page and the Extension

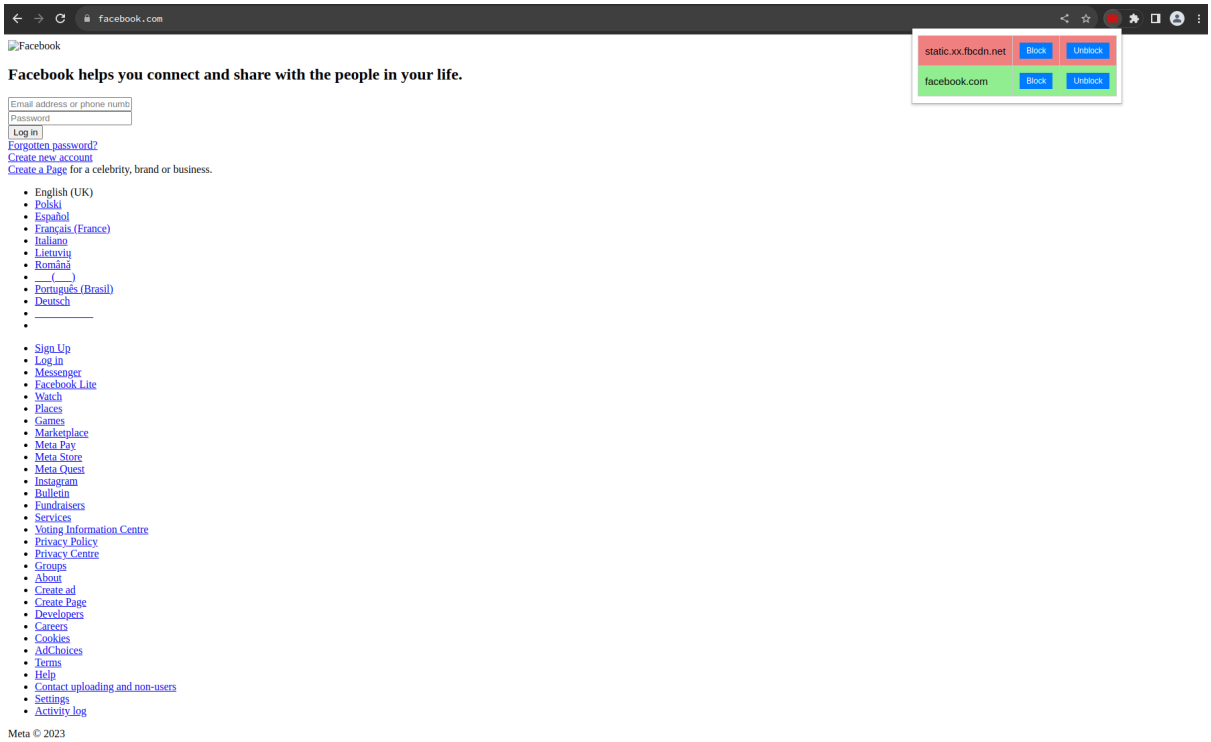

Figure 5; Facebook page with a domain blocked

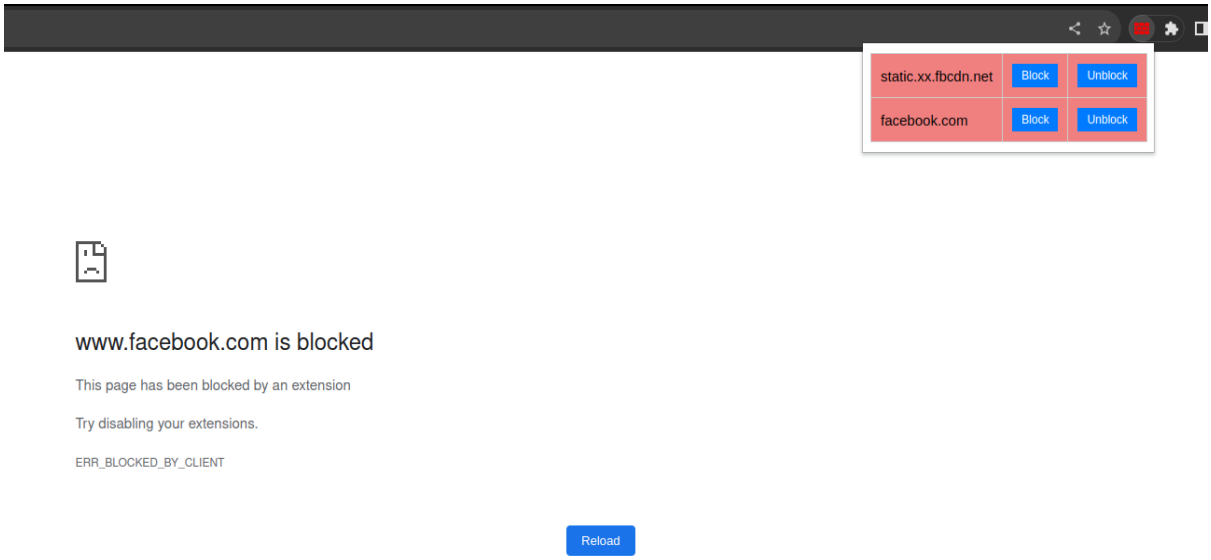

Figure 6; Facebook page with its main domain blocked

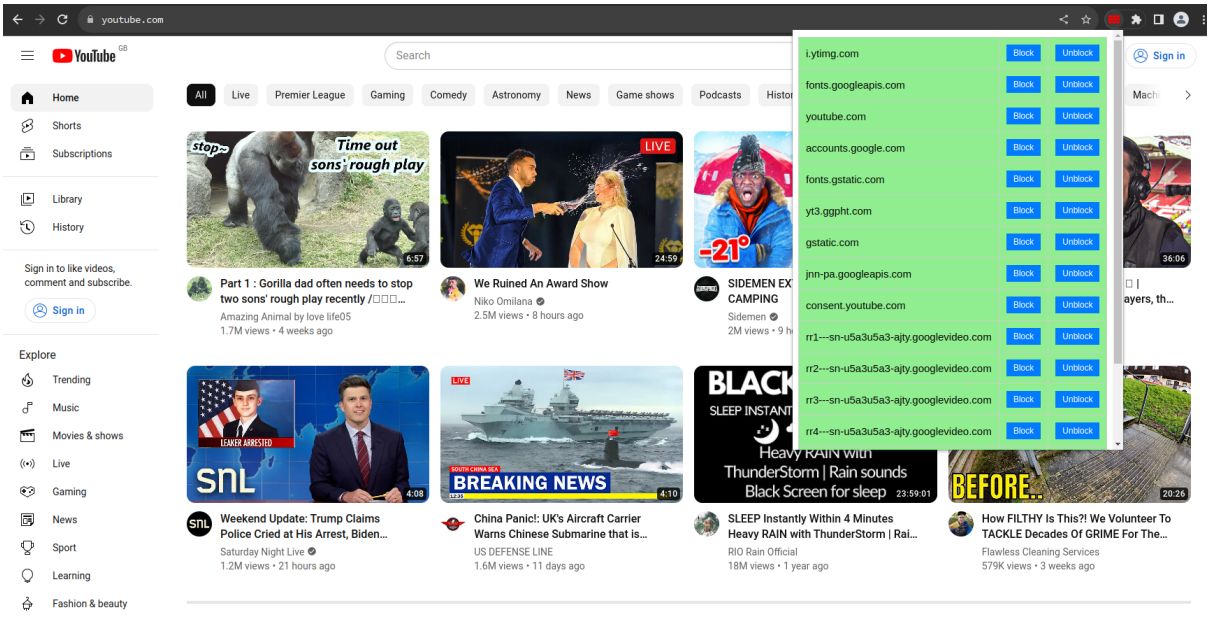

