RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: PDAC-F02

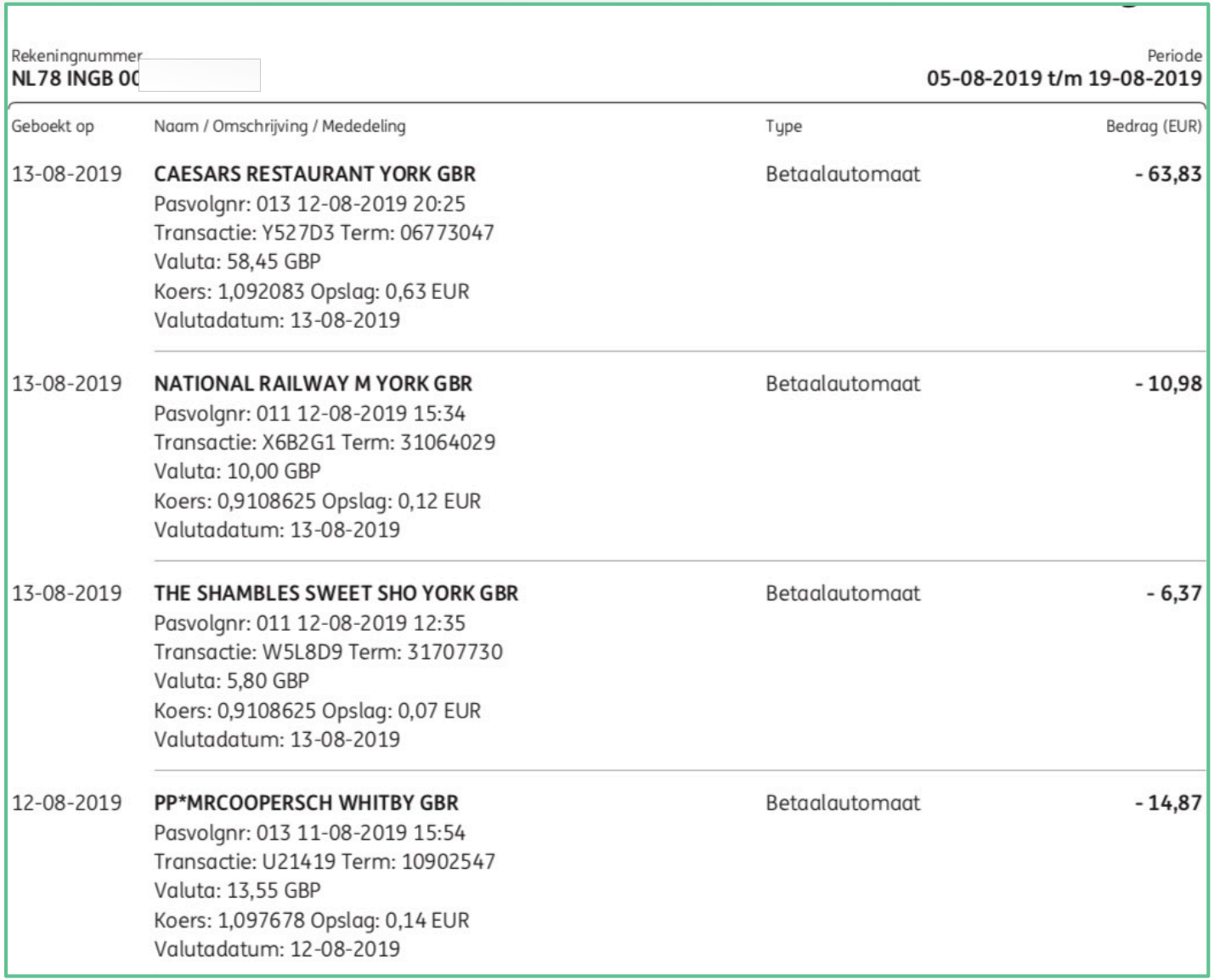
PSD2 Preparedness? Research Reveals FinTech Risks

Feike Hacquebord

Senior Threat Researcher
Trend Micro Research
@FeikeHacquebord

Imagine you have installed a new FinTech App

Would you mind sharing banking statements of the last ***18 months*** with a 3rd party other than your bank?



Rekeningnummer
NL78 INGB 00

Periode
05-08-2019 t/m 19-08-2019

Geboekt op	Naam / Omschrijving / Mededeling	Type	Bedrag (EUR)
13-08-2019	**CAESARS RESTAURANT YORK GBR** Pasvolgnr: 013 12-08-2019 20:25 Transactie: Y527D3 Term: 06773047 Valuta: 58,45 GBP Koers: 1,092083 Opslag: 0,63 EUR Valutadatum: 13-08-2019	Betaalautomaat	**- 63,83**
13-08-2019	**NATIONAL RAILWAY M YORK GBR** Pasvolgnr: 011 12-08-2019 15:34 Transactie: X6B2G1 Term: 31064029 Valuta: 10,00 GBP Koers: 0,9108625 Opslag: 0,12 EUR Valutadatum: 13-08-2019	Betaalautomaat	**- 10,98**
13-08-2019	**THE SHAMBLES SWEET SHO YORK GBR** Pasvolgnr: 011 12-08-2019 12:35 Transactie: W5L8D9 Term: 31707730 Valuta: 5,80 GBP Koers: 0,9108625 Opslag: 0,07 EUR Valutadatum: 13-08-2019	Betaalautomaat	**- 6,37**
12-08-2019	**PP*MRCOOPERSCH WHITBY GBR** Pasvolgnr: 013 11-08-2019 15:54 Transactie: U21419 Term: 10902547 Valuta: 13,55 GBP Koers: 1,097678 Opslag: 0,14 EUR Valutadatum: 12-08-2019	Betaalautomaat	**- 14,87**

Imagine you are a bank

Would you be worried when a FinTech firm uses your customers' passwords to access their banking statements online?

