Chromium: RWI privacy risk mitigation design

DRAFT - for **public** discussion

Rick Byers - last major update Mar 15, 2024

With government-issued digital credentials (“real-world identity”) being increasingly utilized on the web, we propose a flexible system to enable Chromium to continually optimize the balance of benefits and risks to users and the wider ecosystem. Key to this system are: 1) transparency which enables a data-informed public dialog and, 2) risk-modulated control for end-users.

Background

Credential issuers such as governments and relying party websites are increasingly moving to leverage digital credentials in authentication and age verification on the web. This is being implemented on top of existing web primitives (such as custom schemes), and hopefully also via a new browser API. We expect regulation such as eIDAS 2.0 to accelerate adoption of such techniques over the next 5 years, potentially becoming ubiquitous on the web. There are a variety of potential benefits to these approaches including more secure transactions, better child safety and improved privacy via a “credential holder” (wallet) model that can blind issuers from usage.

Of course, there is also an extensive public debate about the risks this all entails, including the risk of sensitive personal information being abused. In many cases (especially highly-regulated markets like the EU) we expect credential issuers will have effective policies about where and how their credentials can be used. Issuers choose the wallets they support and will only select wallets they trust to enforce their rules. But, as a platform, we need to think about cases where such policies don’t exist or are ineffective.

This document aims to outline a design for how Chromium will work to mitigate these risks over time for the scenarios where mitigations higher-up in the stack prove to be inadequate for maintaining Chromium’s promise to users for protecting their privacy online.

Properties of a good mitigation strategy

First and foremost we must acknowledge that this is a rapidly evolving space with a complex set of tradeoffs which nobody is going to be able to reason perfectly about apriori. As a result, any good mitigation strategy will be designed to adapt and **adjust the balance of trade offs over time** as we learn, and as civil society norms and government regulations evolve.

Since the tradeoffs at play have a broad set of stakeholders, Chromium’s system must be designed to **rely on trust signals from a diversity of stakeholders**. Browser vendors such as

Google should not put themselves in the privileged position of unilaterally deciding what is and isn't trustworthy but instead should work with governments, civil society organizations and the industry at large to determine a diverse mix of appropriate trust signals.

There will be no "one-size fits all" solution, and so Chromium's approach will need to be **flexible and customizable** as appropriate for different jurisdictions and user preferences. In particular, behavior and preferences of users may be crowdsourced in order to better predict the expected desires of users, while providing controls for users to modify default preferences.

Given the public dialog and breadth of stakeholders in this space, and the popularity of the Google Chrome web browser, it's valuable for Chrome to make limited **anonymized aggregated data available publicly** about how real-world identity information is being used in Chrome.

End users should always have **transparency and control** over how their information is used. The browser is the user's agent and always seeks to operate in their interest.

Users should be informed and cautioned with **UI signals appropriate for the level of risk** they (or society broadly) are facing. This is important to offer users greater protection when the risk is greater, but also to reduce the risk of habituation in low risk settings causing users to take an action they ultimately regret in a higher-risk setting. Additionally, adjusting the UI treatment in proportion to the risk creates an additional incentive for popular verifiers to work to reduce the risks of their most used scenarios.

Threat model

This is a non-exhaustive list of the sorts of threats we consider in and out of scope for the specific browser design being discussed below.

In-scope threats and potential threats

- Abusive verifiers / RPs
    - For example:
        - Web-sites attempting to coerce or trick users into providing identity information they don't want to supply. Whether for criminal (theft) or questionable business (ad targeting) purposes.
        - Web-sites which have been infected by malware which attempts to leverage user's trust in the website to steal their PII.
- RP behavior open to public debate
    - There's a wide variety of potential uses of identity information which users and governments may or may not consider legitimate. While it's not the role of browsers to determine which of these behaviors are acceptable and which aren't, browser design can contribute constructively to such debates.

    - For example:
        - Websites restricting access to certain information based on verified age (eg. sexual health)
        - Websites restricting access to services based on verified age (eg. social media sites)
        - Arbitrary websites requesting unique non resettable identifiers (such as drivers license number) for login / account creation.
- Potentially deceptive issuers
    - Wallets are being designed to give users a sense of confidence in their use of credentials, such as by keeping their online activity private from the credential issuer. We need to consider that some issuer may violate that expectation by attempting to bypass the protections of the independent credential holder model, whether by colluding with a verifier or on their own.
    - For example:
        - A credential issuer colluding with (or compelling) an RP in order to track where and when specific credentials were being used in scenarios where users reasonably expected this to be private, such as verifying their >=18 age on an adult-content website.