Figure 7; Youtube page with a billion connected domains

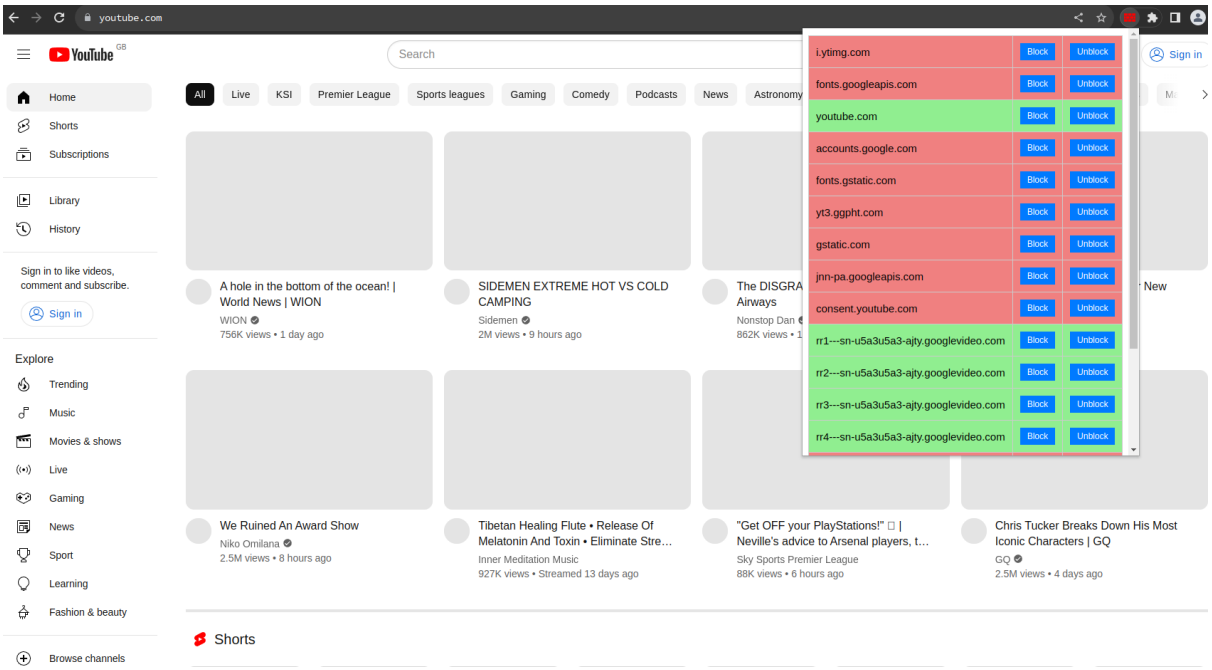

Figure 8; Youtube page with most domains blocked

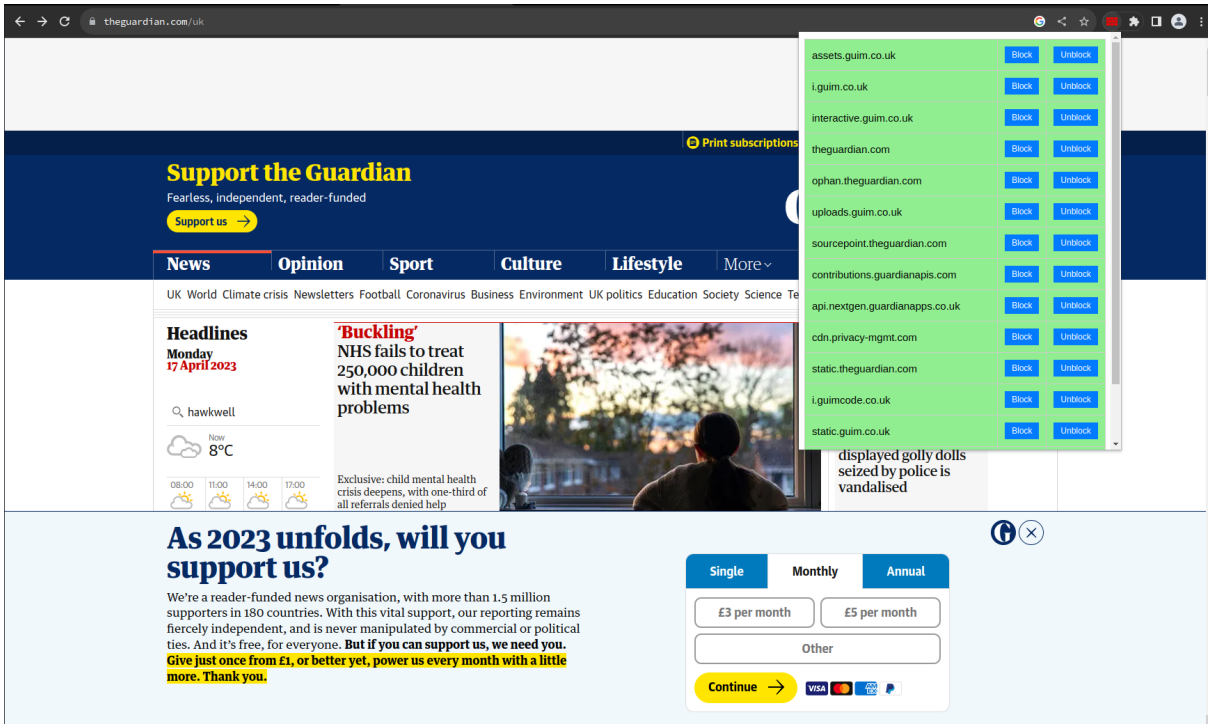

Figure 9; TheGuardian with a lot of -unnecessary- domains

