Ready or Not for PSD2:
The Risks of Open Banking

Feike Hacquebord, Robert McArdle, Fernando Mercês, and David Sancho

TREND MICRO | research

Published by

Trend Micro Research

Written by

Feike Hacquebord, Robert McArdle, Fernando Mercês, and David Sancho

Stock image used under license from Shutterstock.com

Contents

On September 14, a new European Union (EU) banking directive called the Revised Payment Service Directive (PSD2) — more popularly known as Open Banking — took effect. The new set of rules aim to give the user greater control over their banking data, and includes the option to share their information with third-party suppliers for financial software use such as budgeting, bill management, and payment automation. While PSD2 is an EU directive, the effects have been global; banks in the U.S. and Asia are implementing the similar or comparable standards to provide their customers with services that match their European counterparts.

Our research reveals that Open Banking will increase the attack surface that banks and their customers face today, creating new opportunities for cybercriminals that target them. Threats such as:

- Customers who authorize Open Banking apps to manage their data will find themselves in a new trust relationship. Users previously trusted established financial institutions with a long history of security, and users will now have to give that same level of trust to lesser-known third-party providers. A sampling of these financial technology (FinTech) providers showed that they are generally smaller software companies, and often with no specific individual dedicated only to security.
- An Open Banking customer can be more vulnerable to phishing attacks. Today's users are mostly conditioned to the fact that banks will not email with links or requests to login. However, cybercriminals could use Open Banking apps as a new hook for old phishing techniques.
- New does not always mean better, which is especially true when it comes to security. We have seen banks with a history of exposing personally identifiable information (PII) in the URLs of their existing application programming interface (API). We expect to see a number of implementation flaws in Open Banking APIs that attackers will be quick to investigate as these apps go live.
- Not all Open Banking apps will have permission to make payments on the customers' behalf. However, sophisticated attackers consider transaction data as high value information in itself. Knowing when and where a customer makes purchases allows an attacker to build a picture of users' daily movements, routines, interests, and financial standing.

This research paper looks into the standard by which the directive is based on, prior regulations guiding this directive, the new risks brought by Open Banking, and the security recommendations for both FinTech companies and their customers.

Introduction

With increasing mobile device capabilities and internet connectivity, expect the financial and banking industry to follow suit and explore these new standards to improve their respective services. Cash is not as common as users and providers move towards the use of mobile and internet platforms for payments and transactions. These emerging banking industry disruptions have led to the emergence of new players in the financial industry — Apple, Facebook, and Google, along with start-ups — that are eager to roll out their new services. Forced by the new competitors, established banks are working with new financial technology (FinTech) companies — defined for this research as financial technology and software service providers — to adapt and develop more application programming interfaces (APIs) with Open Banking.

The European Union's (EU) new law, the Revised Payment Service Directive (PSD2) that was approved on October 2015 to replace the existing Payment Services Directive, has further accelerated these new developments. The PSD2 aims to promote the innovation of online and mobile payments, making international payments in and from the EU easier, more efficient, and more secure. The regulation also aims to encourage more competition in the banking industry, establish the security requirements to reduce the risks of fraud, establish guidelines for companies' data handling, raise awareness of users and providers of their rights and obligations, and ensure better protection and security for consumers.[1]

It should also be noted that in an effort to make payment processing more efficient and cost-effective, the directive gives service users an overview of their current financial situations, which they can use to better manage their personal finances. For third party and service providers, enabling an open API to share customers' banking data allows additional services, as well as a reduction in surcharge costs. For instance, this allows a customer to use an app that accesses the balances and payment data from the user's bank accounts to get an overview of his/her finances. The data can then be used for analysis such as budgeting, initiating payments, and financial recommendations.

With the directive scheduled for implementation on September 14, 2019, PSD2 inevitably forces European banks to implement open APIs to share customers' banking data with their FinTech providers to comply with regulations and deliver the additional services. Because banking data of customers will be spread over more (FinTech) companies, a much larger attack surface for cybercriminals to abuse becomes evident. We examined the financial sector's readiness for the new regulations, as well as a number of the directive's stipulations. We also looked into the ways that attackers could use and abuse Open Banking to circumvent prevailing security measures, as well as the current solutions and recommendations that these companies have and can put into place.

PSD2 methods

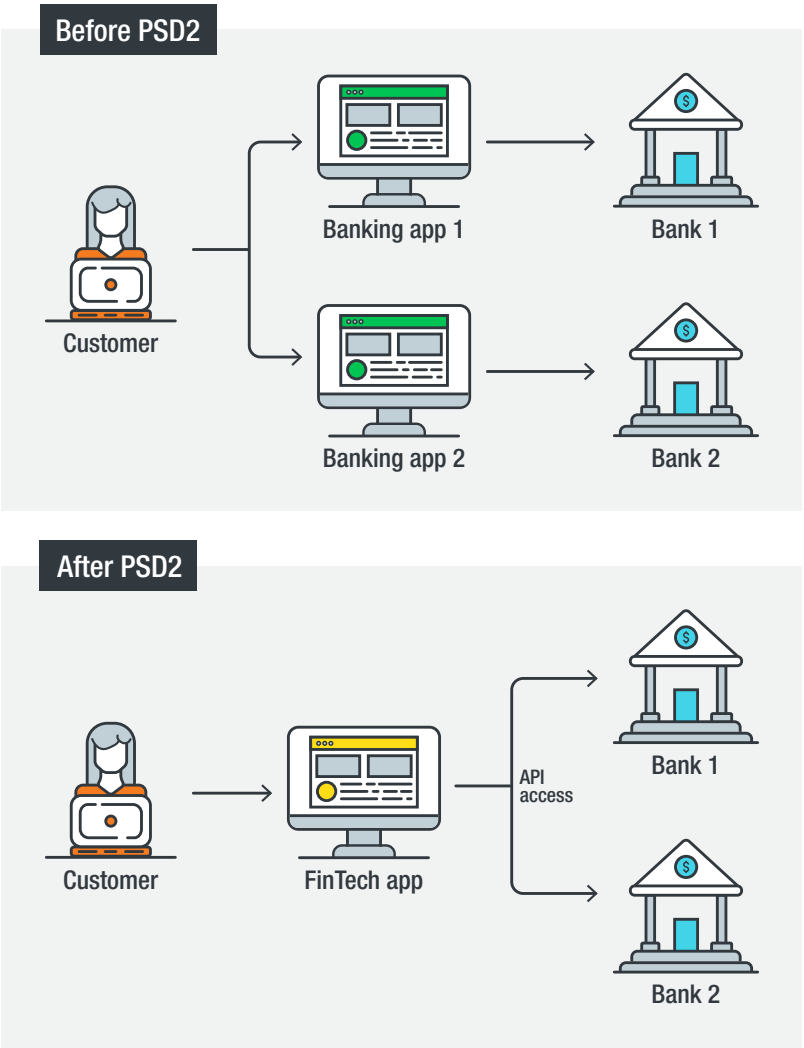

Figure 1. With PSD2, new FinTech companies will launch new apps to aggregate banking data from multiple accounts.

Open Banking is expected to generate innovation and more competition in the banking industry. To mitigate the risks that may come with these changes, PSD2 forces banks to implement mandatory security measures in addition to their established procedures. For instance, the new law requires banks to implement multi-factor authentication (MFA) for all proximity and remote transactions requested from any channel. Accordingly, this means using two of the three elements,[2] below:

- **Knowledge** – Something only the user knows, such as passwords and security questions
- **Possession** – Something only the user possesses, such as a phone, token, or card reader
- **Inherence** – Something unique or inherent to the user, such as biometrics (fingerprint, voice, or face)

The three elements must be independent (a breach of one element must not compromise the reliability of the others) and designed to protect the confidentiality of the authentication data.

PSD2 also introduces a new concept called "Dynamic Linking,"[3] which is a method that requires a unique authentication code for each transaction that is specific to the amount sent and the recipient. Any change made to either the amount or the recipient would invalidate the authentication code, and therefore result in the transaction's failure and cancellation.

Adoption and implementation in other regions

The Open Banking APIs should be available for use beginning September 14, 2019. Currently, the United Kingdom leads the pack within Europe and appears ready for the most part, likely furthered by the Competition and Markets Authority's (CMA) push for PSD2 compliance. However, as this paper shows, the initial ambitions have yet to be met. For example, some of the additional security measures to be established on top of the minimum PSD2 requirements have been postponed. The rest of Europe should be ready by the said date, but there are serious doubts that all banks and FinTech companies can be ready by September.