Imagine you are a FinTech startup

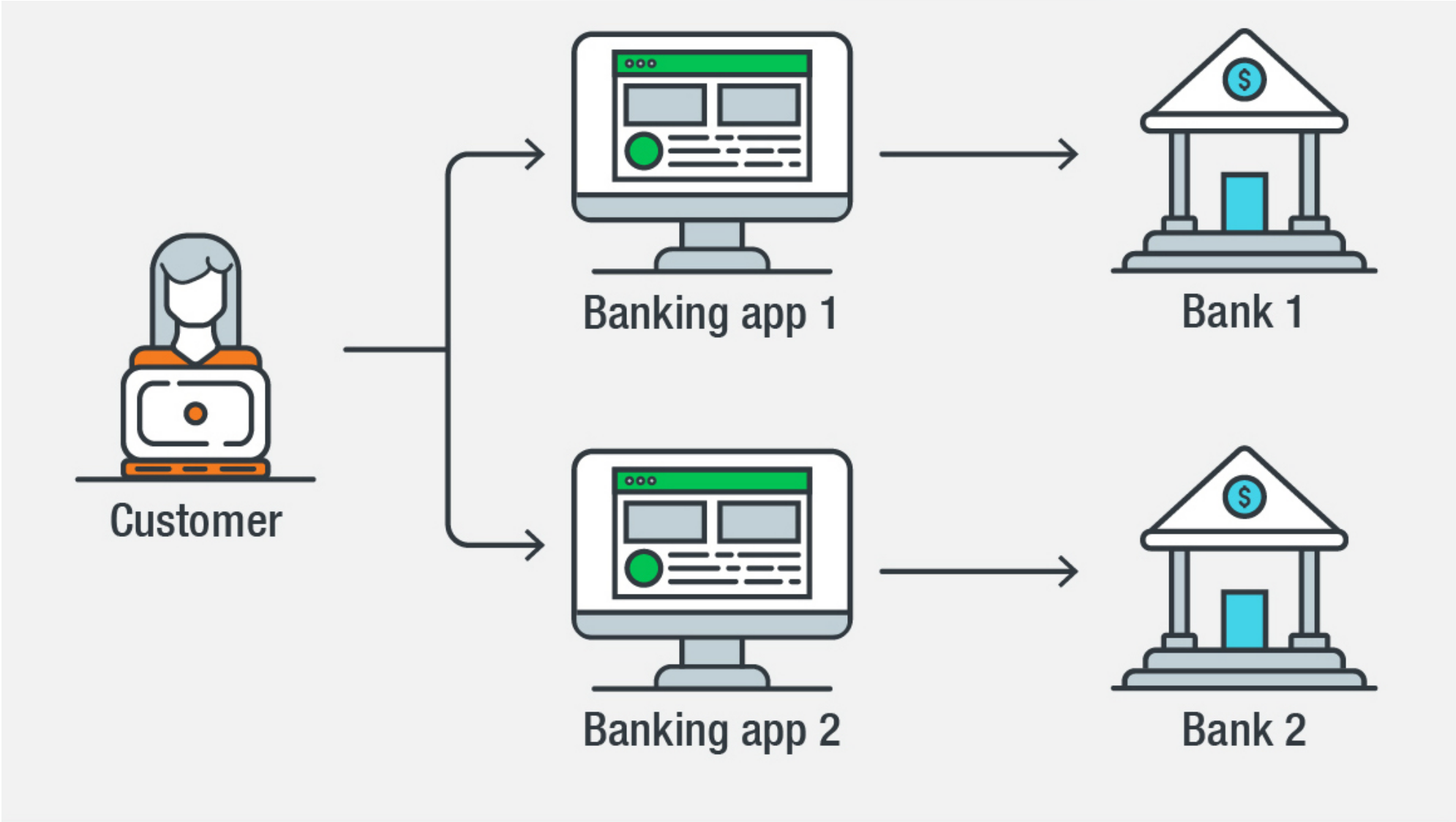
Are you prepared when you ***must*** use protocols like

OAuth, mTLS, Token Binding, PKCE,...

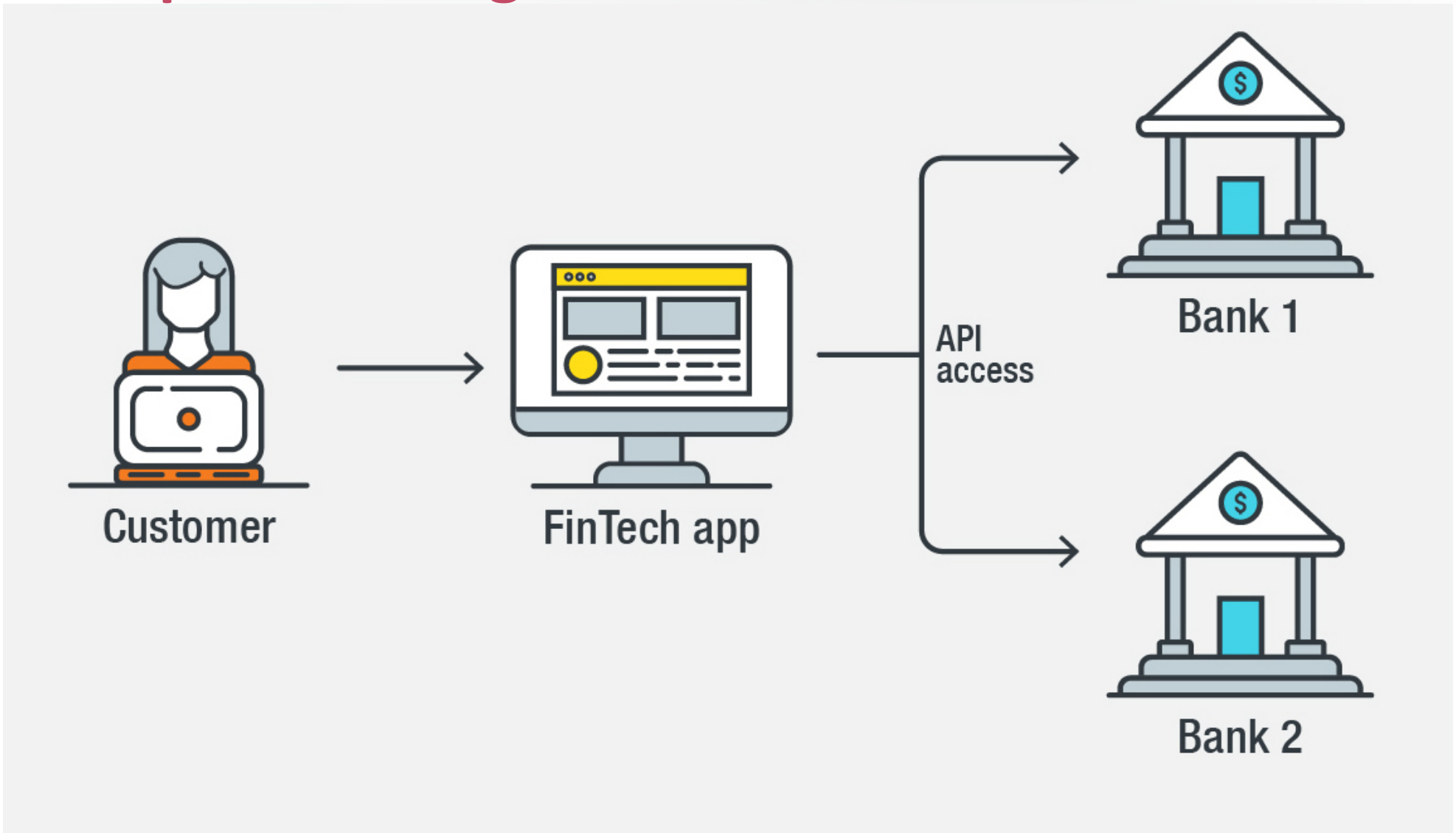
to access banking data on behalf of your customers?