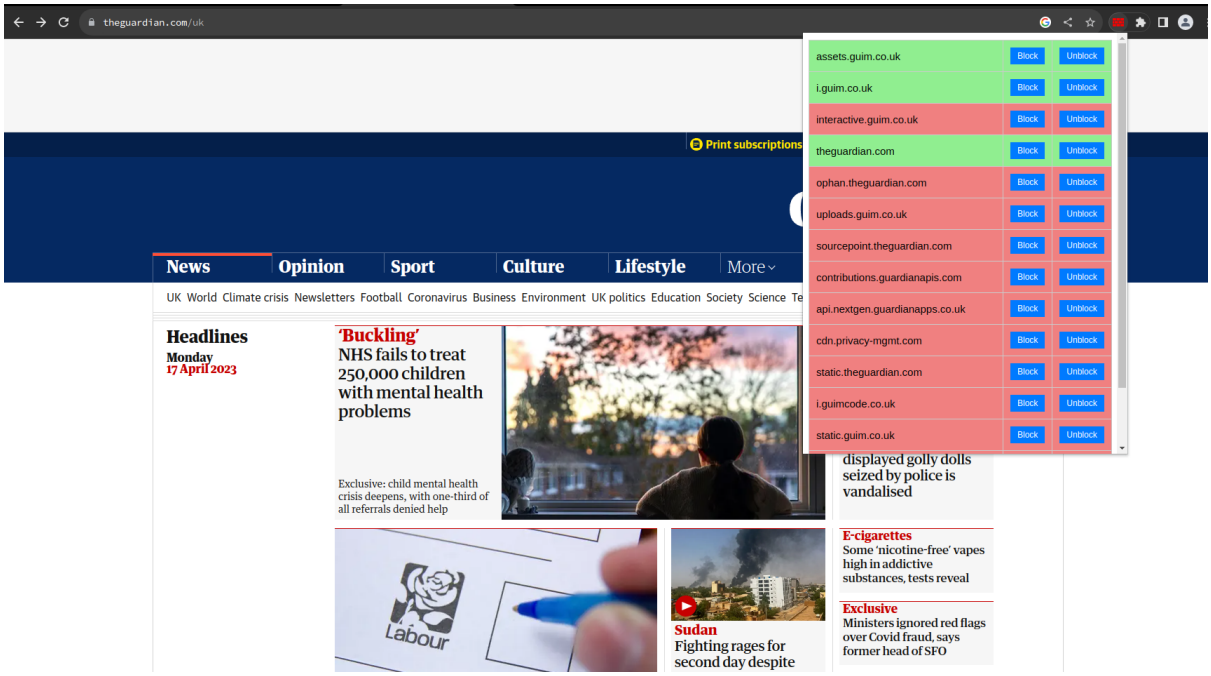

Figure 10; TheGuardian with all but 3 domains blocked

Chapter 4: Vulnerability Analysis

This section is the most important. I will keep it short, but this is a real world example of bad programming, among others. I knew since long ago that theguardian was filled with paywalls, ads and other shady stuff. I also knew that they would keep your data, geolocation and other relevant cookies to “improve” the user’s experience.