The PSD2 will likely affect the rest of the world as well. For instance, the U.S. banking industry seems to have recognized the need to adopt Open Banking to keep up with European banks. Banks with branches that operate in Europe will have to adopt the system to comply and make them competitive in their respective markets, likely prompting the rest of their American counterparts to follow suit. The Financial Services Information Sharing and Analysis Center (FS-ISAC) consortium in the U.S. published API standards in 2018,[4] which stipulates functions similar to the APIs mandated in PSD2. While U.S. banks are under no obligation, the adoption of PSD2 guidelines will likely drive market pressure. Despite these developments, we have seen that U.S. banks in 2019 are still using legacy systems that could be viewed as early versions of Open Banking with much less security in place.

In Australia, the government announced the introduction of the Consumer Data Right (CDR) in 2017.[5] The CDR aims to make it easier for customers to switch between service providers and is expected to spur innovation and better service offers. Open Banking is part of CDR: Banks must make banking data available to third parties on request of their customers beginning February 2020.

In Asia, South Korea launched a platform for a common API structure across financial institutions as early as 2016. The Financial Services Commission in South Korea has announced plans to make Open Banking available by December 2019.[6] Singapore leads the region in Open Banking adoption — financial institutions have been openly developing and sharing their own APIs, as encouraged by the government's Smart Nation Singapore initiative in 2014 and the Monetary Authority of Singapore's (MAS) timeline and guidelines for the adoption of open data and payment technologies.[7] This enables financial entities to work with other service providers, similar to what PSD2 will allow. In other countries like Japan, bigger

enterprises have enabled third-party data sharing since 2015, even with the absence of a regulation to accommodate business counterparts in Europe.[8]

In Latin America, Mexico has also made concrete steps towards Open Banking implementation, passing the FinTech law in March 2018. This should help the country define their terms and guidelines over the next two to three years.

A larger attack surface

The introduction of Open Banking will widen the banking industry's attack surface, mainly because it puts banking data in the hands of more companies with diverse approaches to customer data protection. In addition, these new companies may not be subject to the same stringent regulations of the banks themselves. The following are some of the areas that are susceptible to cybercriminal attacks:

- Publicly accessible APIs of banks and FinTech companies
- Mobile apps of banks and FinTech companies
- New security extensions and modules that may introduce new attacks initially overlooked during the design and testing phases
- Security modules not implemented correctly
- Obsolete banking data sharing techniques that are still in use

The next sections will discuss these issues in detail, showing that the introduction of APIs in the financial sectors also comes with serious concerns. We will also argue that mobile apps often depend on a lot of external software that may introduce risks; U.K. banks had ambitious plans to implement a Financial-grade API (FAPI) for Open Banking, but academic researchers found security issues (which they published in January 2019). We will also discuss the practice of screen-scraping by FinTech companies, as well as the use of a two-decade protocol called Open Financial Exchange (OFX) by U.S. banks.

Banking APIs

Traditional banks were considerably late in setting up APIs. European banks, for instance, only began working with APIs in 2015, around the time PSD2 was approved. Today, an increasing amount of modern banks in the U.S., Europe, and Asia use APIs intended and designed for data sharing. Figure 2 shows the amount of financial APIs registered within the last 10 years.

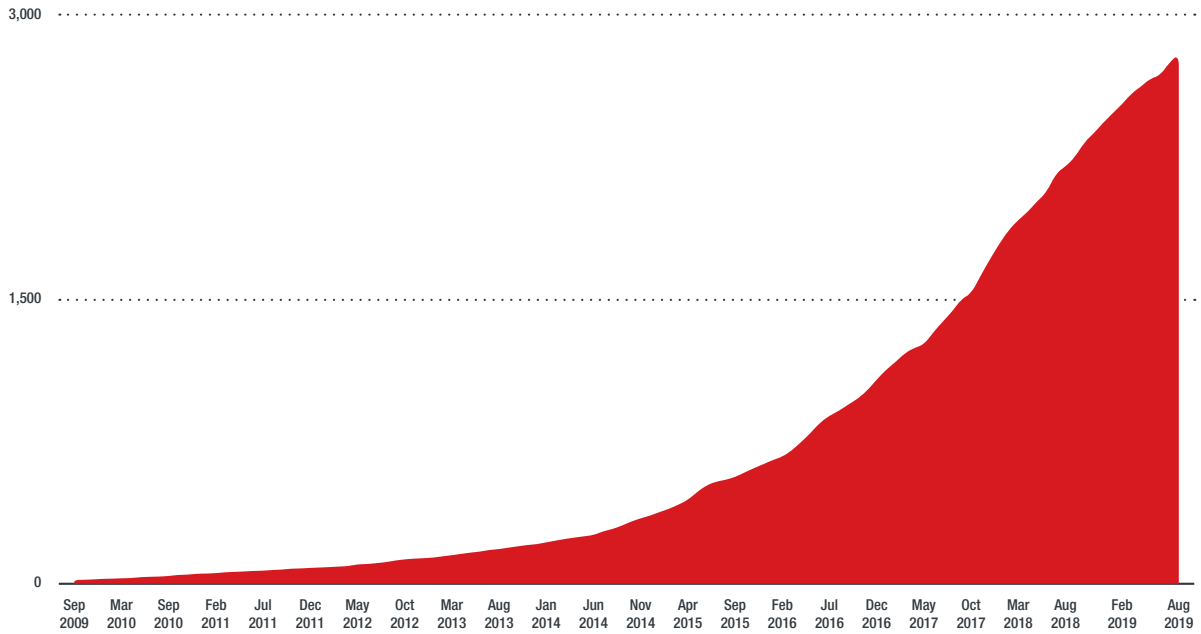

Figure 2. Growth trend of a particular subset of financial sector API hostnames, taken from Trend Micro's Smart Protection Network (SPN)

The Swedish FinTech company Tink published a blog article on the status of European banks' APIs, claiming that they were not even ready for testing purposes three months before the implementation of PSD2.[9] If the report is true, then the September 14, 2019 deadline enacted by PSD2 can be considered tight and likely unrealistic.

We also found other significant problems with numerous financial sector APIs and servers in other countries, though not necessarily meant for Open Banking. Some of the servers are using legacy systems that are still online, but they belong to banks, FinTech companies, and even central banks (national government banks). We found a significant number of banks exposing sensitive information such as authentication

parameters (e.g., one-time passwords or OTPs, access tokens, OAuth client secrets), privacy-sensitive data (e.g., email addresses, IMEI phone codes), and transaction data in the URLs of APIs and websites. While the use of secure sockets layer (SSL) ensures that these URLs are encrypted and unreadable in case traffic is intercepted, it is still not a good practice to put authentication, privacy, and transaction data in the URL path for the following reasons:

- The parameters may end up in web server log files
- The parameters will be visible in the browser's URL
- The parameters may end up in the user's browsing history
- The parameters may still be visible to any device doing legitimate SSL interception, such as network monitoring equipment with a trusted certificate.

Sample URL	Description
`api.[BANK_NAME].[redacted].[redacted]/auth?fingerPrint=[....]&longitude=[...]&deviceToken=[...]&latitude=[..]&client_id=[...]&client_secret=[....]&grant_type=password&password=[...]&username=[...]`	A European bank
`api.[redacted]/NBUStatEntryPoint/oauth/token?grant_type=password&client_id=portal&username=[....]&password=[....]`	A European central bank
`openapi.[redacted]/erh/unlogin/ma_data_query?tradeHead=[...]&queryType=[...]&firstCoinCode=[..]&lastCoinCode=[..]&cardType=[...]&branchNo=[...]&startTime=&uuid=[...]&clientid=[...]&client_secret=[...]&callback=[...]&_[timestamp]`	An Asian bank

Figure 3. APIs exposing users' sensitive information

On some operating systems (OS), the browsing history — including these URLs — is shared among different devices such as a laptop, phone, and tablet. This can potentially weaken the efficiency of multifactor authentication and make it easier for hackers to get through by targeting the weakest device to steal client secrets contained in these URLs.[10]

We did not just find this particular issue in relatively new FinTech companies. We also found sensitive parameters exposed in the URLs of two central banks in Europe and a central bank in Asia, as well as other similar issues for more than 50 banks in Europe, Asia, and the U.S. The names of the said institutions in the list below were hidden to protect their respective customers.