Open Banking – worldwide revolution in FinTech

Before Open Banking



After Open Banking



Open Banking Worldwide

- Europe – PSD2 law since September 14 2019 (mostly postponed)
- US – no regulation but FS-ISAC published standards
- Australia – regulation in 2020
- Asia – some regulation
- South America – some regulation

Open Banking – key drivers

- Innovation
- Breaking monopoly banks
- Enhance service offering
- Additional revenue
- ….
- ….
- Enhanced Security (not a key driver really)

Enhanced Security in Open Banking

- PSD2 in EU
  - 2 factor authentication
  - dynamic linking (link between OTP and payee / amount)
- OAuth (far from 100% in practice)
- Financial grade API (FAPI) in UK (new security flaws to be fixed)

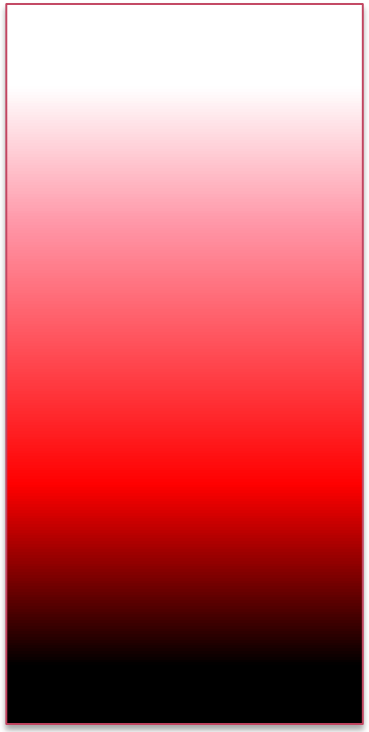
New trust relationships

- Banking customers <-> FinTech
  - customers trusted banks for a long time
  - customers unfamiliar with FinTech companies and unfamiliar how their data gets shared

- Banks <-> FinTech
  - Banks might perceive FinTech startups as a threat
  - FinTech startups might perceive banks as not cooperative

Risks of Open Banking

Actors interested in abuse of Open Banking



advertisers

data brokers

rogue FinTech companies

advanced bank fraudsters

nation state actors

Risks of Open Banking

- APIs
- Obsolete data sharing protocols
- New security modules might introduce new attacks
- Wrong implementation of complicated security modules
- Rogue use of customer data (like banking statements)
- Fraud detection becomes harder

Risks of Open Banking

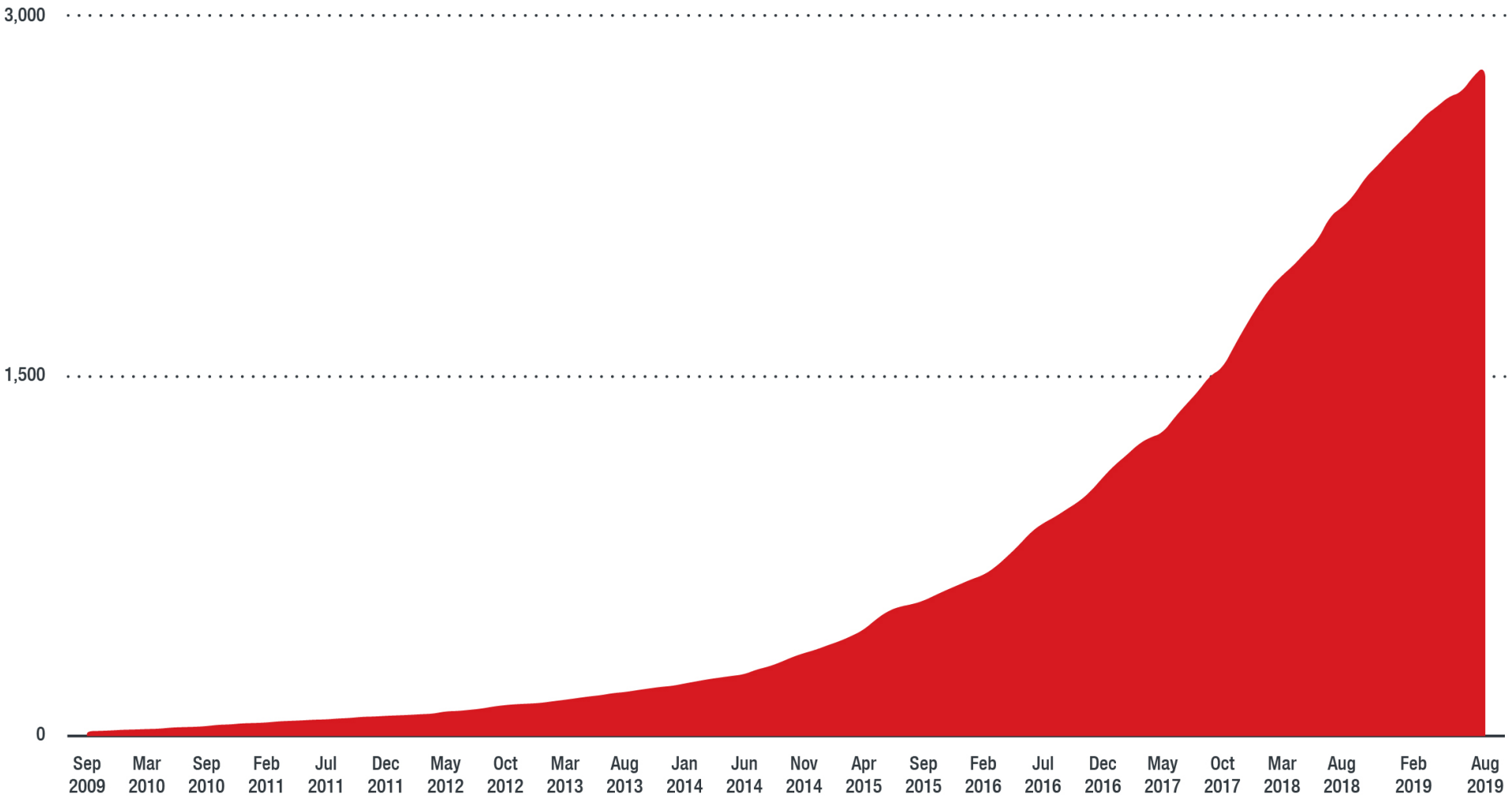
Simply put:

Open Banking widens the attack surface

APIs in (Open) Banking

- APIs are central to the success of Open Banking
- Banks were relatively late with creating APIs

