18636 Browser Security Project Ideas for F24

Browser and extensions

1. **Testing (mobile) Browser implementation of security features.**
   Browsers enforce policies such as SOP and samesite cookies to protect web applications. However, different browsers implement these policies differently and sometimes even erroneously. The goal of this project is to adapt ideas from prior papers and set up an evaluation environment to test browser implementations against policy specifications, or each other. New methodologies are welcome, but not required.
   References:
   - **a.** Web Platform Threats: Automated Detection of Web Security Issues with WPT. USENIX 2024.
   - **b.** Same-Origin Policy: Evaluation in Modern Browsers. USENIX 2017

2. **Status of browser protection mechanisms of mobile browsers**
   Researchers have shown that there are discrepancies in how mobile browsers and desktop browsers implement browser policies. The goal of this project is to design and perform tests to understand these differences and their implications. It is also acceptable to study the support of security mechanisms on mobile browsers alone without comparison to desktop.
   References:
   - a. Uncovering HTTP Header Inconsistencies and the Impact on Desktop/Mobile Websites. WWW 2018
   - b. A Tale of Two Headers: A Formal Analysis of Inconsistent Click-Jacking Protection on the Web. USENIX Security 2020

3. **Run browser fuzzing tool.**
   Browser vendors have been continuously running fuzzing tools to identify browser bugs. The goal of this project is to understand to what extent bugs were found and fixed through the fuzzing efforts and to gain hands-on experience of setting up and running one of the fuzzing tools. If you need resources to run the fuzzer, please contact the instructor for help.
   - **a.** **https://hacks.mozilla.org/2021/02/browser-fuzzing-at-mozilla/**
   - **b.** **https://firefox-source-docs.mozilla.org/tools/fuzzing/index.html**
   - **c.** **https://github.com/googleprojectzero/domato**
   - **d.** **https://chromium.googlesource.com/chromium/src/+/main/testing/libfuzzer/README.md**
   - **e.** **https://github.com/ChijinZ/SaGe-Browser-Fuzzer**

4. **Investigating Privacy Practices of Browser Extensions**

   Similar to websites, browser extensions can also collect users' data and therefore should follow relevant privacy regulations such as GDPR. The goal of this project is to analyze the privacy practices of browser extensions, including but not limited to their privacy policies (or the lack of); their access to either their own cookies or modifying the host cookie store and HTML5 web storage; and whether they follow GDPR cookie consent.

   Reference:

   a. Detection of Inconsistencies in Privacy Practices of Browser Extensions (IEEE S&P '23)

5. **Identifying malicious browser extensions**

   Browser extensions have access to APIs that could harm the users, such as stripping security headers or stealing secrets. The goal of this project is to analyze malicious browser extensions through various metrics, such as permissions request, updates, and network requests. There are several existing papers that analyze extensions. Students are welcome to use open-source resources from these papers to analyze extensions.

   Reference:

   a. You've Changed: Detecting Malicious Browser Extensions through their Update Deltas (CCS '20)

   b. Helping or Hindering? How Browser Extensions Undermine Security. (CCS '22)

   c. Arcanum: Detecting and Evaluating the Privacy Risks of Browser Extensions on Web Pages and Web Content (Usenix 2024). Run the tool and analyze new extensions.

   d. CoCo: Efficient Browser Extension Vulnerability Detection Via Coverage-guided, Concurrent Abstract Interpretation (CCS 2023). Consider re-run the tool and analyze additional extensions

   e. Extending a hand to attacker: Browser Privilege Escalation Attacks via Extensions USENIX 2023**.**

6. **DOM-XSS protection in browsers.**

   DOM-based XSS is one of the most common vulnerabilities in web applications. Browser vendors have been trying to improve the browser to mitigate risks posed by such vulnerabilities. For example, Google Chrome has rolled out support for Trusted Types. The goal of this project is to study DOM-XSS in depth and understand what browsers do to help mitigate the vulnerabilities and where they fall short.

   References:

   a. https://owasp.org/www-community/attacks/DOM_Based_XSS

   b. https://www.w3.org/TR/trusted-types/

c. https://chromium.googlesource.com/chromium/src.git/+/HEAD/docs/trusted_types_on_webui.md

Web security and privacy

7. **Websites cookie preference compliance**
   The cookie consent-policy interface is implemented by the website to which it is meant to apply, however, there is no mechanism to guarantee that the website respects the users' preferences. Prior work has shown that such preferences are ignored by some websites. The goal of this project is for students to perform a quantitative analysis of websites' compliant of users' preferences.
   References:
   a. Celestin Matte, Nataliia Bielova, and Cristiana Santos. Multiple purposes, multiple problems: A user study of consent dialogs after GDPR. In 2020 IEEE Symposium on Security and Privacy (SP), 2020.
   b. Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User tracking in the post-cookie era: How websites bypass GDPR consent to track users. In Proceedings of the Web Conference 2021, WWW '21, 2021.
   c. Iskander Sanchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. Can I opt out yet? GDPR and the global illusion of cookie control. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, Asia CCS '19, 2019.