Sector	Region	Information
A central bank	EU	Username, password
A central bank	EU	Username, password
A central bank	Asia	Username, password
Bank	BE	OAuth client secret
Bank	CA	Password
Bank	CN	Client ID, OAuth client secret
Bank	CN	Client ID, OAuth client secret
Bank	CR	Username, password
Bank	CR	Username, password
Bank	CW	Client ID, OAuth client secret, username, password
Bank	DE	OAuth client secret
Bank	ES	Username, password
Bank	ID	Username, password
Bank	IN	Client ID, OAuth client secret
Bank	IN	Client ID, OAuth client secret
Bank	IS	Username, password
Bank	IT	Client ID, OAuth client secret
Bank	IT	User ID, password
Bank	JP	Token
Bank	JP	Token
Bank	JP	Token
Bank	JP	Token
Bank	JP	Token, password
Bank	RU	Client ID, password
Bank	UK	Client ID, OAuth client secret
Bank	UK	Client ID, OAuth client secret
Bank	US	Username, password
Bank	VE	Username, password, email address
Credit card company	JP	Client ID, OAuth client secret
Credit card company	JP	Password
Credit card company	US	Client ID, API key
Financial regulator	EU	Username, password
FinTech company	AU	Client ID, OAuth client Secret
FinTech company	AU	Client ID, OAuth client secret, username, password

Sector	Region	Information
FinTech company	CN	Client ID, OAuth client secret
FinTech company	FR	OAuth client secret
FinTech company	FR	OAuth client secret, password, email address
FinTech company	JP	Client ID, OAuth client secret
FinTech company	JP	OAuth tokens
FinTech company	JP	Password
FinTech company	NL	Client ID, OAuth client secret
FinTech company	US	Client ID, OAuth client secret
FinTech company	US	Username, password
Investment company	CN	Username, password
Investment company	ES	Client ID, OAuth client secret, username, password
Micro credit company	IN	Client ID, OAuth client secret, username, password, IMEI code
Micro credit company	IN	Client ID, OAuth client secret, username, password, IMEI code
Money transfer service	UK	Username, password
Mortgage bank	DE	Username, password
Mortgage bank	JP	Username, password
Online payment provider	MT	Username, password, email address
Online payment provider	UK	OAuth tokens

Table 1. A selection of financial companies that expose confidential data in the URL path

In some cases, we observed a disparity in the security levels of APIs coming from a single bank across different services. For example, one reputable bank we investigated used a well-designed, secure API for exchanging normal banking data. However, the same bank also had a website for car dealers with a legacy API that exposed sensitive client data in the URLs.

When sensitive information is put in the URL paths that banks and FinTech companies use for their APIs, the extra security measures implemented on top of the basic ones, such as OAuth 2.0 protocol or multifactor authentication, may be reduced.

In another instance, we found that a European FinTech company with millions of customers published its API documentation online. While publishing API documentation is not uncommon, the API documentation makes it clear that the customer's email address, password, client secret authentication, and the client ID are visible in the path of the FinTech's API URL.

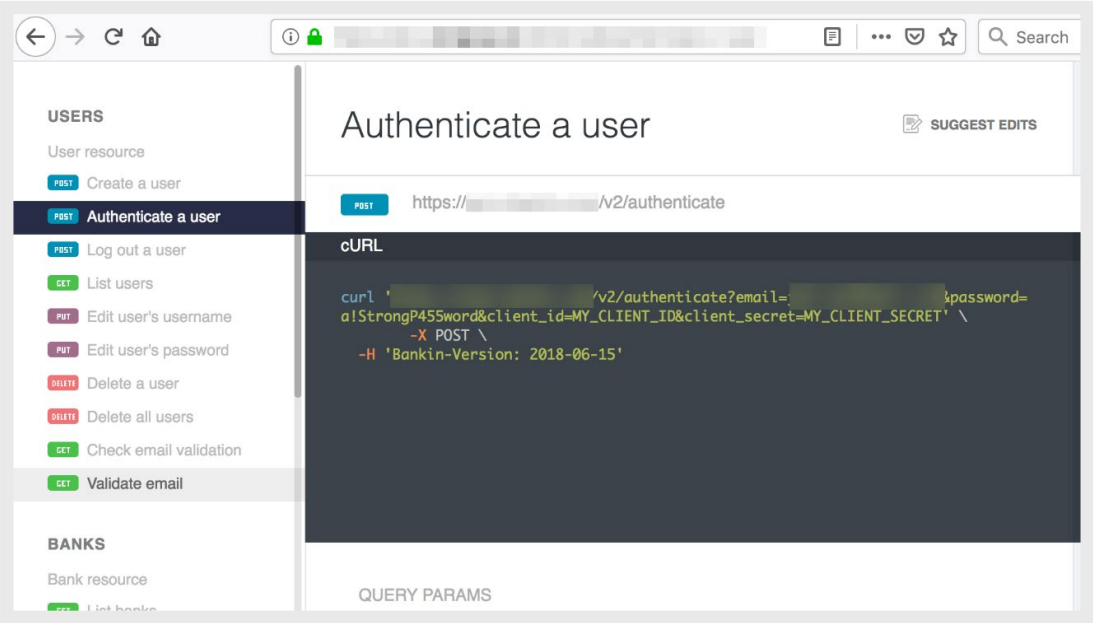

Figure 4. A screenshot of the online documentation of a FinTech company's API, claiming to be regulated by PSD2. Sensitive information can be clearly seen in the API URL path.

Banking Apps

Virtually all banks offer apps for Android and iOS devices today. This is natural as payment methods move from cash and cards to mobile devices and second-factor authentication for payments are initiated via a web browser. In the past, OTPs were transmitted via SMS, though banks are slowly moving away from this procedure for operational cost and security reasons. Instead, banks use apps that allow users to confirm the payments they initiated on another device such as a laptop. Mobile banking apps are expected to use second-factor authentication to secure transactions, but these have also been shown to have some problems. In 2017, a published research from the University of Birmingham found that some banking apps from leading U.K. banks had a couple of security issues due to incorrect handling of SSL.[11] Moreover, the apps also loaded unencrypted third party advertisements, making attacks possible should a user connect to a malicious public Wi-Fi. The banks were notified and have since addressed the vulnerabilities mentioned in the report.

Furthermore, many banking apps now depend on software development kits (SDKs) of third-party service providers. These SDKs may vary in use, from software for crash reports and app performance assessments, to advertising and tracking services. This means a lot of banking apps not only connect to banking servers, but also to APIs of third-party companies,[12] which also means the kinds of information being shared come with corresponding risks. For example, a bank with a third-party provider for an SDK that detects bugs and creates performance reports for banking apps can be compromised via phishing or by exploiting vulnerabilities in the provider's website. There have been instances in the past where default crash reports did not have any authentication protecting it.[13] Crash reports may contain customers' PII and even the exact line of banking app code where the application crashed. This is information that developers can use to make the necessary bug fixes, but malicious actors can also use it to look for vulnerabilities to exploit.

There can be differences between the dependency of banking apps on external SDKs , just like how different banking websites depend on external scripts. Whereas some banking websites do not depend on scripts or contents from external websites, other banking sites load a number of external scripts for tracking and advertising. A website that depends on external scripts may introduce additional security risks. When the third-party advertising or tracking service company gets compromised, the visitors of the banking site are at risk as well.

We looked into one of the biggest available platforms that help app developers better understand their users' app usage by collecting their user statistics and sending crash reports. App developers can use a SDK and integrate the third-party software into their apps. To see how this SDK works and what kind of data it sends, we wrote a simple iOS application to force a crash. The SDK uploaded the information of this crash, and we saw the third-party provider's dashboard include the phone model, amount of memory used, available disk space, OS version, date and time of the incident, session ID, app name, and the source code line number that generated the crash.