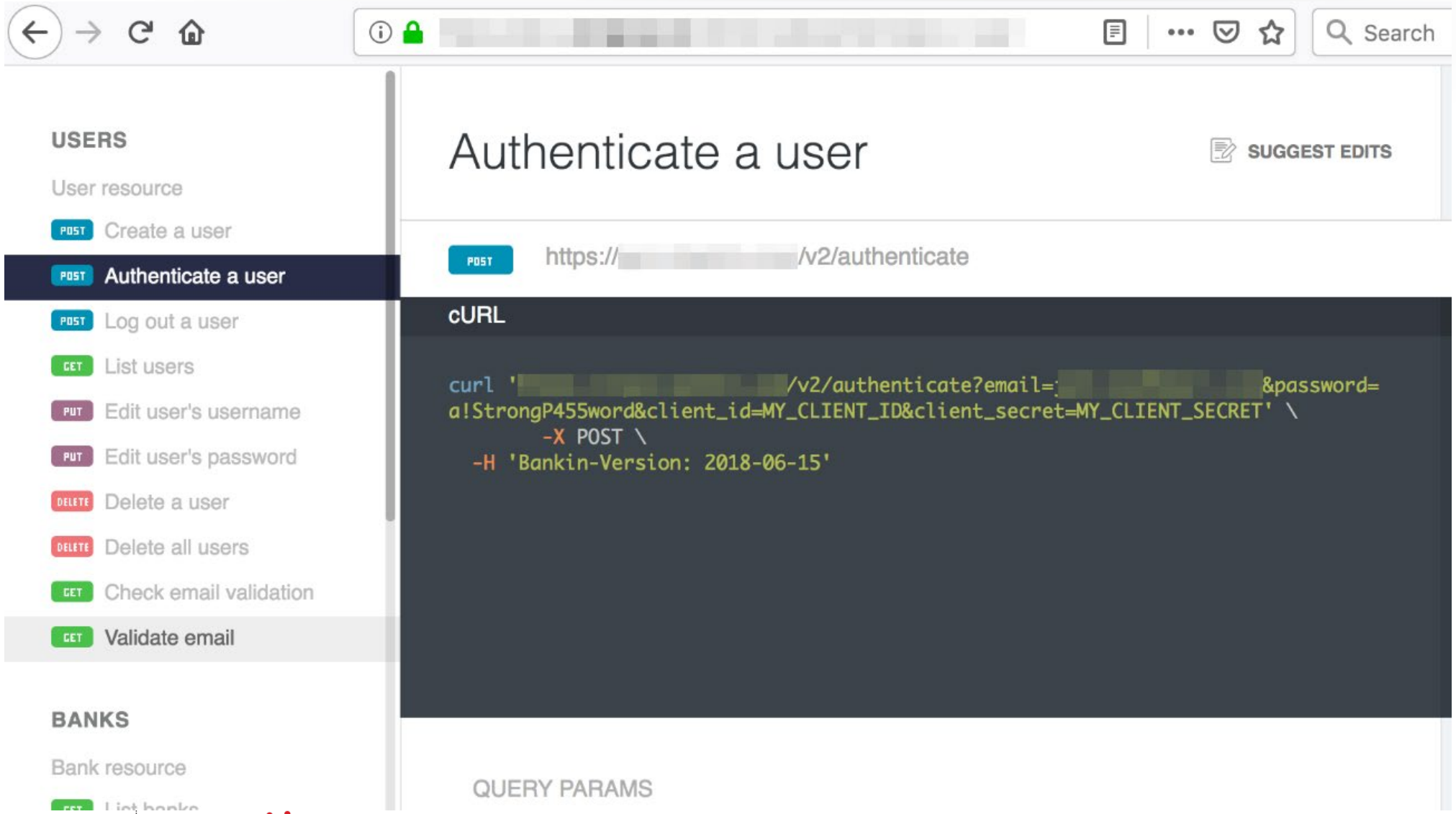
Growth of banking APIs over time



APIs in (Open) Banking

APIs should not leak sensitive information...

Credentials in API URL of FinTech company



Sensitive info in API URL of an EU bank

```
api.[BANK_NAME].[redacted].[redacted]/auth?fingerPr
int=[....]&longitude=[...]&deviceToken=[...]&latitu
de=[..]&client_id=[...]&client_secret=[....]&grant_
type=password&password=[...]&username=[...]
```

Sensitive info in API URL of an Asian bank

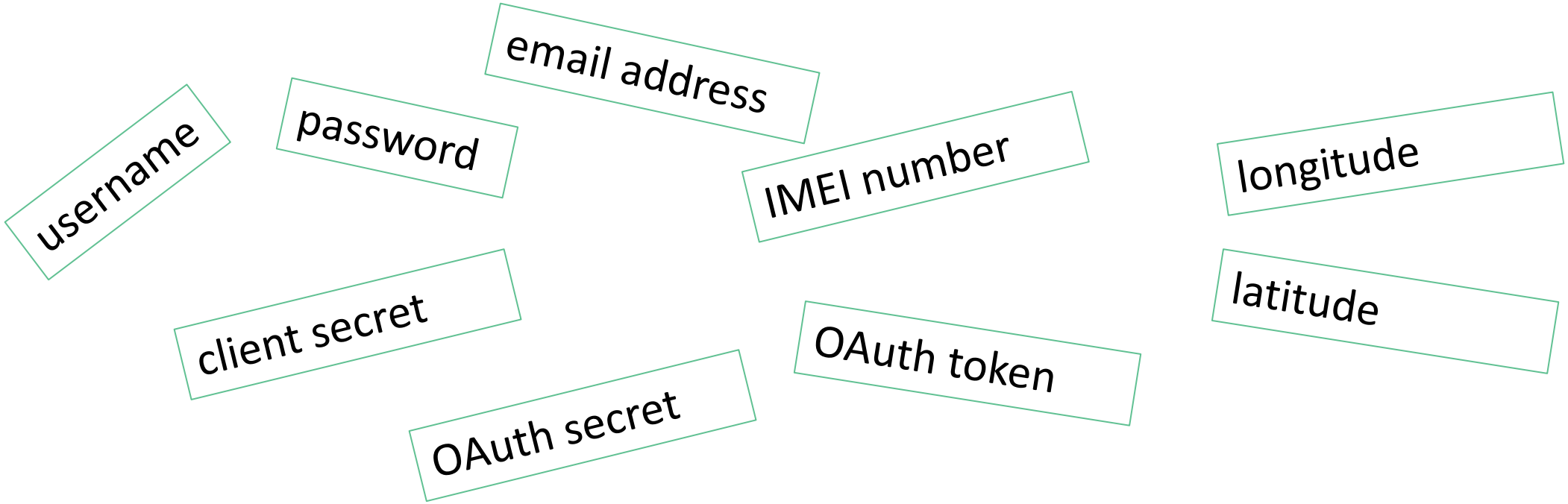
```
openapi.[redacted]/erh/unlogin/ma_data_query?tradeH
ead=[...]&queryType=[...]&firstCoinCode=[..]&lastCo
inCode=[..]&cardType=[...]&branchNo=[...]&startTime
=&uuid=[...]&clientid=[...]&client_secret=[...]&cal
lback=[...]&_[timestamp]
```

Sensitive info in API URL of a European central bank

```
api.[redacted]/NBUStatEntryPoint/oauth/token?grant_
type=password&client_id=portal&username=[....]&pass
word=[....]
```

Sensitive info in URL paths of APIs

- We looked into dozens APIs of financial companies
- Many of them contain sensitive info in the URL path



Sensitive info in URL paths: a problem

- URLs stored in browsing history
- URLs stored in access log files
- URLs stored in proxy logs
- URLs shared between different devices

- Might hinder enhanced security in Open Banking
- Bad practice anyway – “easy” to fix.