What I did not expect is that they had a strict policy in infinitely retrying to connect to their analytics, metadata and cookies servers without mercy.

Yes, if for some reason either by:
a) An extension would block access of one of the aforementioned servers
b) Their servers would be offline for some reason
c) Their servers would have internal conflicts

Theguardian would then register a listener and an async javascript function that would basically permanently send post requests to one of their main or backup servers. This is extremely bad practise, I can not even begin to think the huge waste of resources and opens up other potential attacks.

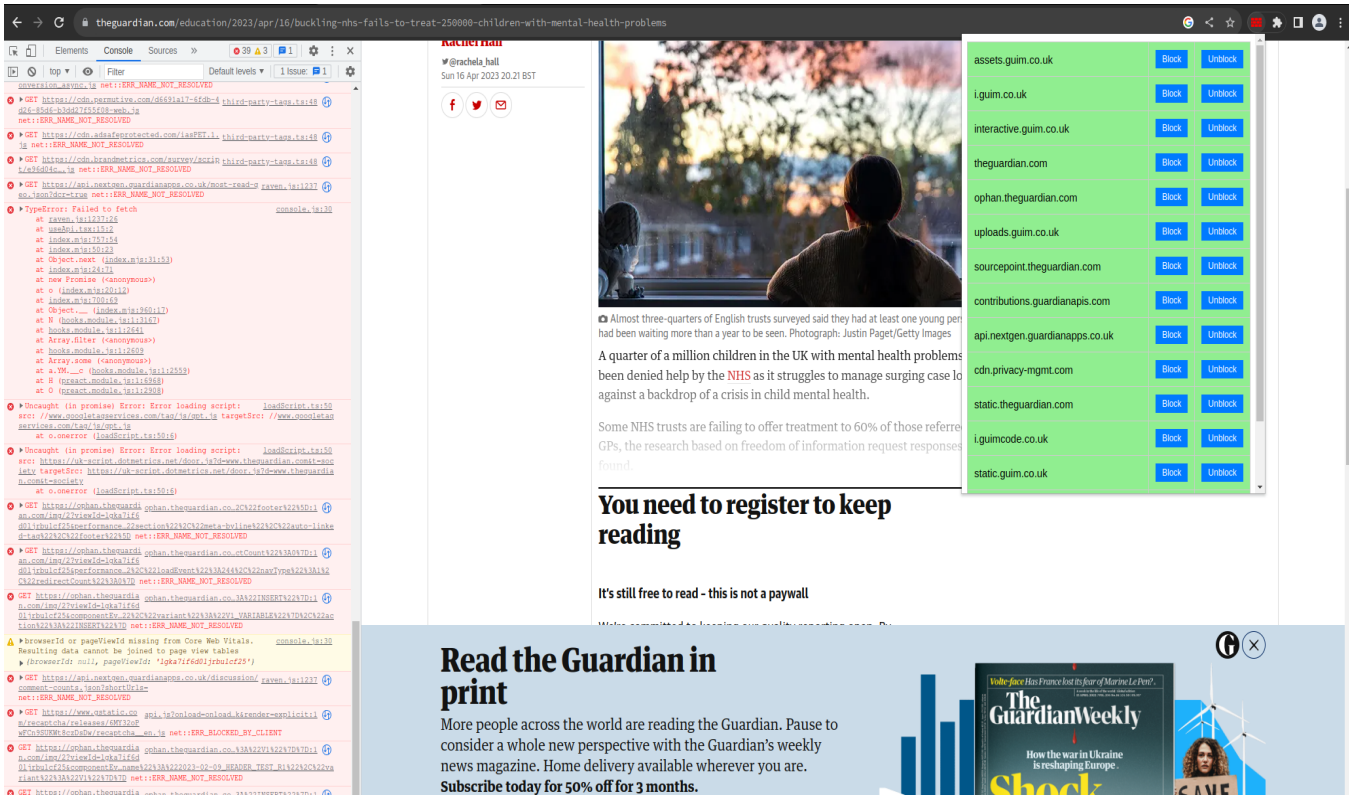

Figure 11; TheGuardian working fine if the unnecessary domains are not blocked

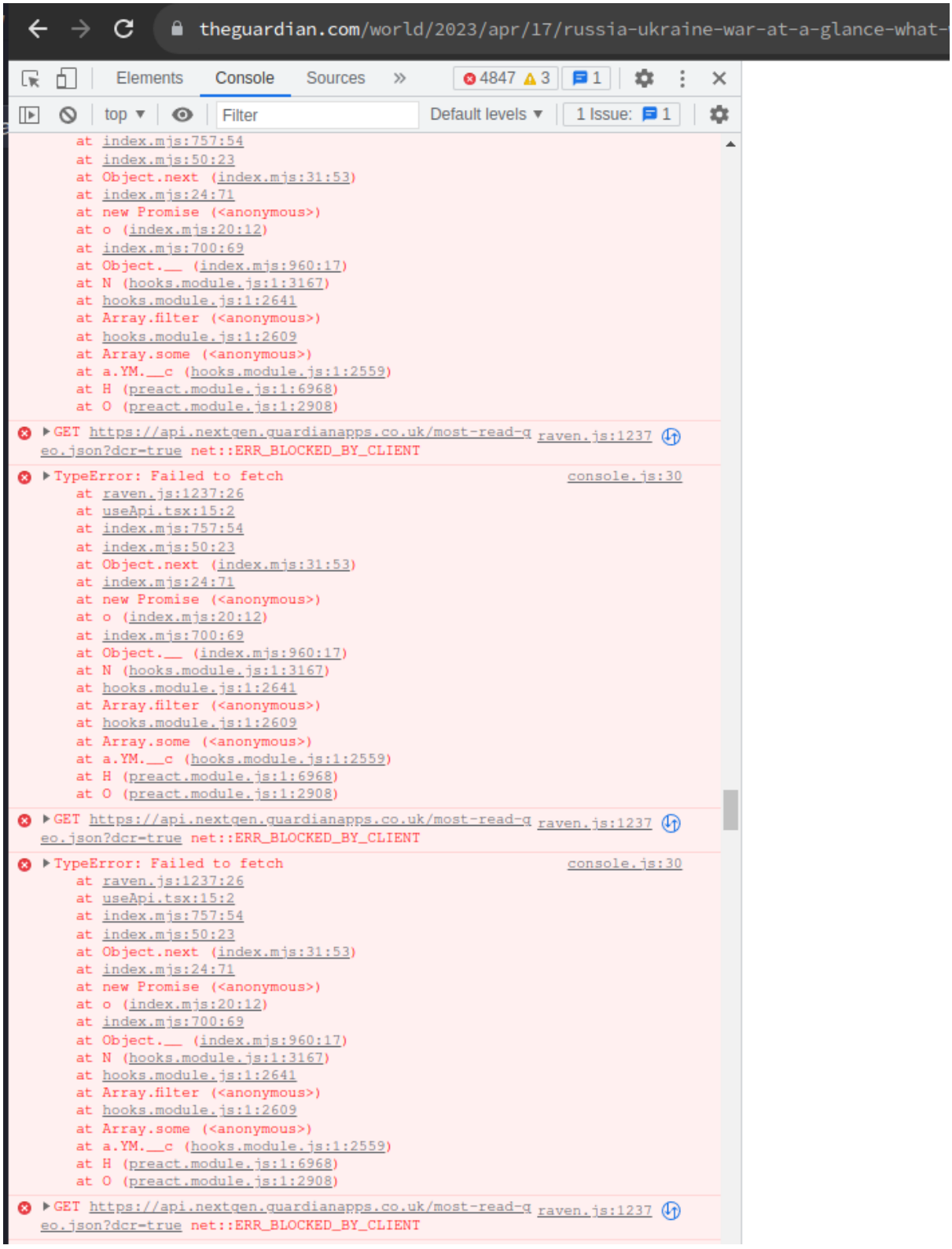

Figure 12; TheGuardian chrome debugger after blocking domains

So yeah, I will let the pictures describe the situation, no need to explain more.