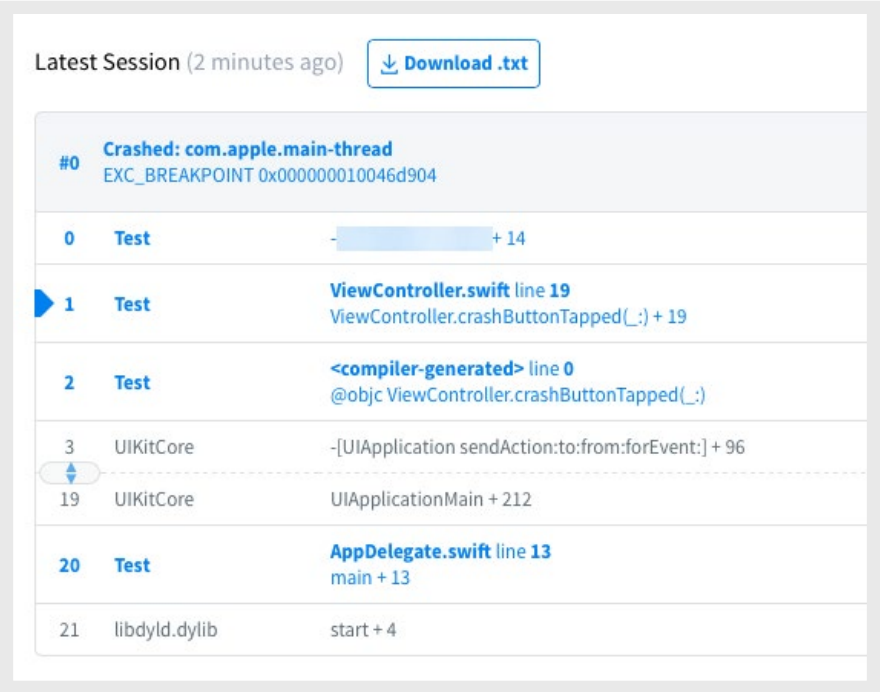

Figure 5. Screenshot of a crash report

From the crash report we obtained, we found that it also shared source code details to the third-party service provider. While this information allows developers to understand the bug that caused the app to crash, the risks are also evident if malicious actors see it. The information in the shared reports are not limited to crash events; it also includes PII.

Figure 6. Source code testing capability of sending PII

Crash reporting SDKs are also more integrated with advertising and tracking modules. If an attacker compromises the third-party provider of the crash report software, they can potentially access the same PII, introducing additional risks to bank app users.

In 2014, the app of a popular beverage company exposed user information by saving log files in clear text, leaking user credentials that would eventually give attackers access to their payment information.[14] In the same year, a man-in-the-middle (MiTM) vulnerability was found in a widely used crash analytics service's incorrect use of the OS' platform's SSL libraries.[15] In 2017, researchers found cross-site scripting (XSS) vulnerabilities in a mobile apps' analytics website where all the developers and app users' data is stored.[16] In 2018, another third-party service provider used by well-known apps appeared to have a code injection vulnerability.[17] These companies have since announced that they have resolved the flaws.

Open Banking generally aims to make mobile banking more secure for users, and in most countries, this is done by implementing strong authentication with strict regulation and screening of new FinTech companies allowed to access banking data after getting customers' consent. From a developer's perspective, banks using third-party SDKs for enabling more app functions makes more sense, but not necessarily without risks. Specifically, SDKs meant for tracking, advertising, and crash reports often used by banking apps add a layer of risk for the bank customer.

We had the chance to use a crash reporting service often used in Open Banking applications. For this particular service, we could not find a way to enable the two-factor authentication feature, which would make it easier for cybercriminals to perform credential phishing.

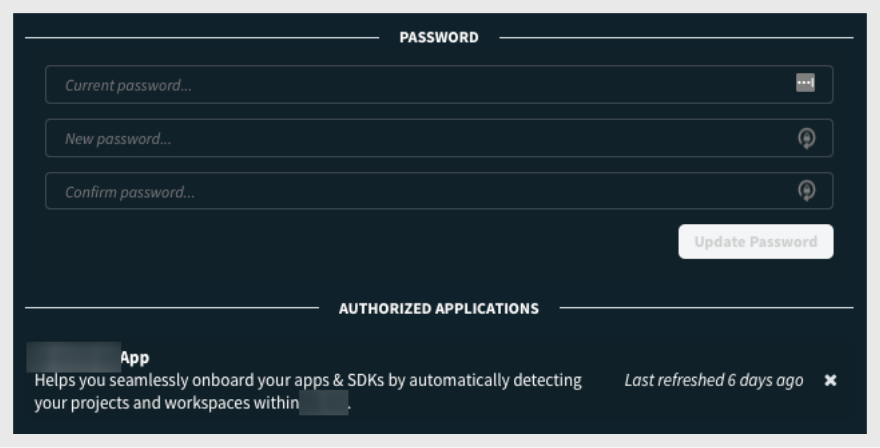

Figure 7. Screenshot of the login page of a crash report service used by banking apps

Previous Technologies: Screen Scraping and OFX

The main idea of Open Banking is not new. Parts of it were introduced in 1997 when Microsoft, Intuit, and CheckFree developed the Open Financial Exchange (OFX) protocol for exchanging financial data. Over time, other companies developed similar protocols, and they are still in use in different regions today.[18]

The OFX protocol is an open-source standard that allows applications to interact with banks and other financial institutions such as credit unions, brokerage houses, and FinTech companies. An app can use it to aggregate data from a user's different banking accounts so they can be managed by the same app. The website of a financial aggregate service provider can also use it to collect banking data on a customer's behalf. According to the OFX consortium website, the protocol is deployed by more than 7,000 financial institutions.[19] Based on internet-wide scans, OFX is only used in the U.S. and Canada by around 1,000 financial companies today, making it seem like OFX use is on the decline.[20] While newer versions of the protocol have already been released to add more functions, we observed that not all banks use the latest OFX versions:

```
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
```

Banks in the U.S., along with a number of Canadian banks, either deploy their OFX server on their own network infrastructure, or use a third-party service provider where webservers run OFX server software. The client application sends hypertext transfer protocol (HTTP) requests to this server and consumes its HTTP replies. The data is transmitted using either Standard Generalized Markup Language (SGML, for versions prior to 1.6), or via eXtensible Markup Language (XML) for the protocol's newer specifications. Using this scheme, client apps can send different types of transactions to the OFX server and the financial institution. Types of transactions may include wire transfers, bank statement retrievals, or stop checks, among a range of services that depend on the server-supported features.

Given the fact that OFX uses HTTP, it can take advantage of the current security layers that can be applied on top of HTTP, such as SSL and transport layer security (TLS). In addition, different authorization mechanisms were added to OFX specifications over time, including MFA in version 1.0.3 and OAuth in the latest version. However, the MFA method used here actually consists of security questions that ask for additional information like the user's mother's maiden name or place of birth. Version 1.0.3 does not support digital tokens or other more secure features present in other environments.

Similar to previous studies by Security Innovation in 2018,[21] we checked how many OFX servers were connected to the internet, their respective locations, and if they use the latest and more secure OFX versions. We collected OFX URLs of 721 different financial institutions from the OFX website,[22] and found that none of them were using the latest 2.2 version. The majority of them were using version 1.0.2 — a 22-year old protocol.

OFX Version Used	Count of OFX Version
CAN	**2**
1.0.2	2
US	**719**
1.0.2	564
1.0.3	151
2.0.2	4
Grand Total	**721**

Table 2. Financial institutions in the U.S. and Canada using old OFX versions

According to our telemetry, OFX servers are still in use, but banks did not update to newer versions for a long time. And while this presents obvious security risks, there have been no reports of big data breaches where OFX was abused. One possible reason for this is because banks do not offer OFX access for free to their customers.

FinTech companies not only use OFX for aggregating bank data, but also employ "screen scraping" as a technique. In screen scraping — also known as "direct access" — a FinTech company uses the bank customer's login credentials on banking websites, mimicking a web browser session as if they were a real customer. For this to work, the FinTech company parses through the HTML contents to extract the balance statements and initiate payments.

It is somewhat surprising to learn that this technique is still in use today, considering the risks introduced by screen scraping. For instance, with FinTech companies storing login credentials, there is no other way to do it unencrypted or use some kind of encryption that can be reversed. In another example, parsing account statements on banking sites is prone to errors when a bank introduces a new layout. Some FinTech companies acknowledge this on their websites: A change in the layout of the bank's website may cause balance statements to become unavailable in the FinTech company's app for days.

