FIDO Privacy CA

Proposal

Adam Langley <agl@google.com>
Matt Braithwaite <mab@google.com>
Christiaan Brand <cbrand@google.com>
Alexei Czeskis <aczeskis@google.com>
Dirk Balfanz <balfanz@google.com>
(October 2017)

This document is intended to inform other webauthn parties about our plans and to encourage them to implement the changes to webauthn that's required to support this model.

The problem

Webauthn and FIDO supports the concept of attestation: the ability for a relying party to determine the provenance of the authenticator-to-be-registered. We postulate that many consumers services in reality don't care about provenance of authenticators especially when deployed as a second factor, since the primary threat model is scalable, remote attacks. However, in enterprise environments there are usually the need for a much stricter enforcement of authenticator types and as such it is important in these cases for the device to disclose information about itself to the RP.

In the current FIDO 1 world an RP would take the batch attestation certificate sent by the authenticator and query the FIDO MDS to determine the relevant attributes: whether the device has passed protocol compliance testing, and perhaps the relevant security level of the device. We want to make implementing attestation-checking easier, so that sites are more likely to do it correctly and the webauthn experience overall will be better for users.

The solution

Chrome intends to implement what we are calling a Privacy CA, but which might be called an "attestation proxy". During webauthn and U2F registrations the attestation certificate and signature from the token will be sent to a Privacy CA, along with the hash of the signed data. We are planning on following this same model for built-in Authenticators on Android too, even when registrations are performed by apps on the device.

The Privacy CA will:

1. Check the attestation signature, given the attestation certificate and signed data hash.
2. Check the certificate (or other attestation) against its local policies.

3. If the signature is valid and the certificate is recognised, it will return a new packed attestation certificate and signature of the same hash to the Chrome instance.

Chrome will pass the Privacy CA certificate to the calling Javascript so that the token appears to be attested by the Privacy CA. If the Privacy CA returns an error then Chrome will substitute a generic, meaningless attestation certificate for U2F and, in webauthn, potentially return a dummy attestation type.

The Privacy CA will support two levels of attestation: Basic and FIDO Security Certification Level *tbd* (hardware attestation + some code review coverage). In time, we intend for both of these to be defined by the MDS, which the Privacy CA will reload regularly. In exceptional circumstances (such as a security issue with a token that should be responded to immediately), or in order to bootstrap the system before the MDS is ready, we may augment the MDS data.

The two levels of attestation will be exposed as two different attestation roots.

The certificates from the Privacy CA, and the randomly generated certificates from Chrome, will copy the transport type extension from the token's certificate.

The Privacy CA will not learn of the sites that a user is registering with because it only receives the hash of the signed data, and that hash includes a random challenge which blinds the included rpID.

Enterprise cases

We do acknowledge that there are other relying parties out there that have an obligation to ensure that the authenticators they accept meet a certain security and usability bar, while not necessarily having control over the client platform. These relying parties rarely (if ever) have the need to uniquely identify authenticators or even authenticator vendors, but rather are interested in being able to tell whether an authenticator conforms to some minimum requirement.

Chrome has a mature enterprise policy system. A policy control will be added to allow a token's attestation certificate to be returned directly to the calling Javascript for whitelisted rpIDs.

This will obviously not apply to clients that don't have the enterprise policy installed, but we note that a token need only be *registered* on a configured client. It can then be used in other machines.

Retrospective unblinding of tokens

We understand that the ability to identify affected users when a security issue with a token is disclosed is desired. While issues with weak keys and bad key-handle construction can likely be identified without attestation information, some cases cannot be spotted that way.

In order to support this, Privacy CA certificates will contain an extension containing a series of 32-byte values. The first 16 bytes of each value will be random and the remaining 16 bytes will be the truncated HMAC-SHA256 of that random value under a key. The HMAC key will be specific to some property of the token's certificate, for example the certificate itself, the Issuer name, or perhaps the AAGUID. A given certificate can contain several such 32-byte values and thus may be identified by several different properties.

Google will maintain a public URL serving a JSON file containing the HMAC keys corresponding to token certificates that are linked to a known security issue. In this way, the Privacy CA can retrospectively unblind Privacy CA certificates.

For example, if a specific batch of tokens is found to be flawed, the HMAC key linked to the specific attestation certificate for that batch can be published. RPs that wish to take special measures to respond to the flaw can search their recorded attestation certificates and, for those from the Privacy CA, look for a value/HMAC pair that matches when using the published HMAC key.

Individual certificates

Since Chrome will have an enterprise policy control for direct attestation, it can expose that signal to the token in case the token should wish to use an individual attestation certificate in that situation.

Would ECDAA be a better choice than a privacy CA?

Firstly, ECDAA is currently moot as millions of U2F tokens are already deployed with batch certificates. We have to support them in any case.

Secondly, ECDAA is a smarter way to do batch attestation, but it still inherently exposes vendor and, likely, model, and so causes many of the same concerns as batch certificates.

Access to the Privacy CA

Expediency requires that Chrome's Privacy CA be run by Google, at least at first. We are open to other browsers using our Privacy CA should they so desire.

Requests from webauthn and FIDO

In priority order:

1. That the AAGUID be moved from the signed registration data to the token's attestation certificate.
2. That an option be provided at registration time for sites to indicate whether they "care" about attestation. If not, the Privacy CA round-trip can be omitted. PR is here.

3. That the option default to false, i.e. so that people implementing webauthn in the long-tail of sites and who will never care about attestation, get the correct behaviour by default.
4. That the browser be able to add a blinding value that's included in the signed registration data. (This eliminates the need for the RP's registration challenge to have enough entropy to blind the rpID from the Privacy CA.) This does not require a change to the API and is simply something the browser could do today.
5. A boolean in the CTAP2 registration message to indicate to tokens that individual attestation certificates may be used.