Side Channel Attacks in the Browser

Case for Support

Background

Side-Channel Attacks (SCA) are an important class of cryptanalytic techniques which target unintentional information leakage in order to disclose private data. In general, Side-Channel attacks monitor a device performing an interesting operation and try to find a correlation between the side-channel information and the internal state of the processing device, which is related to the secret parameters involved in the computation. Side-channel information can be obtained by monitoring variations in the execution time (*Timing Attacks*), the power consumption (*Power Analysis*), the electro-magnetic emission (*Electromagnetic Analysis - EMA*), or the state of the cache memory (*Cache Attacks*) [1]. The goal of an attacker in side-channel attacks is to either recover the secret parameters involved in the computation by targeting faults in the implementation of some cryptographic algorithm, or disclose private information about a user, such as their location or browsing history. Side-channel attacks exploit vulnerabilities in a specific implementation rather than targeting an abstract algorithm, making them less generic but more powerful [1].

The expansion of the Internet determined browser vendors to add more and more features to their software in order to offer richer, dynamic and more interactive web pages. The evolution of browser software not only improved the overall user experience but also opened the way for a new category of side-channel attacks known as browser attacks. These types of attacks target vulnerabilities in the browser software. Browser attack fall into two categories: traditional timing attacks which measure the time it takes the server to respond, and cache timing attacks which measure how long it takes to retrieve a resource from the browser's cache memory.

Browser attacks are a very powerful category of side-channel attacks, which require minimal set up and affect a wide range of users. In general, these types of attacks require a user to navigate to an untrusted website, which contains attacker-controlled content. The web page has a background script which collects information about the victim which can be used by the attacker to infer a user's location, browsing history, or even current browsing activity. The victims are usually unaware that the website is collecting private data about them. One might think that such attacks can be avoided by simply not visiting untrusted websites; however, researchers discovered that even high-profile websites are using similar techniques to collect information about their visitors[2]. For example, a health insurance provider might want to know if the user has previously visited any health related web pages, this can easily be achieved using a side-channel attack.

Vulnerabilities in browser software emerge every day, but they also disappear within weeks. This gives attackers a limited time frame in which they can exploit potential vulnerabilities in web browsers. Although, most browser vulnerabilities get patched immediately, there are side-channel attacks in the browser discovered over fifteen years ago which are still possible.

In 2000, Felten and Schneider[3] presented the first browser side-channel attack, where they were able to determine if the user has previously visited a web page. A few years later, Bortz et al.[4] used a similar technique to determine the size of a hidden file, which can disclose private information about the user.

Their work was extended by Van Goethem et al.[5], who used newer web development technologies in order to determine the size of a remote origin resource. Browser attacks are not only used to determine how much information a user has access to, in 2015 Jia et al.[6] demonstrated that it is possible to determine a user's location. Additionally, Oren et al.[7] showed that it is possible to disclose a user's current browsing activity by analysing the state of their browser's cache memory.

Motivation

The aforementioned papers present various techniques for disclosing private information about users, by either measuring the amount of data the users have access to or simply checking what web pages are in their browser's cache memory. Additionally, most of the papers present real world scenarios where the attacks can be used to de-anonymize a user; however, the attacks are presented from the perspective of a malicious user who is trying to gain access to private data.

The world of browser attacks is rapidly changing with new vulnerabilities being discovered and patched every week. In general, adversaries have a limited time frame in which they can exploit the newly found vulnerabilities.

The fast-paced world of browser attacks and the lack of real world applications of side-channel attacks motivated us to focus on the following research themes:

1. Discover new vulnerabilities in newer browser technologies. Analyse how newer web development technologies improve existing attacks and discover new vulnerabilities.
2. Find new applications for existing browser attacks. Explore how side-channel information obtained through browser attacks can be used to disclose a user's list of acquaintances on social networking websites.

Project Objectives

The aims of this research are to detect new vulnerabilities in existing and newly released web development technologies, and to discover new real world applications for existing browser attacks which can ease the work of security agencies. More specifically, we will focus on the following objectives:

1. To detect new vulnerabilities in newly released browser software and compare them to existing side-channel attacks in the browser.
2. To research how existing side-channel attacks can be used to disclose a user's network.
3. To build a model which disclosed a user's network using data provided by security companies.
4. To extend the previous model to work on real world social networks.

Workpackages

Workpackage 1: Discover new vulnerabilities in web browsers

Principal research objective:

Principal deliverables:

Workpackage 2: Disclosing a user's network

References

[1] François-Xavier Standaert. Introduction to side-channel attacks. In *Secure Integrated Circuits and Systems*, pages 27–42. Springer, 2010.

[2] Dongseok Jang, Ranjit Jhala, Sorin Lerner, and Hovav Shacham. An empirical study of privacy-violating information flows in javascript web applications. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 270–283. ACM, 2010.

[3] Edward W Felten and Michael A Schneider. Timing attacks on web privacy. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 25–32. ACM, 2000.

[4] Andrew Bortz and Dan Boneh. Exposing private information by timing web applications. In *Proceedings of the 16th international conference on World Wide Web*, pages 621–628. ACM, 2007.

[5] Tom Van Goethem, Wouter Joosen, and Nick Nikiforakis. The clock is still ticking: Timing attacks in the modern web. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1382–1393. ACM, 2015.

[6] Yaoqi Jia, Xinshu Dong, Zhenkai Liang, and Prateek Saxena. I know where you've been: Geo-inference attacks via the browser cache. *Internet Computing, IEEE*, 19(1):44–53, 2015.

[7] Yossef Oren, Vasileios P Kemerlis, Simha Sethumadhavan, and Angelos D Keromytis. The spy in the sandbox–practical cache attacks in javascript. *arXiv preprint arXiv:1502.07373*, 2015.

Budget

a one-page (maximum) Budget: a table/spreadsheet of expenditure

Justification For Resources

a one-page (maximum) Justification For Resources: essentially a written narrative on why you need the expenditure on each line-item in your budget

Impact Statement

a one-page (maximum) Impact Statement: a description of how you intend to ensure that your work makes a difference in the world, rather than sitting on a shelf gathering dust)

Workplan

and also a one-page (maximum) Workplan:typically a GANTT chart or similar diagram indicating the order in which workpackages are carried out)