Screen scraping as a technique will make it harder for FinTech companies when PSD2 takes effect in the EU. As strong authentication is mandated, screen scraping will no longer be allowed. Yet dozens of FinTech companies have tried to organize political pressure to push back against the strict implementation of PSD2, claiming they were the ones who brought innovation to the banking industry and that there have been no known incidents or data breaches related to screen scraping.[23] They also claim that OAuth 2.0 is not user-friendly.

A strict interpretation of PSD2 requirements would obviously lead to a ban on using old OFX versions and screen scraping. However, there are calls for exemptions and postponing strong authentication in certain cases. At the time of writing, it is unclear if screen scraping will be banned in the EU.

Financial Grade API

Financial grade API (FAPI) is a protocol intended to be secure even in high-risk scenarios. For example, if an attacker steals a mobile app user's temporary access token, the attacker should not be able to use that token to access the victim's financial data. FAPI is currently being developed by the OpenID Foundation and the U.K. Open Banking Implementation Entity.[24] While intended for use once Open Banking takes effect in the U.K., it is not entirely ready; FAPI reportedly still has issues (which include significant security flaws) to resolve at the time of this paper's writing.

FAPI is intended to be a more secure version of OAuth 2.0, a widely used protocol that allows third parties to access a service provider's data without the need for the user to provide credentials. Instead of a password, the third party gets a token for accessing the user's data after the user gives explicit consent. The user controls the third party's access rights through the token, such as a temporary and read-only access. The user can revoke the token any time; in case it gets stolen, no password change is needed as no password was shared with the third-party provider in the first place. The OAuth protocol is typically used for importing emails from different providers into one account. Social media platforms also use OAuth.

OAuth 2.0 is regarded as a secure protocol, but it does not eliminate all risks. For instance, it does not eliminate advanced social engineering schemes where a rogue app is presented as something with extra features. Malicious actors have used this scheme in the past to gain access to users' mailboxes, and we have reported on past phishing attacks that involved the abuse of OAuth.[25]

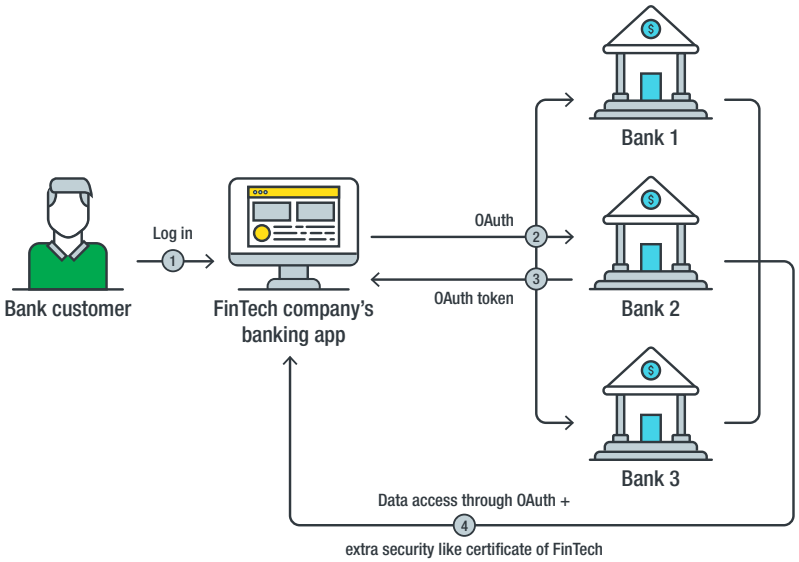

Figure 8. FAPI for Open Banking

The OAuth protocol is particularly useful in Open Banking, as it eliminates old and insecure methods that FinTech companies use to aggregate banking data from different accounts, such as screen scraping and old versions of OFX discussed in the previous sections. The U.K. Open Banking Implementation Entity planned to make OAuth even more secure in high-risk scenarios. The developers of FAPI have proposed several modules in the initial design to be implemented on top of OAuth 2.0, such as those applied for connections between clients (usually dedicated mobile apps or webservers of FinTech companies) and servers of banks. A distinction is also being made between confidential clients — webservers supposed to keep access tokens for a long time, for example — and public clients such as apps on a mobile phone that are not supposed to keep tokens for an extended period. Some of the proposed extra modules include:

- **mutual TLS (mTLS).**[26] In a secure encrypted connection such as HTTPS, the server uses an SSL certificate to authenticate itself to the client. In mTLS, the client also authenticates itself to the server, which means that the client will have to prove to the server that it has a private key. In Open Banking, a client would be a dedicated app or a website that belongs to a financial company.
- **OAuth 2.0 Token Binding (OAUTB).**[27] This will bind access tokens and/or authorization codes to a TLS connection. If an attacker manages to steal an authorization code, the attacker will not be able to use that code in another TLS connection as it is bound to one particular TLS session set up between the client and the server. Though this module sounds like a great idea, the support of token binding is lacking. For instance, it is currently not supported by any other internet browser aside from Microsoft Edge. While this in itself is not a real problem for Open Banking as OAUTB is meant to be used in connections between mobile apps and bank servers, the FAPI working group has temporarily removed OAUTB from the FAPI specifications in May 2019 due to a lack of implementation.

- **JavaScript Object Notation (JSON) Web Signature (JWS) Client Assertions.** The goal of JWS Client Assertion is to make sure only a particular confidential client can use an access token at the authorization server of a bank. This can be done by proving it has a token — a certain private key, for instance — that would give access to customer data that can only be used by a particular webserver, such as that of a FinTech company.
- **Proof Key for Code Exchange (PKCE).** This would be used to prevent an attacker from using an access token that is sent to a client — such as an app on a mobile device — that cannot keep a key for an extended period. The app will send an extra verifier to the authentication server to determine that it is the client who wants to have the access token, and not an attacker.

These extra modules were designed to make OAuth 2.0 secure in high risk scenarios. However, significant issues have surfaced. Among them, OAuth 2.0 Token Binding has been temporarily removed from the FAPI specifications because the FAPI working group deems there is a lack of support. Moreover, the FAPI standards are still in development and — as shown by German academic research presented during the IEEE Symposium on Security and Privacy (S&P) in May 2019 — it might not be able to deliver on what it intends to offer.[28] Among several attacks discussed by the researchers, we will describe the simplest attack routine below.

To understand the attack, here is a general use case of Open Banking:

- A banking customer ("Bob") is using a FinTech service to manage his household bills. This is possible as the FinTech service is one of the clients of Bob's bank, and he has authorized them to use their Open Banking API.
- To use this FinTech service, Bob must first authenticate with his bank to allow them access to his accounts.
- Bob logs in to the FinTech service site, which redirects him to the website of his own bank.
- Bob authenticates himself with his bank and the FinTech service gets an access token that allows them access to Bob's banking details, stored on a resource server of the bank.

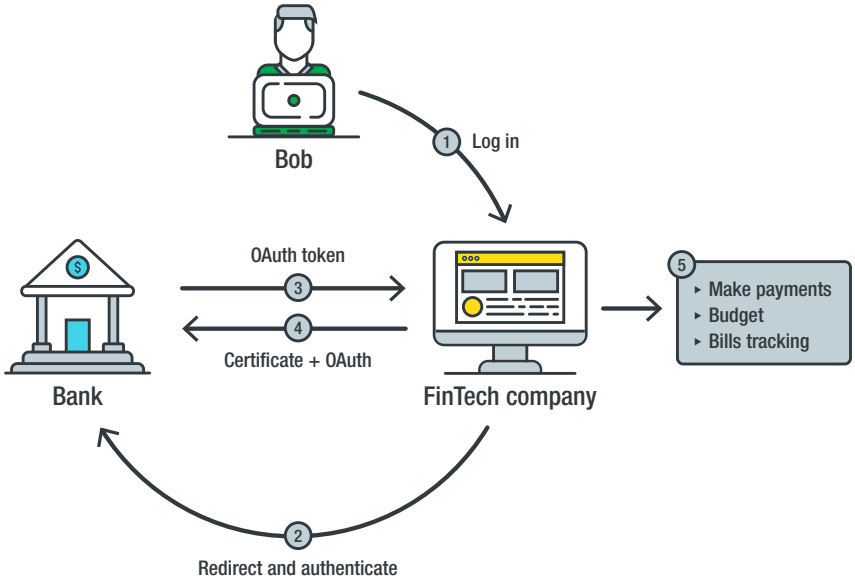

Figure 9. Open Banking in FAPI using mTLS as additional security feature on top of OAuth

This is the usual scheme for OAuth. But in FAPI, the FinTech service will also have to prove possession of a secret key associated with a certificate — the mTLS module on top of OAuth 2.0 — before the bank allows them to access the data. This is a certificate the bank grants to the FinTech service when they become a client of the bank.

