Co-Chairs of the ERPB PIS Identification SG
Working Group on Payment Initiation Services
Euro Retail Payments Board

10th May 2017

Secure Communication and Identification of PSPs

This is a letter from the Financial API Working Group at the OpenID Foundation. We are writing to you for the following reasons:

1. To raise awareness of international standards that could help your working group as it develops "detailed technical specifications" for an integrated payment initiation market.
2. To highlight the risks with some of the technical proposals that are being considered across Europe

The OpenID Foundation is a nonprofit international standardization organization that promotes OpenID Connect and related standards. Its members are key authors for many of the IETF standards relating to OAuth 2.0 and OpenID Connect.

The Financial API Working Group (FAPI WG) was proposed by Nat Sakimura (Nomura), Tony Nadalin (Microsoft), and Cindy Barker (Intuit) and was formed in mid-2016. Its main aim is to create a secure open standard for financial APIs and in so doing, facilitate a shift from "screen-scraping" to secure API based access to payment accounts.

The FAPI WG has an established liaison with other international standardisation bodies, for example X9 in the USA and is the process of establishing a formal liaison with ISO TC68.

I understand that the working group is mainly looking at the issue of identification of Third Party Providers (TPPs) to Account Servicing Payment Service Provider (ASPSPs). It is important to consider the Payment Service User (PSU) when considering solutions as this is a 3-way authentication and authorisation issue.

The Need for an Authorisation Standard

The European Banking Authority's Regulatory Technical Standards (RTS) requires ASPSPs to allow their PSU's to access data and initiate payments via registered TPPs. This involves authentication and authorisation between 3 parties:

1. The ASPSP must authenticate (securely identify) the TPP
2. The TPP should authenticate (securely identify) the ASPSP
3. The ASPSP must authenticate the PSU (Strong Customer Authentication - SCA)
4. The ASPSP must collect authorisation from the PSU for an action initiated via a TPP (SCA)
5. The ASPSP must generate an "authentication code" that the TPP can use to access account information or initiate a payment.

This 3 way communication is a common and solved problem - the most widely accepted and adopted standard for it is RFC6749 - OAuth 2.0.

OAuth 2.0 has already been adopted by the following Banks:

- BBVA
- AXA Banque
- Fidor Bank
- Monzo Bank
- Starling Bank
- Česká spořitelna
- Citi
- Capital One

OAuth 2.0 is well known by developers, it is an established international standard and it is supported by all major IAM and API Gateway vendors.

OAuth 2.0 however is a framework and can be used with different security profiles depending on the sensitivity of the API it is protecting. The FAPI WG has a profile for OAuth 2.0 that is designed for financial APIs and provides additional security and non-repudiation guarantees.

OAuth 2.0 can also work with x509 certificates for mutual authentication (e.g. eIDAS certificates). The OAuth working group is currently working on a draft to standardise this: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-mtls-00

I strongly urge the working group to consider the FAPI Profile for OAuth 2.0 in its considerations as it has the following advantages over simply using TLS Mutual Auth or a custom PKI based solution for identity:

1. It is based on a battle tested framework that is used at scale by companies such as Google & Microsoft.
2. It has a well understood security model
3. It provides a clear framework where all the authentication and authorisation requirements of the RTS can be met - including the hardest requirement: securely conveying PSU authorization decisions (represented by the “authentication code” across a network between a TPP and an ASPSP.
4. It is a recognised international standard - not owned or controlled by any commercial entity
5. The OpenID foundation provides a robust testing framework that enables ASPSPs and TPPs to test conformance with the standard

Using a single certificate for all operations is bad practice

Using a certificate identifying a company to identify it’s software in every operation is not best practice. It would mean that all services that a TPP / ASPSP operates would need access to a

single private key. This would increase the chance of that key being compromised and make key rotation harder..

There is an important difference between a company and the software it produces:

- Many registered TPPs will have multiple software products
- A software product will often have a shorter lifetime than a company
- A software product may require a subset of the permissions that a company is granted

Our strong recommendation, based on significant PKI experience, is that the eIDAS cert with the role is used for onboarding a TPP with an ASPSP, but not for subsequent interactions.

Ensuring that a TPP hasn't had its permissions revoked and therefore its eIDAS certificate revoked can be achieved through other means, e.g. an intermediate CA that syncs with all eIDAS CAs and issues the certificates used by the software or a back-channel process that syncs revocation according to a schedule that matches the agreed liability regime.

While it may seem simpler for each ASPSP to perform an OCSP check on an eIDAS certificate issued to the company on every interaction there are practical difficulties with this:

- The TLS endpoints at the ASPSP would need to trust and make outbound requests to every eIDAS CA (QTSP)
- There is likely to be an increased latency and a chance for requests to fail

Strong Customer Authentication is best handled with "redirects"

Using OAuth 2.0 for an RTS interface involves the TPP redirecting the PSU to the ASPSP and the ASPSP performing SCA in it's own environment (either it's own website or it's own native app).

Without a contractual relationship between the TPP and the ASPSP we believe the redirect model is the only viable solution for RTS compliance.

Allowing the TPP to collect the PSU's "personalised security credentials" on behalf of the ASPSP is bad security practice and is no better than the current practice of "screen scraping". Furthermore it breaks the liability model that PSD2 builds around strong customer authentication.

SCA requires 2 out of 3 of the following elements: knowledge, inherence & possession. If an ASPSP is confirming any these elements indirectly (i.e. with a TPP in the middle) then they have a lower degree of confidence that it is the PSU inputting the data. For example a TPP could save a PSU's password and any time the ASPSP wants to perform SCA they could input the password on behalf of the PSU. The PSU would then only need to provide 1 element rather than 2 - in such a case has SCA taken place?

The ASPSP, TPP, PSU triangle is a classic 3 party auth problem that has been solved by OAuth 2.0. This standard allow a decent user experience for the PSU while having a clear separation of concerns (and liability) between the ASPSP and the TPP.

We trust that the above information is useful for you in your deliberations. We are happy to answer any questions or provide further contributions to your working group.

Best Regards

Dave Tonge
UK Implementation Entity Liaison Officer
Financial API Working Group
OpenID Foundation