Obsolete sharing protocols still in use

- screen scraping / direct access
- Open Financial Exchange (OFX)

Open Financial Exchange (OFX)

- Protocol developed in 1997 by Microsoft, Intuit and CheckFree
- No real multi factor authentication
- OAuth in latest version, but not used
- Still used in US by many banks

Open Financial Exchange (OFX)

- Which protocol version most popular in US?

```
OFX v.1.0.2 - Released May 1997
OFX v.1.0.3 - Unconfirmed date
OFX v.1.0.6 - Released October 1999
OFX v.2.0.3 - Released May 2006
OFX v.2.1.1 - Released May 2006
OFX v2.2.0 (current) - Released November 2017
```

Open Financial Exchange (OFX)

- Which protocol version most popular in US?

```
OFX v.1.0.2 - Released May 1997
OFX v.1.0.3 - Unconfirmed date
OFX v.1.0.6 - Released October 1999
OFX v.2.0.3 - Released May 2006
OFX v.2.1.1 - Released May 2006
OFX v2.2.0 (current) - Released November 2017
```

<- 79%

<- 21%

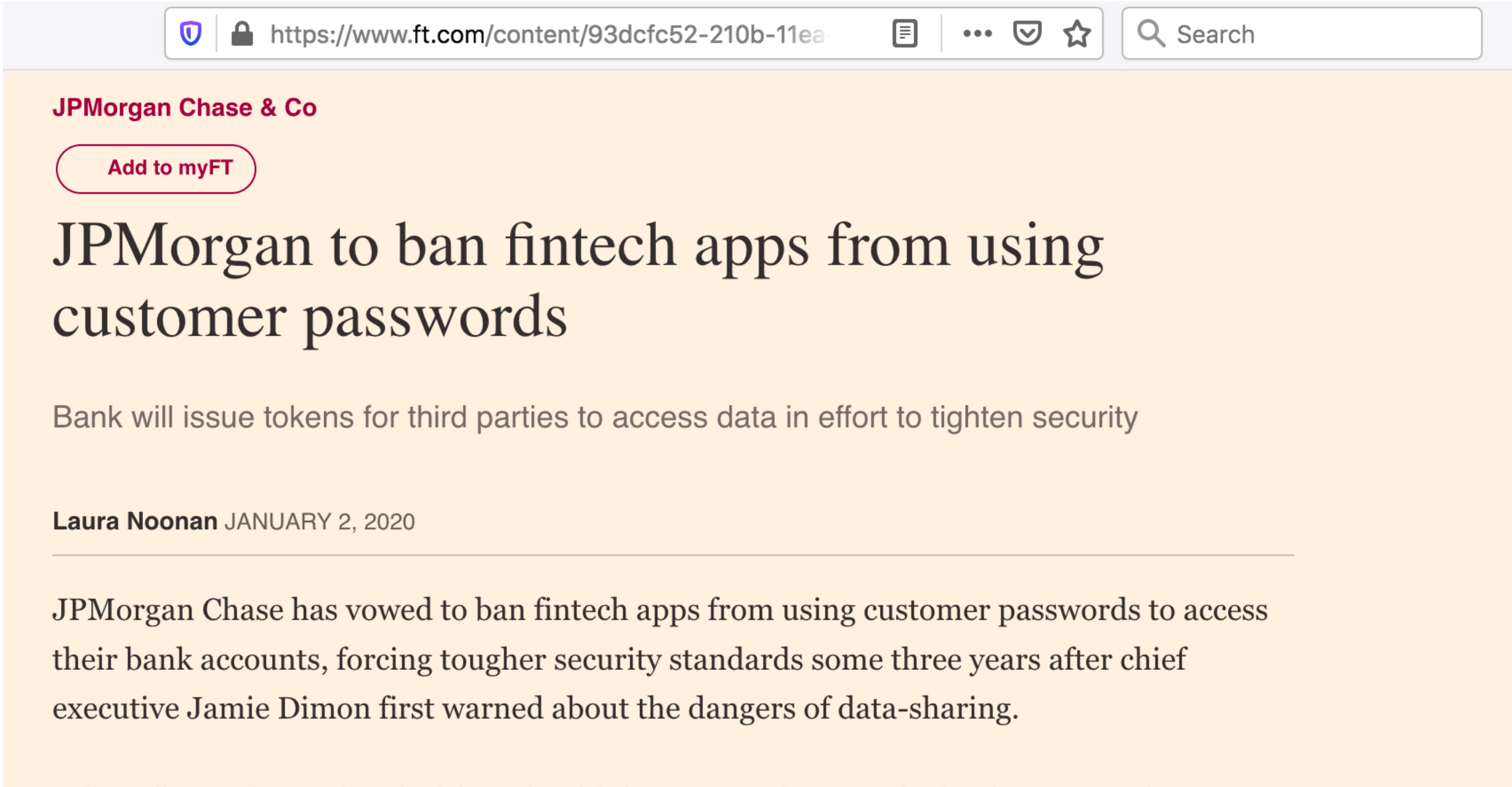
What is Screen scraping

- Fintech company uses banking customers' password to log on and scrape financial data by parsing HTML code
- Also called "direct access" by FinTech companies
- FinTech companies often claim there is no known breach because of screen scraping
- Some FinTech companies are reluctant to use OAuth instead

Obvious risks of Screen scraping ("direct access")

- Banking customers should never share passwords
- Goes against good practices to educate banking customers
- Risk of data breaches
- No easy way to revoke 3rd party access to banking details
- No way to give different levels of access permissions