Now we assume that the access token is phished by an attacker ("Eve"). The intention of FAPI in this scenario is to protect Bob's data and prevent Eve from accessing his banking details. Eve would not be able to immediately use the phished access token because she does not have the mTLS key from the FinTech provider that the bank is expecting.

- Eve could set up a fake bank or use a compromised bank ("Evil Bank" or "EB") and create a contract with the same FinTech service. This is less extreme than it may sound. With Open Banking, an app of a FinTech company potentially gives access to data of banks in different regions worldwide with different jurisdictions. FAPI was designed to anticipate high risk scenarios where there might not be an established trust relationship between banks from different regions in the world.
- Eve now logs into her account on the FinTech service and requests to authenticate with EB to access "her" bank details. EB returns to the FinTech service with Bob's phished access token, along with the URL of the resource server of Bob's bank.
- None the wiser, the FinTech service will use Bob's phished token to get his customer data and send it to Eve. This will work as the FinTech service can prove it is in possession of the mTLS key that Bob's bank provided.

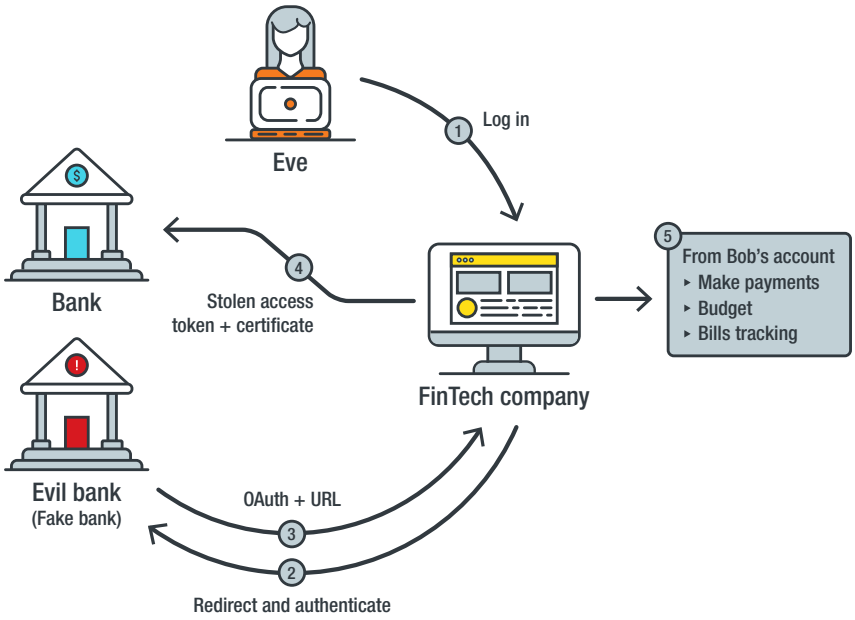

Figure 10. Attack scenario in FAPI of Open Banking

According to the intentions of FAPI, this should not be possible. Even though FAPI has been in development since 2017 and a form of FAPI has been implemented in the U.K., such attack scenarios were found in 2019. This shows that implementing the extra security on top of OAuth 2.0 is not yet ready and will not be that easy for banks in practice.

Risks and Predictions

The new banking paradigm with open APIs that access banking data leads to new opportunities for attackers. We will discuss some of the new risks that will be introduced by Open Banking, and describe the types of actors who will likely be interested in abusing and attacking the new system.

Banking data of customers alone is interesting for malicious actors: It contains valuable data on what products somebody has bought, income, personal preferences, and the data also can give quite precise information about the physical location of the account holder. For example, payments done in a store or coffee shop with a debit or credit card are shown almost in real time on account balance statements. We expect some advanced attackers to have a big interest in this data. In the past, advanced attackers have been compromising airline reservation systems, airlines and telecom providers likely because they are interested in tracking their targets.[29] By breaking into FinTech companies that are authorized to access data through Open Banking, actors can potentially track their targets as well.

Blackhat and greyhat actors may try to set up a FinTech company and do more activities with customers' banking data than their supposed purposes. Once banking APIs transfer the data to a malicious FinTech company, there is not much that banks can do to prevent abuse of the data. After PSD2 is implemented, banks will allow the FinTech company to access up to 24 months' worth of a customer's banking data once permission has been granted, which potentially means that a lot of valuable financial transactions and information could fall into the wrong hands.[30]

Malicious actors will surely try to compromise legitimate FinTech companies. We have seen some of the FinTech companies strongly advocate for screen scraping. From a security perspective, this is a cause for concern as customers will have to provide their bank credentials to the third party provider before they can scrape, and third party providers will have to store them for the long term. Compared to established banks, these FinTech companies may be easier targets for hackers.

Below are summaries of some of the attacks we foresee in the near future:

- **Attacks on the APIs**

Once the APIs are in place, the attackers can hack and abuse the servers themselves. Additionally, a successful distributed denial of service (DDoS) attack could result in downtime for payments or other banking operations. Since the API is public, a hacker could learn how the system works and find unexpected responses from the back end through trial and error.

- **Attacks on FinTech companies**

Not all FinTech companies will have the same security level and experience as a bank does, making them ideal targets for hackers. They usually tend to be smaller companies or start-ups with few employees. In a quick survey of five of these entities, we found them to have an average of 20 employees, none of whom was fully dedicated to security.

A company's servers are ideal targets from which to obtain customers' banking data. Several middleware providers already exist in the Open Banking arena, providing services to multiple FinTech companies. Compromising one of these companies is, in effect, a supply chain attack that can affect multiple FinTech companies, with neither the customers nor the FinTech companies becoming aware of the situation. Cybercriminals that can get to this information could monetize the stolen data — and the routines — in new and innovative ways.

Attacking third-party providers would make these attacks much more difficult for the banks to detect. Current anti-fraud systems can be fooled by having the attacker one layer removed from the bank. For instance, brute forcing a third-party provider would not alert the bank of an ongoing attack; banks would lack the information needed by their anti-fraud team because transactions and downloading bank data would be proxied by the third-party provider.

- **Attacks on the app or mobile platform**

Since most of these systems will be implemented as mobile apps, these can also become prime targets for attackers. Finding the username, passwords, or encryption keys within the app would allow an attacker to retrieve banking data and pose as the user. Like the user, the attacker would also be able to perform any banking operation, such as send payments on their behalf.

- **Attacks against the user**

Out of all the possible targets, end users will likely be the most vulnerable. Although this seems mild by comparison, the fact that third-party provider apps can now operate seamlessly on the user's behalf would mean that phishing of such apps can be done much more effectively and with more destructive results. Mass bank phishing emails tend to be less effective nowadays.

Phishing emails such as "We are Bank X, please click here to reset your password" have been around for many years, and users have become conditioned to be wary of them. However, something like "We are Third-Party Provider X of Bank Y, and we are upgrading your account, please click here" seems more innocent in comparison. In effect, this spreading of trust for sensitive bank logins also makes it easier to phish a user.

Banks have evolved their fraud detection methods for years in order to protect users and themselves by factoring in conditions like number of login attempts, user location, and computer settings, among others. With Open Banking, all of these fingerprints are a step removed from the bank: As their customers interact with new third-party providers, these new companies may have less mature anti-fraud systems in place.

Users are not aware of the fact that losing the password of their mobile app provider can be just as dangerous. As banks will not even be aware of these phishing emails (there is not much they can do about them anyway) these can open up new and more strategies for attackers.

The same can be said for any modern social engineering technique. Instead of trying to install a new fake banking app on a user's phone, attackers could try to install a new budgeting app that can capture a user's account credentials.

Another aspect of this evolving Open Banking world — and one that remains to be seen — is the increasing complexity of proving responsibility when a fraudulent transaction occurs. The fault can potentially lie with the bank, the user, or the third-party provider; how smoothly will communication between these three parties go to resolve any such incident?

Open Banking and the Criminal Underground

Our scan of the criminal underground found few mentions of the new law, and the few that we did see appear to come from more amateur criminals. Underground actors appear to have mixed views on the regulation, depending on how they see it affecting their respective businesses.