Out-of-scope threats

- Malicious browsers and operating systems
    - While malicious browsers and operating systems (especially malware on Windows) are threats of concern, it's not technically possible to meaningfully mitigate such threats through the design of the browser. Any such mitigation could just be undone by a malicious actor with control of the operating system.
    - Instead such threats should be mitigated at other layers such as by credential issuers and wallets in their choices of where to issue credentials and when to revoke them.
- Malicious wallet applications
    - Key to the design of the 3-party (issuer/holder/verifier) identity system is that users are free to select a wallet application that they trust. Wallet applications have full access to the user's credentials and so could arbitrarily misuse those credentials regardless of the design of any browser behavior.
    - Issuer certification of wallets will prove a key component of the overall privacy and security model, but is out-of-scope for the design of browser behavior.
    - The one exception is that as an extra layer of defense against buggy wallets, browsers should not share information with wallets until explicitly selected by the user.

Proposed design

What follows is a proposal for the design and initial prototype implementation in Chromium. The intent is to update this document as the design evolves.

Credential presentment risk engine

Chromium will rely on a set of evolving heuristics to assign a “credential presentment risk score” between 0 and 10 to any operation we can identify as likely requesting a real-world digital credential or derivative. For testing purposes, it will be possible for developers to override Chromium’s engine and return a fixed result. It’s important that these scores represent a strictly monotonic value because the strategy relies on incentivizing a reduction in risk score where possible (eg. encouraging the adoption of ZKP protocols). Here’s an **example of how risk scores might be assigned long-term**, but it’s expected that the details will evolve significantly over time:

Score	**Meaning**	**Examples (for illustration only)**
0	No risk	No identity-related operations performed at all
1	Verifiably anonymous (including from abusive issuers)	Zero-knowledge-proof-based age-assertion from a predetermined set of acceptable issuers. Issuers are blinded from usage while still being able to revoke stolen credentials.
3	Lower risk	Selective disclosure of privacy-preserving identity properties such as age or age-range with large k-anonymity (eg. representing a large jurisdiction like a US state). Trustworthy issuer openly committed to privacy preservation. Response is encrypted to the requester.
5	Moderate risk	Sharing cryptographically attested sensitive personal information (eg. unique ID number) with websites having some meaningful explicit trust signal. Eg. eIDAS certification for legitimate identity use, crowdsource signals of user trust such as high acceptance rate. Response is encrypted to the requester. Sharing pseudonymous / resettable information with arbitrary websites. Eg. signing into a site using a Google account created entirely with user-supplied data. Note that using a digital credential for this scenario would be a strict privacy improvement over the status quo of federated identity systems due to blinding the IDP on credential usage.
7	Unknown or arbitrarily high risk	Sharing cryptographically attested sensitive personal information with arbitrary websites lacking any trust signals.

		Widespread use of an identity presentment protocol which is opaque to or unsupportive by the browser. Lack of response encryption.
9	Predicted abuse	High confidence of criminal or highly deceptive activity.
10	Known abuse	Behaviors explicitly identified to be in violation of laws in multiple democratic jurisdictions. Eg. known phishing sites attempting to steal personal and financial information for criminal purposes.

For an **initial simplified prototype implementation** we will simply assign risk scores as follows:

Score	**Criteria**
3	Limited selective disclosure (eg. age verification with mdoc or SDJWT-VC)
5	Full identity shared with origin having an explicit trust signal and response encryption. Eg. eIDAS certification of legitimate identity use or browser maker partner in an experiment with contractual commitments around user privacy.
7	Any other usage lacking any trust or risk-reduction signal. Eg. arbitrary website requesting unique IDs with Digital Credential API. Once the Digital Credential API has been fully launched, we anticipate treating mdoc / and OpenID4VP custom scheme invocations as this level of privacy risk (not something the browser can effectively reason about)

Risk engine implementation

We have to decide what process to implement the risk engine in. The main options are:

1. Renderer process:
    - Pros: easiest to implement and safest for the rest of Chrome
    - Cons: means a renderer vulnerability could be exploited to circumvent the risk engine
2. Browser process:
    - Pros: Defends against renderer vulnerability
    - Cons: Can't do non-trivial parsing in C++ (Chromium's rule of two), may need to be written in a memory safe language (Rust or C++ verified to use safe buffer operations only, etc.).
3. Utility process or another dedicated sandboxed process:
    - Pros: Safest from both above attacks.
    - Cons: extra process hops overhead and lots of extra complexity. Almost certainly not worth the complexity.

For now we have chosen option #2 as adequate. The risk engine today just does a RegEx match and JSON parsing using ParseJsonIsolated which is designed for this purpose (eg. relying on Java on Android). If the parsing gets more complex we may want to re-evaluate.

