Universität des Saarlandes
MI Fakultät für Mathematik und Informatik
Department of Computer Science

Bachelorthesis

Capabilities as a Solution against Tracking

submitted by

Tim Christmann

on January 01, 1970

Reviewers

Dr. Sven Bugiel

Noah Mauthe

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Saarbrücken, January 01, 1970,

(Tim Christmann)

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, January 01, 1970,

(Tim Christmann)

What is the dedication page?

Yes, you can

Abstract

Trusted Web Activities and Custom Tabs enable Android developers to seamlessly integrate web content into native applications, offering a powerful tool for features such as Single Sign-On and in-app monetization. However, as shown by HyTrack, this integration also introduces severe privacy risks by blurring the boundary between web and app contexts, allowing persistent cross-app tracking through the browser's shared cookie storage.

Adjust to "final" solution

In this work, we propose a novel framework based on capability-based access control to mitigate these risks. By issuing fine-grained security tokens, our framework limits the access of third-party libraries to browser state, without compromising core functionalities such as SSO or web-based UI components.

Adjust Results

We evaluate our solution against the threat model and methodology introduced in HyTrack. Preliminary results indicate that our framework is easy to integrate, preserves application behavior, and successfully blocks unauthorized cookie access across applications. In our tests, it prevented [X]% of third-party cookies from being shared, while maintaining [Y]% compatibility with existing third-party SDKs.

how to best test this?

maybe add some other number stuff: Cookies set, sent, types, permissions ...

Acknowledgements

Thanks Obama, Sven and Noah Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, Android applications have increasingly leveraged web content within their interfaces to enhance user experience and streamline features such as authentication and monetization. To enable this, developers often use Custom Tabs (CTs) and Trusted Web Activities (TWAs), technologies that provide seamless, browser-backed web integration while maintaining native-like performance and features. This approach allows web-based functionality like Single Sign-On (SSO), for example login via Facebook or embedded advertising, without forcing users to switch between app and browser.

However, these benefits come at a cost. CTs and TWAs share the browser's cookie storage across all apps, enabling continuity of web sessions—but also opening serious privacy vulnerabilities. Recent research by Wessels et al. introduced HyTrack, a novel tracking technique that exploits this shared browser state to persistently track users across different applications and the web, even surviving device changes, cookie clearing, or browser switching [1]. HyTrack works by embedding a third-party library into multiple unrelated apps. Each app, unaware of the library's true purpose, opens a CT or TWA to the same tracking domain. This domain sets a unique identifier in a cookie, stored in the browser's shared cookie jar. When another app using the same library loads content from the same domain, the cookie is sent, enabling the tracker to correlate activity across apps—and even into regular browser use. Due to Android's backup mechanisms, the tracking ID can be restored even after a factory reset, rendering it more persistent than traditional evercookies

Cite here or directly after name?

This thesis explores whether Capabilities, a fine-grained access control model, can be used to limit or prevent these privacy issues without breaking legitimate use cases of CTs and TWAs. Specifically, we aim to design and evaluate a framework that allows developers to retain the benefits of third-party libraries—such as SSO or monetization—without exposing users to invisible, cross-app tracking. The framework should be simple to

integrate, practical in real-world deployments, and minimize interference with established app workflows.

Chapter 2

Related Work

Explaining difference between stateful and stateless tracking here wrong place???
Cite same sources as HyTrack?

Tracking mechanisms are typically divided into two broad categories: stateful and stateless tracking.

Stateful tracking relies on storing unique identifiers on the client device, most commonly through cookies or local storage. When a user revisits a site or interacts with embedded third-party content across domains, these identifiers are sent along with requests, allowing persistent recognition. While straightforward and highly effective, stateful tracking has become increasingly restricted through browser policies (e.g., third-party cookie blocking) and mobile platform changes such as the ability to disable the Google Advertising ID (GAID) on Android.