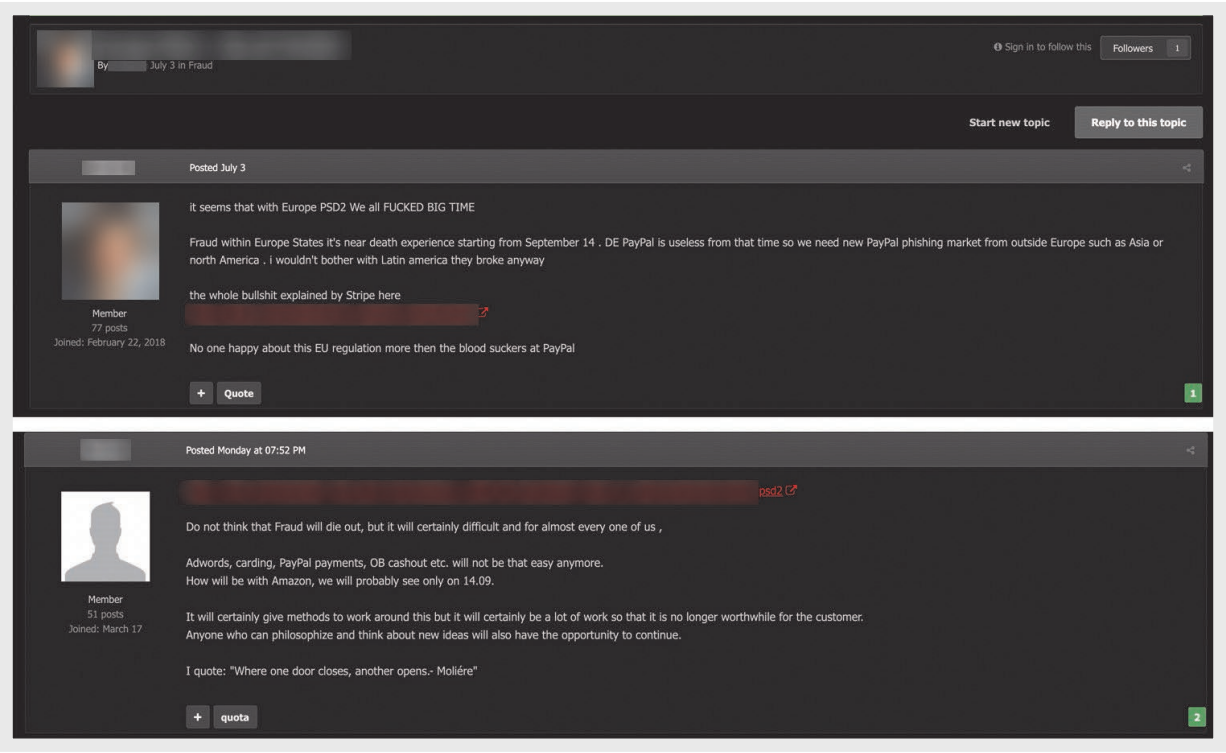

Figure 11. A post warning other members of the new law (top).
A comment on the post hopes that the status quo will be maintained (bottom).

Other members of the underground mentioned the new law's specific features and argue whether the changes can be implemented or not.

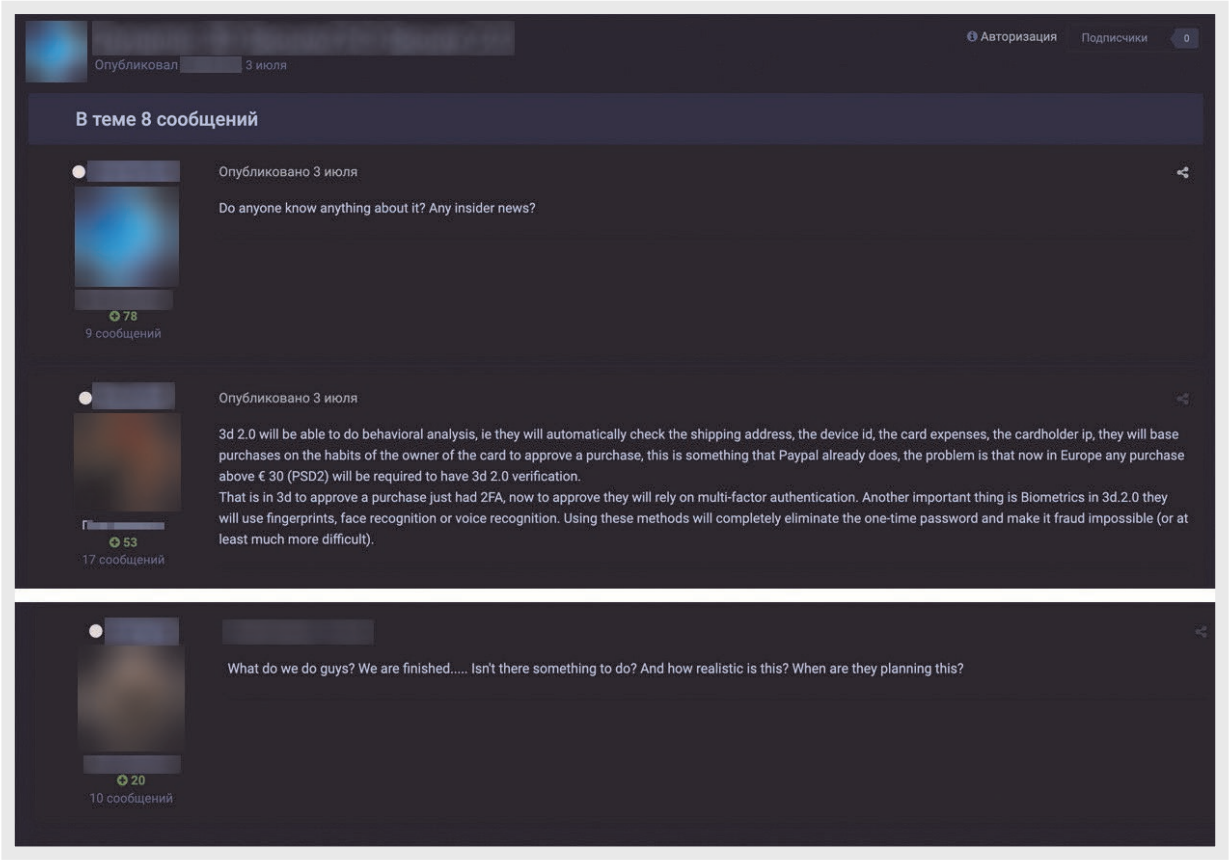

Figure 12. An inquiry on the new law and a reply consisting of copy-pasted information on one of the possible security features (top). Another inquiry on the companies' level of compliance (bottom).

Conclusion and Recommendations

This research paper looked into the status of Open Banking from a security standpoint. While the Open Banking initiatives promise better security for customers, the current implementation status reveal a couple of significant issues. First, there are conflicts between emerging FinTech companies eager to introduce new and innovative services and the traditional banks trying to protect their current positions. From a cybersecurity perspective, it is understandable for banks to insist on strict security requirements: Opening up banking data to third parties is far from trivial and could potentially lead to serious data breaches, abuse of data, loss of insight on money flow, and problems in detecting fraud.

We recommend that **FinTech companies embrace secure protocols** such as OAuth 2.0 and stop using risky techniques such as screen scraping and outdated protocols like OFX. The lack of documented incidents involving these methods do not refute the fact that they are highly insecure.

We also discovered potential problems with banks. We found a significant number of banks and financial companies putting sensitive information like usernames, passwords, client secrets, tokens, IMEI numbers, and transaction data in the path of call back API URLs, potentially nullifying the extra security measures taken in Open Banking. **Sensitive information should never be in URL paths**. Even when sent over TLS, attackers can still use a number of ways to obtain this data, leaving customers vulnerable.

We also see risks in many modern banking apps that share performance data, crash reports, and PII with third-party providers. For banks and FinTech companies, we recommend **keeping banking apps as clean as possible: only share data with servers of banks**. This can be compared to the practice of several banks' online login portals that are not dependent on any tracking, scripting, or advertising from third-party websites.

The financial sector has long been the most targeted by cybercriminals who go to great lengths to find vulnerabilities in all aspects of the customer-bank interaction process. Open Banking app developers must **develop software that is secure by design and employ regular external (or thorough in-house) security audits** on all aspects of the software.