US bank announces to ban screen scraping



Debate in Australia on Screen Scraping

CBA's Comyn wants screen scrapers scrubbed from Consumer Data Right

By Julian Bajkowski
Feb 4 2020
12:35PM

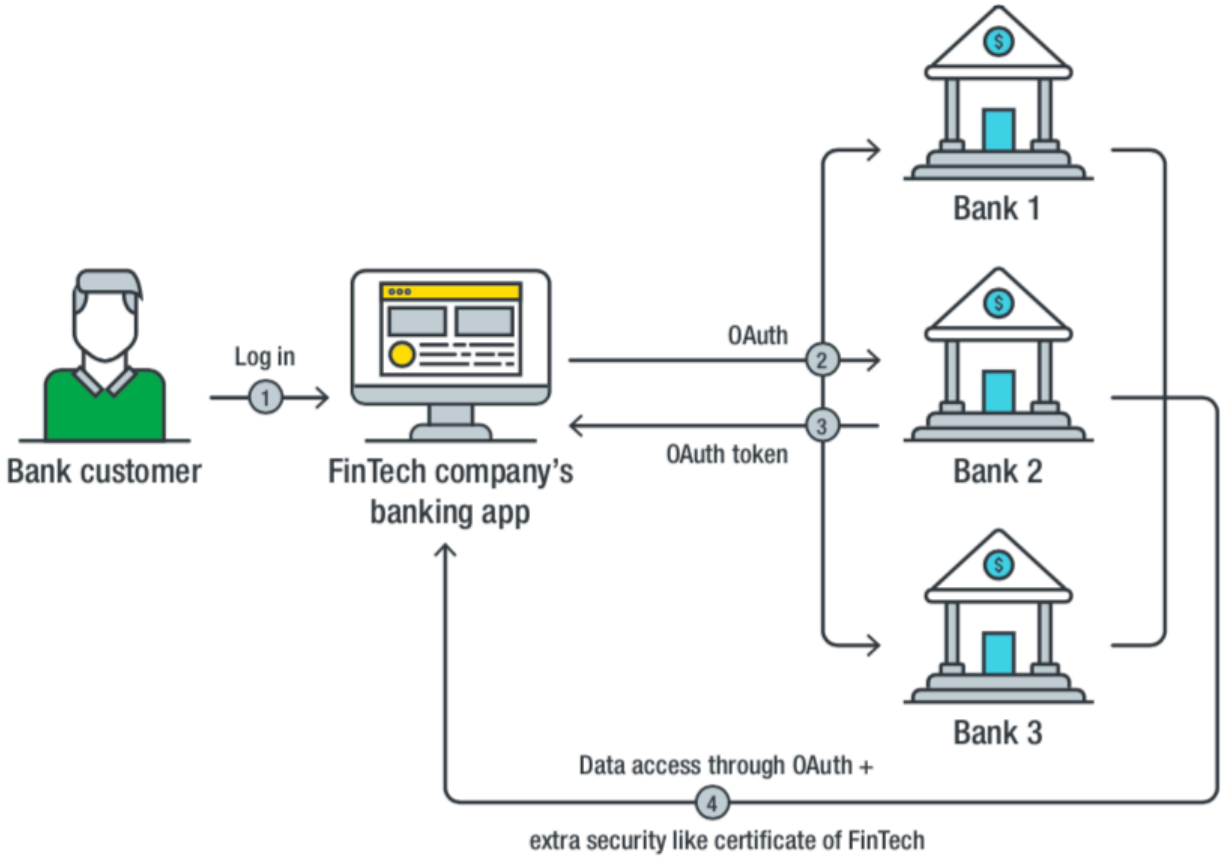
Security credentials not for sharing.

The chief executive of the Commonwealth Bank of Australia, Matt Comyn, has publicly hit back against calls by sections of the fintech sector to legitimise the controversial practice of screen scraping, saying the

Financial Grade API (FAPI) in UK

- FAPI is meant to be secure in high risk environment
- Developed in UK by OpenID and banks since 2017
- Serious flaws found early 2019 by German academic researchers Daniel Fett et al (2019).
- Not ready on September 14 2019 (PSD2 deadline)

Financial grade API (FAPI)



FAPI =

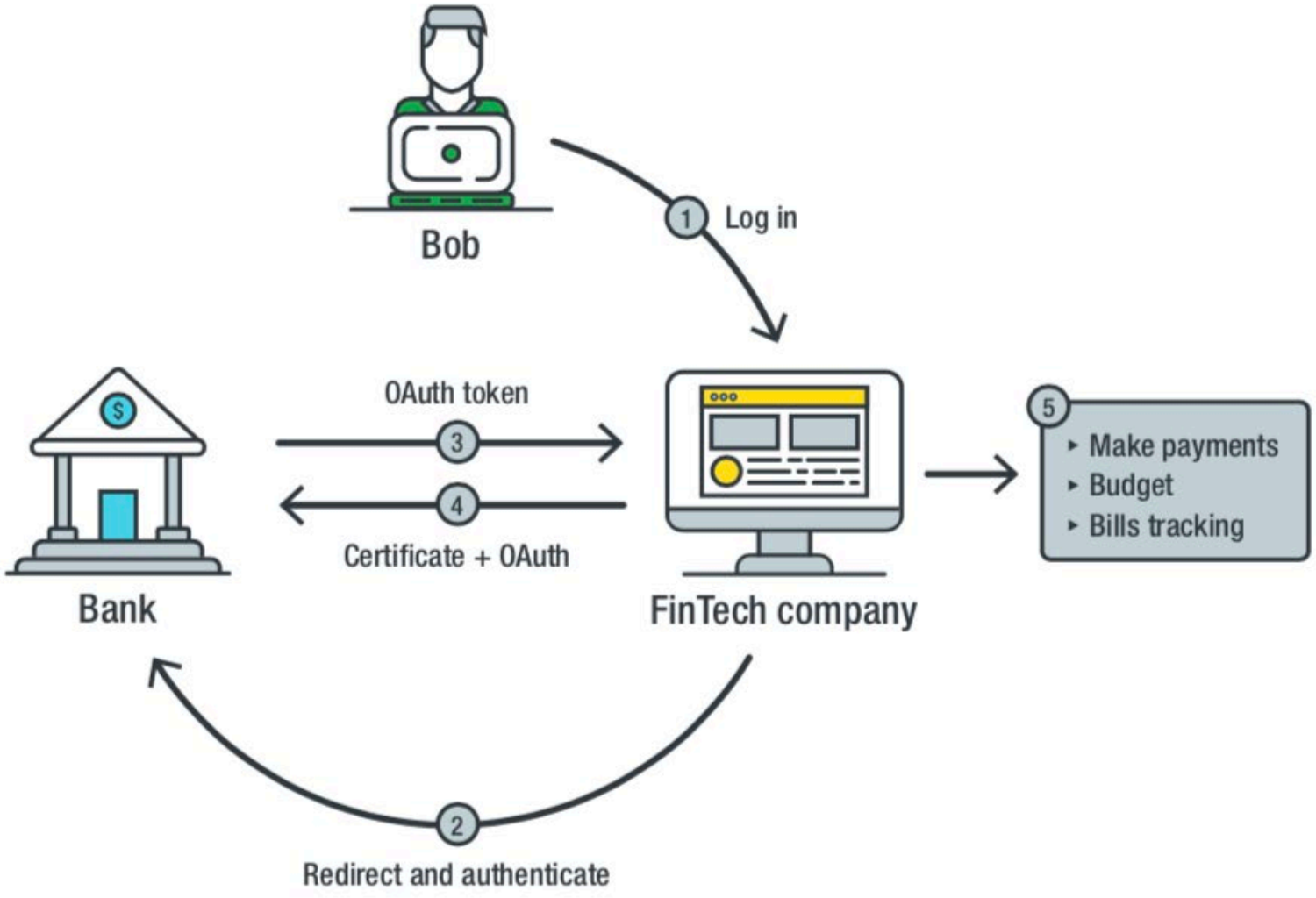
OAuth2.0 + extra modules

Financial Grade API (FAPI)