Public Metrics

Chromium anonymous UseCounter and UKM metrics will be collected (with eligibility according to Chrome's privacy policy) to provide insight into the following questions:

- How common are identity presentment requests at the various risk levels?
- What fraction of these requests are done via the Digital Credential browser API (where we have some visibility into the status of the result), vs. opaque mechanics like custom URL schemes (which we cannot tell even the success/failure of).
- How often are users able to satisfy Digital Credential API requests (i.e. are presented with a credential selection screen, if available from the OS)?
- How often do users choose to satisfy such requests vs. cancel (if available from the OS)
    - Note that current Android design will not let us differentiate the reason for data not being returned. So we'll probably have to make due with combining the above two into a single metric "How often do users satisfy Digital Credential API requests".
- For each popular public origin with a high volume of such requests, what is the breakdown of each of the above?

To enable the above, the following data will be collected in UKM:

Trigger	**Parameters**
Page load (denominator for all below triggers)	Page URL* country
Opaque credential request (eg. custom scheme navigation)	Risk score Page URL* country
Digital Credential API invocation	Result, one of: - Success - User cancelation - No matching credentials - Other failure Risk score Page URL* country

*URLs collected only for eligible contexts

This data will be analyzed internally to Google and periodically summarized publicly, but also directly made available publicly via the UseCounter dashboard and CrUX data set (similarly to how push permission acceptance rate is exposed). In particular, we expect the following lines of analysis to be useful in the public debate about risks and benefits of online RWI presentment and welcome analysis by independent researchers:

- To what extent is RWI presentment being used in Chrome, and at what rate is it growing (overall risk level) relative to web browsing generally.
- What is the breakdown between lower risk and higher risk scenarios and are their growth rates similar?
- What fraction of this usage is done via the Digital Credential API, for which we can have greater visibility (required for all of the below)?
- How is credential availability changing over time
- When credentials are requested and available, how likely are users to accept such a request and is that changing over time?
- Which origins in particular are requesting credentials most often?
- How do different origins rank according to their user's willingness to share credentials, or share credential-derived information like age assertions. In particular, what are examples of popular sites which have low acceptance rates, and examples that have high acceptance rates.
- To what extent are their variations in availability and acceptance rate by country and how are these rates changing per-country.

Safety interstitials

As an initial prototype of an abuse mitigation technique, we will add two forms of interstitials which can be presented to the user based on the credential presentment risk score. These interstitials will initially be presented after the user interacts with the OS and wallet since that's when the user has the most context on whether they really want to complete the action or not.

Low risk interstitial

This interstitial is intended to warn users of elevated risk while guiding them towards potentially confirming the operation.

Dialog text	This website would like to ask Android to give you options for sharing your identity information.
Dialog options	OK, Cancel

High risk interstitial

This interstitial is intended to warn users of high risk while guiding them towards canceling the operation.

Dialog text	This website would like to ask Android for access to your identity. Chrome cannot determine what privacy risks this may pose.
Dialog options	Cancel, Continue Anyway

Interstitial selection criteria

For an initial prototype implementation, we will trigger interstitials as follows:

Triggering risk score	**Interstitial triggered**
5	Low risk interstitial
7	High risk interstitial

In the future we may relax the triggering in normal browsing modes while leaving the triggering relatively high in Incognito modes.

Using the metrics collected and other sources of feedback we will tweak this critical (along with the risk score definitions) as appropriate to strike a balanced tradeoff.

Some evidence for being more permissive (higher trigger):

- At the moment, usage of real-world identity online is very limited, mostly for testing purposes. Overall ecosystem risks are low.
- Options for reducing risk (such as the browser API and ZKP protocols) are still in development and it's not yet reasonable to require their usage.
- It's already common practice to implement credential presentment in ways the browser will always be totally blind to (such as sending a push notification to a native app, with all communication between the wallet and verifier occurring via a server). Using a lower trigger would incentivize more products in development down this path.

Arguments for being more conservative (lower trigger)

- We're just beginning to enter this area. It's safest to start conservatively and be prepared to relax as we learn.

Feedback channels

- Discussion on the definition and evaluation of risks is being co-ordinating by the W3C PING in this GitHub repository. New issues and pull requests are welcome.
- The design of a browser API is being coordinated by the W3C WICG in the Digital Credentials GitHub repository. New issues, slack and attendance at group meetings are welcome.

- Bug reports and feature requests for the Chromium implementation can be filed using this chromium bug template.
- Feedback and suggestions for this document in particular (outside any of the above) can be sent by email to the authors.