In fact, FinTech companies should be looking to go beyond secure code reviews and **design their applications to run safely on an already hostile and compromised environment**. Software that adds very limited additional risks (such as using an isolated container environment) when the app user is compromised is a good security benchmark to aim for and could be a key competitive advantage compared to other offers.

Open Banking apps can provide numerous advantages and conveniences that did not exist before. However, it is important for customers of Open Banking apps to **thoroughly educate oneself on the reliability, trustworthiness, and security culture of any company entrusted with access to banking data**. Additional checks and research will be necessary before installing any app to make an informed decision.

From a security standpoint, our assessment of Open Banking is that it's not yet complete. It is important for all parties involved to get the extra security safeguards implemented as soon as possible. Even when all planned security measures of Open Banking are in place, we expect that the added convenience it affords global banking users will be met with an equivalent increase in ingenuity from cybercriminals.

References

1. *EUR-Lex*. (3 August 2017). "'What is the aim of the directive?' and 'Key points' of the 'Revised rules for payment services in the EU.'" Last accessed on 20 August 2019 at https://eur-lex.europa.eu/legal-content/EN/LSU/?uri=CELEX:32015L2366.

2. *EUR-Lex*. (25 November 2015) Number 30, Article 4 of "EU Directive of the European Parliament and of the council.'" Last accessed on 20 August 2019 at https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32015L2366.

3. *EUR-Lex*. (27 November 2017). Article 5 of "Commission delegated regulation (EU)." Last accessed on 20 August 2019 at https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.069.01.0023.01.ENG&toc=OJ:L:2018:069:TOC.

4. Warwick Ashford. (13 February 2018). *Computer Weekly*. "FS-ISAC enables safer financial data sharing with API." Last accessed on 20 August 2019 at https://www.computerweekly.com/news/252434931/FS-ISAC-enables-safer-financial-data-sharing-with-API.

5. *Australian Competition & Consumer Commission*. (n.d.). "Consumer data right." Last accessed on 19 August 2019 at https://www.accc.gov.au/focus-areas/consumer-data-right-cdr-0.

6. Sohn Ji-young. (16 April 2019). *Korea Herald*. "Korea to launch 'open banking' system at year-end." Last accessed on 16 August 2019 at http://www.koreaherald.com/view.php?ud=20190416000660.

7. Graham Rothwell. (27 September 2018). *Accenture*. "The brave new world of Open Banking in APAC: Singapore." Last accessed on 16 August 2019 at https://bankingblog.accenture.com/brave-new-world-open-banking-apac-singapore.

8. Graham Rothwell. (16 October 2018). *Accenture*. "The brave new world of Open Banking in APAC: Japan." Last accessed on 27 August 2019 at https://bankingblog.accenture.com/brave-new-world-open-banking-apac-japan.

9. *Tink*. (14 June 2019). "The sobering September review: Banks' PSD2 APIs far from ready." Last accessed on 16 August 2019 at https://tink.com/blog/2019/06/14/psd2-updated-sandbox.

10. Vincent Haupert and Stephan Gabert. (n.d.). *Friedrich-Alexander University Erlangen-Nürnber*. "Short paper: How to attack PSD2 Internet Banking." Last accessed on 16 August 2019 at http://fc19.ifca.ai/preproceedings/31-preproceedings.pdf.

11. Tom Chothia, et. Al. (n.d.). *University of Birmingham*. "Why banker Bob (still) can't get TLS right: A security analysis of TLS in leading UK banking apps." Last accessed on 26 August 2019 at https://fc17.ifca.ai/preproceedings/paper_83-2.pdf.

12. *AppSearch*. (n.d.). "Finance Apps Category." Last accessed on 16 August 2019 at https://search.appcensus.io/category/Finance/1.

13. Michael Cobb. (9 October 2018). *Tech Target*. "How was Google Firebase security bypassed?" Last accessed on 16 August 2019 at https://searchsecurity.techtarget.com/answer/How-was-Google-Firebase-security-bypassed.

14. Daniel Wood. (13 January 2014.). *Seclists*. "[CVE-2014-0647] Insecure data storage of user data elements in Starbucks v2.6.1 iOS mobile application." Last accessed on 16 August 2019 at https://seclists.org/fulldisclosure/2014/Jan/64.

15. Adrian Mettler et. Al. (20 August 2014). *FireEye*. "SSL vulnerabilities: Who listens when Android applications talk?" Last accessed on 16 August 2019 at https://www.fireeye.com/blog/threat-research/2014/08/ssl-vulnerabilities-who-listens-when-android-applications-talk.html.

16. Eshwargetenv. (10 July 2017). *WeSecureApp*. "Fabric.io API permission apocalypse – Privilege escalations." Last accessed on 16 August 2019 at https://wesecureapp.com/2017/07/10/fabric-io-api-permission-apocalypse-privilege-escalations/.

17. Mdv. (14 November 2018). *HackerOne*."Possibility to inject a malicious JavaScript code in any file on tags.tiqcdn.com results in a stored XSS on any page in most Uber domains." Last accessed on 16 August 2019 at https://hackerone.com/reports/256152.

18. *Afinis*. (n.d.). "IFX Forum… is now Afinis." Last accessed on 28 August 2019 at http://www.ifxforum.org/.

19. *Open Financial Exchange*. (n.d.). "About OFX." Last accessed on 16 August 2019 at http://www.ofx.org/about-ofx.html.

20. Steven Danneman. (15 November 2018). *Security Innovation*. "Your bank's digital side door: Is it opening you up to risks?" Last accessed on 26 August 2019 at https://blog.securityinnovation.com/digital-side-door.

21. Steven Danneman. (15 November 2018). *Security Innovation*. "Your bank's digital side door: Is it opening you up to risks?" Last accessed on 26 August 2019 at https://blog.securityinnovation.com/digital-side-door.

22. *OFX Home*. (n.d.). "Welcome to OFX Home." Last accessed on 16 August 2019 at http://www.ofxhome.com.

23. (n.d.). "Manifesto for the impact of PSD2 on the future of European Fintech." Last accessed on 26 August 2019 at https://www.futureofeuropeanfintech.com/assets/Manifesto-for-the-impact-of-PSD2-on-the-future-of-European-Fintech.pdf.

24. *OpenID*. (n.d.). "What is the Financial-grade API (FAPI) WG?" Last accessed on 26 August 2019 at https://openid.net/wg/fapi/.

25. Feike Hacquebord. (25 April 2017). *Trend Micro*. "Pawn Storm abuses Open Authentication in advanced social engineering attacks." Last accessed on 16 August 2019 at https://blog.trendmicro.com/trendlabs-security-intelligence/pawn-storm-abuses-open-authentication-advanced-social-engineering-attacks/.

26. B. Campbell et. Al. (3 July 2019). *OAuth Working Group*. "OAuth 2.0 mutual TLS client authentication and certificate-bound access tokens." Last accessed on 26 August 2019 at https://tools.ietf.org/html/draft-ietf-oauth-mtls-15.

27. M. Jones et. Al. (19 October 2018). *OAuth Working Group*. "OAuth 2.0 Token Binding." Last accessed on 26 August 2019 at https://tools.ietf.org/html/draft-ietf-oauth-token-binding-08.

28. Daniel Fett et. Al. (2019). *2019 IEEE Computer Society*. "An extensive formal security analysis of the OpenID Financial-grade API." Last accessed on 26 August 2019 at https://www.computer.org/csdl/proceedings-article/sp/2019/666000b054/19skg9V4UjS.

29. Sarah Hawley et. Al. (29 January 2019). *FireEye*. "APT39: An Iranian cyber espionage group focused on personal information." Last accessed on 16 August 2019 at https://www.fireeye.com/blog/threat-research/2019/01/apt39-iranian-cyber-espionage-group-focused-on-personal-information.html.

30. (n.d.). *ING*."Vertel me: Welke bedrijven kunnen bij mijn betaalgegevens (Tell me which companies have access to my payment details)" Last accessed on 16 September 2019 at https://www.ing.nl/particulier/mobiel-en-internetbankieren/campagne/PSD2/PSD2_Welke_bedrijven_kunnen_bij_mijn_betaalgegevens_index.html.

TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threats techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com

TREND MICRO™ | research