8. **Investigating Privacy Practices of Progressive Web Applications (PWAs)**
   PWAs are gaining popularity. They look like a native app, even though it is a web application running inside the browser. Because they are essentially web applications, how they collect and store users' data is also governed by policies such as GDPR. The goal of this project is to examine the privacy practices of PWA to answer some of the following questions. Are they tracking the users using similar techniques employed by websites? Do they implement GDPR cookie banners properly like other webpages? Do they respect users' selections?
   https://web.dev/progressive-web-apps/.

9. **Understanding dark patterns:**
   Dark pattern refers to a "user interface" that tricks users into doing things that they would otherwise not do, for instance, giving all their private data to third party trackers. The goal of this project is to study dark patterns in the context of websites' privacy content interfaces. Students can either perform a quantitative analysis of websites to identify the prevalence of

such dark patterns in the wild or provide a comprehensive report about the history, current status, and future research, industry, and government actions of dark patterns.

References:

a. Dark patterns and the legal requirements of consent banners: An interaction criticism perspective. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21, 2021.

b. What makes a dark pattern... dark? In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, may 2021.

10. Can layout thrashing/flickering be harmful?

Browser needs to redraw the layout of the page when new elements are loaded or styles change. If not designed property, some webpages suffer from layout thrashing. It is a performance issue because pages may take longer to complete rendering. However, this could also lead to users clicking on content that they do not intend to, for example, when the user is about to click on a link of an article, the ad frame appears at the same location of the link, because of layout change, the user's click will be on the ad, instead of the intended link. The layout changes during page loading is particularly noticeable on a mobile device where the screen is much smaller. The goal of this project is to understand how layout thrashing happens, the solutions to it, and its impact on security. Students can also perform a measurement study to understand what fraction of the websites suffer from it and what are the precise causes and quantify their security implications.

References:

a. https://web.dev/avoid-large-complex-layouts-and-layout-thrashing/

b. https://developers.google.com/speed/docs/insights/browser-reflow

c. https://dev.to/gopal1996/understanding-reflow-and-repaint-in-the-browser-1jbg

11. Facebook tracking pixels

Researchers have shown that web developers are seeking ways to continue to track users even when browsers block third-party tracking cookies. In particular, facebook uses first-party tracking pixels to track user information. This project aims to study how first-party tracking, like facebook pixels work and their prevalence.

Reference:

a. Paschalis Bekos, Panagiotis Papadopoulos, Evangelos P.Markatos, and Nicolas Kourtellis. The hitchhiker's guide to facebook web tracking with invisible pixels and click ids. In Proceedings of the ACM Web Conference 2023, WWW '23.

12. **Compare (tracking) behavior of not-logged in pages and logged in pages**

There are numerous studies on website tracking behavior. However, none of those studies analyzed pages that have users logged into the websites. The reason is to improve the scalability of the study. The goal of this project is to quantify the differences of tracking

behavior between a website without user login and the same website with valid user login. One hypothesis to test is that more information is gathered when users are logged in (i.e., third-party trackers know more about the user). One stretch goal of this project is to develop an infrastructure to scale up such experiments. Below are papers that discuss methodologies for identifying login pages, which would be helpful for this project.

References:

a. Asuman Senol, Alisha Ukani, Dylan Cutler, and Igor Bilogrevic. 2024. The Double Edged Sword: Identifying Authentication Pages and their Fingerprinting Behavior. In Proceedings of the ACM Web Conference 2024 (WWW '24). Association for Computing Machinery, New York, NY, USA, 1690–1701. https://doi.org/10.1145/3589334.3645493
b. M. Ghasemisharif, C. Kanich and J. Polakis, "Towards Automated Auditing for Account and Session Management Flaws in Single Sign-On Deployments," *2022 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2022, pp. 1774-1790, doi: 10.1109/SP46214.2022.9833753
c. Kostas Drakonakis, Sotiris Ioannidis, and Jason Polakis. 2020. The Cookie Hunter: Automated Black-box Auditing for Web Authentication and Authorization Flaws. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20). Association for Computing Machinery, New York, NY, USA, 1953–1970. https://doi.org/10.1145/3372297.3417869
d. Jonker, Hugo & Karsch, Stefan & Krumnow, Benjamin & Sleegers, Marc. (2020). Shepherd: A Generic Approach to Automating Website Login. 10.14722/madweb.2020.23008

Self-proposed project

13. Self-proposed browser extensions for enhancing/analyzing web/browser security and privacy

You may choose to implement a browser extension of your choice, for instance, a Bitcoin wallet extension. In the proposal, please list the set of desired features and why it fits the goal of gaining a deeper understanding of web/browser security or infrastructure.

14. Experience with WebAssembly

WebAssembly (wasm) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable target for compilation of high-level languages like C/C++/Rust, enabling deployment on the web for client and server applications. The goal of this project is for you to gain in-depth knowledge of WebAssembly. In this project, you can choose to do any of the following: build a website that includes wasm modules, measure the performance differences between websites written in Javascript vs the same website built using

WebAssembly, or a detailed report on the pros and cons of WebAssembly. https://developer.mozilla.org/en-US/docs/WebAssembly

15. **Self-proposed projects on web and browser security**
    You can propose a project. Please send in a paragraph describing the scope of the project and explaining why it fits the goal of gaining a deeper understanding of web/browser security or infrastructure.