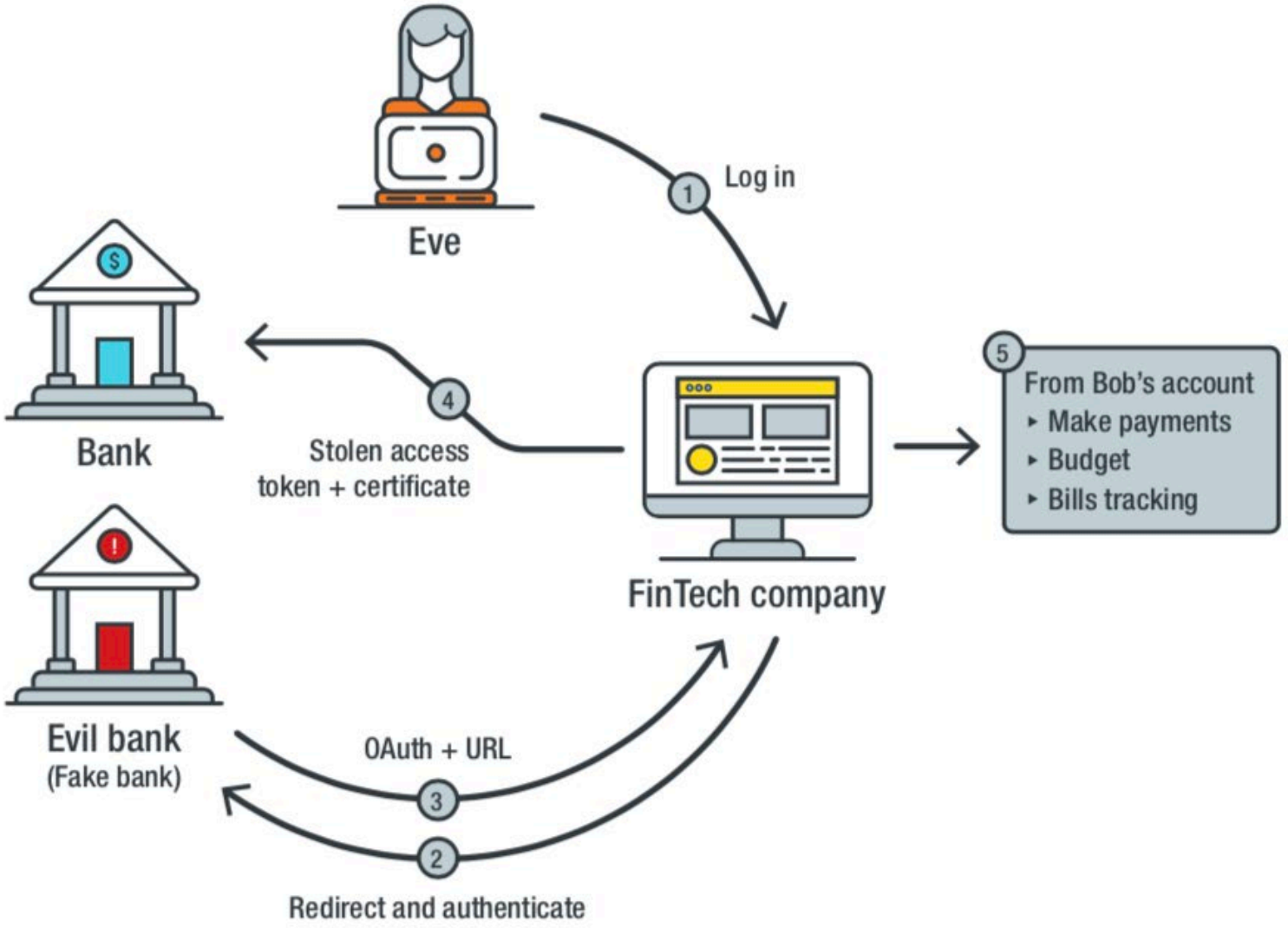
FAPI = OAuth2.0

+ mutual TLS (mTLS)
+ OAuth token binding (probably dropped)
+ JWS Client Assertion
+ Proof Key for Code Exchange (PKCE)

Example of mTLS



Attack scenario against mTLS



Actions to take – for banks

- Review your (legacy) APIs
- Retire/update APIs with sensitive info in URL path
- Retire OFX servers
- Learn from FAPI experiences in UK
- Consider adopting zero trust model
- Educate your customers about new phishing attacks

Actions to take – for FinTech companies

- Review your (legacy) APIs
- Retire/update APIs with sensitive info in the URL path
- Stop “screen scraping”, adopt OAuth
- Learn from FAPI experience in UK
- Learn how to implement security protocols
- Think about Threat Actors who might attack you
- Educate your customers about security

Lessons learned – for banking customers

- FinTech companies may offer useful new financial services
- However, review which data the FinTech company is collecting from you
- Read the fine print about sharing your data with 3rd parties
- Be prepared for new phishing techniques
- Know how to revoke access permissions of the FinTech service you have signed up with

More details in our paper

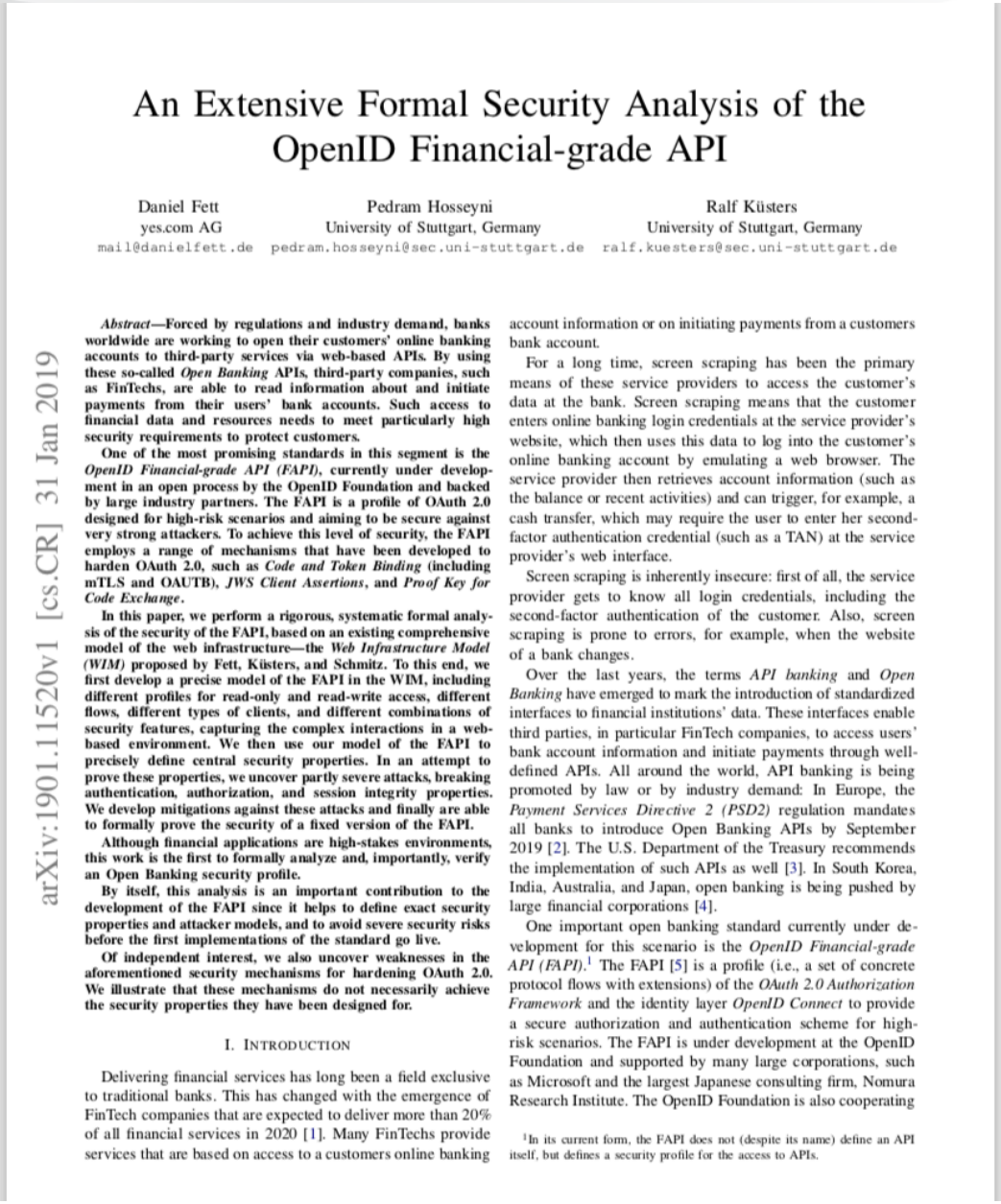
Ready or Not for PSD2:
The Risks of Open Banking