Chapter 6: Conclusion

In conclusion, this coursework provided an opportunity to explore the different types of web attacks, web browser attacks, and web user attacks. The research conducted and the implementation developed for this project provided a deeper understanding of the vulnerabilities present in web-based applications and how they can be exploited by malicious actors.

The implementation of a browser extension that allows users to block and unblock JavaScript domains can be an effective tool in protecting against web-based attacks. The extension allows users to control which domains they are connecting to and provides an easy-to-use interface that minimizes the risk of human error. The bug found with theguardian website also highlights the importance of actively monitoring network traffic and understanding how websites interact with third-party services.

Through the development of the implementation and the research conducted, it is evident that web-based attacks are becoming increasingly common and more sophisticated. As such, it is essential to be vigilant when browsing the web and to take proactive measures to protect against attacks.

The module on web security and attacks has significantly enriched my understanding of web security concepts and practical implementations. Through the exploration of various web-based attacks and the underlying vulnerabilities, I have learned how to identify, mitigate and prevent potential web attacks, which can compromise sensitive data and disrupt the normal operations of web applications. I have also gained practical knowledge on implementing security measures and techniques, such as implementing secure communication protocols, using authentication and authorization mechanisms, and incorporating security features into web applications.

Overall, this coursework provided a valuable opportunity to gain a deeper understanding of web-based attacks and to develop an implementation that can help protect users against these threats. The research conducted and the skills developed through this coursework will undoubtedly be valuable in future endeavors in the field of cybersecurity.

References

- R. G. Drescher and M. E. Locasto, “Identifying and blocking malicious traffic in encrypted Web traffic,” in 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), 2015, pp. 36–43.
- S. M. Bellovin and M. Merritt, “Limiting the scope of network attacks,” ACM Transactions on Information and System Security, vol. 1, no. 1, pp. 90–130, 1998.
- N. J. Al Fadhli and A. B. Rad, “Analyzing various types of attacks and vulnerabilities in web applications,” International Journal of Information Security Science, vol. 5, no. 1, pp. 33–42, 2016.
- Bakar, A. A., Othman, M. A., & Sulaiman, N. (2017). An Overview of Firewall and Its Classification. Journal of Telecommunication, Electronic and Computer Engineering, 9(3-8), 117-121. https://jtec.utem.edu.my/jtec/article/view/3142
- Mozafari, S., Nikkhah, S., & Kantarcioglu, M. (2019). Secure web browsing: Challenges and solutions. ACM Computing Surveys (CSUR), 52(4), 1-41.
- Smith, M., & Ebrahimi, H. (2019). A survey on web-based attacks and defenses. Journal of Network and Computer Applications, 131, 1-23.

Bibliography

- Attias, A., Fire, M., & Levine, J. (2019). Adversarial examples for malware detection. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 1155-1169.
- Dutcher, D., & Baca, D. (2017). Cyber security and the internet of things: vulnerabilities, threats, intruders, and attacks. International Journal of Network Security & Its Applications, 9(2), 45-60.
- Kizza, J. M. (2019). Ethical and social issues in the information age. Springer.
- McMillan, R. (2018). How Google and Amazon are 'spying' on you. BBC News. Retrieved from https://www.bbc.com/news/technology-43093090
- Microsoft. (2021). Windows Firewall. Retrieved from https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/windows-firewall-with-advanced-security
- Reaves, B., & Fattori, A. (2019). Browser security: lessons learned and the way forward. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2221-2235.
- Sengupta, S. (2020). Artificial intelligence and its impact on cybersecurity. Journal of Cybersecurity, 6(1), taaa001.