Stateless tracking, also known as fingerprinting, infers a user's identity based on a combination of device-specific attributes. These can include screen dimensions, installed fonts, and even subtle hardware or rendering quirks. Although this method is harder to detect and block—since it does not rely on persistent storage—it is also inherently less reliable, as small system changes may alter the fingerprint and disrupt identification.

cite same source as HyTrack? + similar work on fingerprinting on mobile devices?

Despite mitigation efforts, stateful tracking techniques are now re-emerging in new contexts. A notable example is HyTrack [1], which demonstrates a novel cross-app and cross-web tracking technique in the Android ecosystem. HyTrack exploits the shared cookie storage between Custom Tabs and Trusted Web Activities (TWAs) to persistently track users across multiple applications and the browser, even surviving user efforts to reset or sanitize their environments. While HyTrack highlights a serious privacy vulnerability, no concrete mitigation has been proposed that balances privacy with the legitimate need for seamless web integration—such as Single Sign-On or ad delivery—within mobile apps.

add that Zimmeck et al. have shown existence of cross-device tracking?

Our work addresses this gap by proposing a capability-based access control framework for Android applications using CTs and TWAs. By issuing fine-grained tokens that restrict third-party libraries' access to shared browser state, we prevent cross-app cookie tracking without breaking essential functionality. In contrast to prior work that focuses on browser-side or user-driven defenses (e.g., partitioned storage or consent prompts), our approach provides developers with a practical and enforceable way to contain third-party behavior within application boundaries.

like Safari

current state when opening TWA

check if current idea works; adjust accordingly

Chapter 3

Methodology

Write the methodology after sync-up with Noah

Our evaluation is guided by the following hypotheses:

- **H1**: Our framework prevents unauthorized cookie reuse across applications using the same tracking domain (i.e., blocks HyTrack-style tracking).
- **H2**: It does not break legitimate web interactions within a single app, including login flows and session persistence.
- **H3**: It preserves the seamless user experience expected from CTs and TWAs.

Chapter 4

Evaluation

To assess the effectiveness of our proposed mitigation strategy, I adopt the three primary goals identified by the authors of HyTrack as essential for any viable defense:

1) **Support all features of the web platform:** The solution must allow applications to display fully functional web content, including support for cookies, JavaScript, and modern APIs.

2) **Preserve seamless integration:** The user experience must remain uninterrupted. This includes avoiding obtrusive permission dialogs and maintaining smooth transitions between native and web content.

3) **Enable controlled access to shared browser state:** While isolation is required to prevent cross-app tracking, legitimate scenarios such as Single Sign-On (SSO) should continue to work within the context of a single application.

These criteria reflect the fact that HyTrack exploits standard Android behavior—specifically, the shared browser state exposed through Custom Tabs and Trusted Web Activities—rather than relying on unauthorized access or system vulnerabilities. Therefore, naive approaches like disabling shared cookies entirely would break common use cases and are not acceptable.

To validate these hypotheses, I will build on the open-sourced measurement tooling and proof of concept applications provided by the authors of HyTrack. Specifically, I plan to:

- Replicate the original HyTrack experiments under controlled conditions using two unrelated Android apps that embed the tracking library.

- Instrument network traffic (e.g., using mitmproxy or Frida) to observe cookie setting and reuse across CTs and TWAs.
- Apply the mitigation framework and compare observed behavior against the baseline.

Use itemize again?

I will collect and analyze the following metrics:

- Number of tracking cookies set and sent across app boundaries.
- Detection of identifier reuse across apps.
- ...

add more

In doing so, I aim to demonstrate that the solution effectively blocks HyTrack's cross-app tracking channel while maintaining compatibility and usability.

optimally yes, but possible in practice?