available on trendmicro.com

More details on FAPI

"An Extensive Formal Security Analysis of theOpenID Financial-grade API" by Daniel Fett et al.

available on arxiv.org



An Extensive Formal Security Analysis of the OpenID Financial-grade API

Daniel Fett
yes.com AG
mail@danielfett.de

Pedram Hosseyni
University of Stuttgart, Germany
pedram.hosseyni@sec.uni-stuttgart.de

Ralf Küsters
University of Stuttgart, Germany
ralf.kuesters@sec.uni-stuttgart.de

***Abstract*—Forced by regulations and industry demand, banks worldwide are working to open their customers' online banking accounts to third-party services via web-based APIs. By using these so-called *Open Banking* APIs, third-party companies, such as FinTechs, are able to read information about and initiate payments from their users' bank accounts. Such access to financial data and resources needs to meet particularly high security requirements to protect customers.**

One of the most promising standards in this segment is the *OpenID Financial-grade API (FAPI)*, currently under development in an open process by the OpenID Foundation and backed by large industry partners. The FAPI is a profile of OAuth 2.0 designed for high-risk scenarios and aiming to be secure against very strong attackers. To achieve this level of security, the FAPI employs a range of mechanisms that have been developed to harden OAuth 2.0, such as *Code and Token Binding* (including mTLS and OAUTB), *JWS Client Assertions*, and *Proof Key for Code Exchange*.

In this paper, we perform a rigorous, systematic formal analysis of the security of the FAPI, based on an existing comprehensive model of the web infrastructure—the *Web Infrastructure Model (WIM)* proposed by Fett, Küsters, and Schmitz. To this end, we first develop a precise model of the FAPI in the WIM, including different profiles for read-only and read-write access, different flows, different types of clients, and different combinations of security features, capturing the complex interactions in a web-based environment. We then use our model of the FAPI to precisely define central security properties. In an attempt to prove these properties, we uncover partly severe attacks, breaking authentication, authorization, and session integrity properties. We develop mitigations against these attacks and finally are able to formally prove the security of a fixed version of the FAPI.

Although financial applications are high-stakes environments, this work is the first to formally analyze and, importantly, verify an Open Banking security profile.

By itself, this analysis is an important contribution to the development of the FAPI since it helps to define exact security properties and attacker models, and to avoid severe security risks before the first implementations of the standard go live.

Of independent interest, we also uncover weaknesses in the aforementioned security mechanisms for hardening OAuth 2.0. We illustrate that these mechanisms do not necessarily achieve the security properties they have been designed for.

I. Introduction

Delivering financial services has long been a field exclusive to traditional banks. This has changed with the emergence of FinTech companies that are expected to deliver more than 20% of all financial services in 2020 [1]. Many FinTechs provide services that are based on access to a customers online banking account information or on initiating payments from a customers bank account.

For a long time, screen scraping has been the primary means of these service providers to access the customer's data at the bank. Screen scraping means that the customer enters online banking login credentials at the service provider's website, which then uses this data to log into the customer's online banking account by emulating a web browser. The service provider then retrieves account information (such as the balance or recent activities) and can trigger, for example, a cash transfer, which may require the user to enter her second-factor authentication credential (such as a TAN) at the service provider's web interface.

Screen scraping is inherently insecure: first of all, the service provider gets to know all login credentials, including the second-factor authentication of the customer. Also, screen scraping is prone to errors, for example, when the website of a bank changes.

Over the last years, the terms *API banking* and *Open Banking* have emerged to mark the introduction of standardized interfaces to financial institutions' data. These interfaces enable third parties, in particular FinTech companies, to access users' bank account information and initiate payments through well-defined APIs. All around the world, API banking is being promoted by law or by industry demand: In Europe, the *Payment Services Directive 2 (PSD2)* regulation mandates all banks to introduce Open Banking APIs by September 2019 [2]. The U.S. Department of the Treasury recommends the implementation of such APIs as well [3]. In South Korea, India, Australia, and Japan, open banking is being pushed by large financial corporations [4].

One important open banking standard currently under development for this scenario is the *OpenID Financial-grade API (FAPI)*.[1] The FAPI [5] is a profile (i.e., a set of concrete protocol flows with extensions) of the *OAuth 2.0 Authorization Framework* and the identity layer *OpenID Connect* to provide a secure authorization and authentication scheme for high-risk scenarios. The FAPI is under development at the OpenID Foundation and supported by many large corporations, such as Microsoft and the largest Japanese consulting firm, Nomura Research Institute. The OpenID Foundation is also cooperating

[1] In its current form, the FAPI does not (despite its name) define an API itself, but defines a security profile for the access to APIs.

Thank You

feike_hacquebord AT trendmicro.com

@FeikeHacquebord on Twitter

RSAConference2020

Appendix

References

References

- Daniel Fett et al (2019). *2019 IEEE Computer Society*. “An extensive formal security analysis of the OpenID Financial-grade API.” - https://arxiv.org/abs/1901.11520
- Laura Noonan (2020), Financial Times, “JPMorgan to ban fintech apps from using customer passwords”. https://www.ft.com/
- Julian Bajkowski (2020), itnews.com.au, “CBA's Comyn wants screen scrapers scrubbed from Consumer Data Right”. https://www.itnews.com.au
- Feike Hacquebord et al (2019). “Ready or Not for PSD2. The risks of Open Banking”. https://www.trendmicro.com/

RSA Conference 2020

RSA Conference 2020

RSA Conference 2020