Chapter 5

Schedule

I plan to complete the project in the following phases:

- **Phase 1 (Weeks a-b):** *Background Research and Literature Review*
  Study existing work on web and cross-app tracking, especially the HyTrack paper. Familiarize with Android's Custom Tabs (CTs), Trusted Web Activities (TWAs), and Chromium's cookie storage mechanisms. Reproduce the HyTrack proof of concept to establish a baseline for evaluation.
- **Phase 2 (Weeks c-d):** *Framework Design and Initial Implementation*
  Design the capability-based access control framework. Define requirements, threat model, and integration points. Begin implementation of a prototype mitigation layer in an Android environment.

  same threat model, how to integrate not sure yet
- **Phase 3 (Weeks e-f):** *Evaluation and Experimentation*
  Set up experimental infrastructure using HyTrack's open-source tooling. Measure the effectiveness of the proposed solution by replicating tracking scenarios and comparing cookie behaviors across test cases.
- **Phase 4 (Weeks g-h):** *Analysis and Refinement*
  Analyze experiment results. Identify any shortcomings and refine the framework. Evaluate performance and integration feasibility. Summarize findings in preparation for documentation.
- **Phase 5 (. . .):** *Thesis Writing and Finalization*
  Write the thesis document, including all sections: introduction, background, methodology, evaluation, and discussion. Prepare for final presentation and submission.
- **Phase x (Weeks . . .):** . . .

5.1 Risks, Impact and Mitigation

wait for Mauthe Feedback

At worst, the (capability-based) approach may not effectively prevent cross-app tracking in practice and indeed a complete redesign of the Custom Tab and Trusted Web Activity API is needed-as suggested by the authors of HyTrack. If this occurs, the thesis will pivot to a critical analysis of why the capability model falls short in this context. The focus will shift towards identifying structural barriers in the Android platform and recommending future improvements, such as .... .

figure this out

At much lower impact, the implementation may not be feasible within the given timeframe or due to unforseen technical challenges. In this case, development will focus on a minimal viable proof-of-concept that demonstrates core functionality and missing features will be discussed as future work.

what else can go wrong?

⋮

Chapter 6

Success Criteria

The following criteria define the scope of this thesis and provide measurable goals to assess its success.

6.1 Must-have criteria

- The solution must effectively prevent or significantly limit cross-app tracking via the shared browser state, while preserving the core functionality of Custom Tabs and Trusted Web Activities (e.g., Single Sign-On).
- It must adhere to the three primary goals outlined by HyTrack: (1) support all features of the Web platform , (2) do not break the seamless integration, and (3) make shared state available to the Custom Tab or Trusted Web Activity.

directly from HyTrack paper

- A working proof of concept must be implemented to demonstrate the feasibility of the approach and to reproduce and compare against the HyTrack attack methodology.
- The thesis must provide a thorough discussion of how capability-based access control contributes to mitigating the identified tracking risks, including analysis of potential trade-offs (e.g., ambient authority).

Because concept of Capability technically not needed for implmentation but makes solution nicer and helps to circumvent Ambient Authority.

6.2 May-have criteria

- Provide an analysis of the solution's impact in terms of performance, usability, integration effort for developers, and potential security limitations.

anything else?

- Evaluate the framework across multiple Chromium-based browsers to assess generalizability.

- Investigate whether enhancements to the Digital Asset Links (DAL) mechanism could further strengthen the solution or propose a good alternative.

If not already necessary in solution.

6.3 Must-not-have criteria

- The thesis will not produce a production-ready implementationk, as the focus is on proof-of-concept and feasibility.
- The solution is not intended to prevent all forms of cross-app tracking, particularly not against malicious developers who intentionally bypass protections. The goal is to protect well-meaning developers from inadvertently integrating tracking libraries.
- It will not attempt to mitigate stateless tracking methods such as fingerprinting, which are outside the scope of shared state via cookies.

Bibliography

[1] M. Wessels, S. Koch, J. Drescher, L. Bettels, D. Klein, and M. Johns, "Hytrack: Resurrectable and persistent tracking across android apps and the web," in *34th USENIX Security Symposium (USENIX Security 25)*. Seattle, WA: USENIX Association, Aug. 2025.

Additional Something

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Notes