Open Banking

Security Considerations & Potential Risks

Prepared by: Matthew Pettitt

Table of contents

Executive summary

What is Open Banking?

The UK banking market has been dominated by a relatively small number of large banks, all of which provide basically the same services, and which have traditionally been subject to strong brand loyalty, partly due to perceived difficulty in switching banks. The Competition and Markets Authority (CMA) undertook an investigation in 2016, and one of the outcomes of this was to suggest that there should be a standardised way of providing banking information: both in terms of account features for customers looking for a new bank, and in accessing information about existing accounts, to aid with budgeting and choice. The Open Banking implementation entity was formed to guide this process, developing the APIs, messaging and security standards required.

The initial release of Open Banking, aimed at providing account information in a standard way, happened in March 2017. This comprised six API definitions, providing standard data about branches, ATMs, personal and business current accounts, loans, and credit cards. These APIs are read-only from the point of view of end users, and can be used by anyone to build applications and websites. As a result, from a security perspective, they are essentially equivalent to the bank public information websites.

The second release however, comprises a set of Read/Write APIs that allow customers to provide third parties with access to their bank account details, and even to allow third parties to take payments from their accounts. Obviously, the ability to access details of a customer account requires high levels of security, and all users of these APIs must be registered with the Financial Conduct Authority (FCA) as a specific type of business, relating to how they intend to use the data. These APIs are therefore almost equivalent to a bank's online banking websites, with a similar level of security required.

These were supposed to launch for the nine largest banks (known as the CMA9[1], and comprising AIB Group, Bank of Ireland, Barclays Bank, HSBC, Lloyds Banking Group, Nationwide Building Society, Northern Bank Limited, trading as Danske Bank, Royal Bank of Scotland Group, and Santander UK) in full on 13th January 2018. However, five of these were granted extensions in December 2017, ranging from six weeks to a year, to complete the process. One was granted a last minute extension, meaning that only Allied Irish Bank, Danske and Lloyds Banking Group actually launched on time. Of these, only Lloyds Banking

[1] https://www.openbanking.org.uk/about-us/glossary/#cma

Group offers personal bank accounts, with all three offering business accounts making use of the APIs.

Since this date was the first time third party providers were able to access actual bank implementations of the APIs, there were no immediate changes to banking aggregation applications like Yolt[2], Money Dashboard[3], or Emma[4]. However, the launch enabled these types of provider to move towards testing against live implementations with real data, rather than reference implementations with fixed output data. It is likely to take several months for these services to start to use the Open Banking APIs rather than their current methods, which mostly involve logging into online banking services and scraping the required data. The FCA stated that approximately 40 providers were approved and registered in the three weeks following the launch date.

Open Banking in the UK has been aligned in many ways with the EU's Payment Services Directive 2 (PSD2) regulations, which specified requirements for banking providers to provide access to data, but did not go as far in defining the specific interfaces. As a result, the "Berlin Group", comprised of representatives from a range of banks across the EU has developed an alternative API covering many of the same features under the title NextGenPSD2. Given that many banks operate both in the UK and in the rest of the EU, it is likely that at least some banks will be implementing both the UK Open Banking APIs and an EU equivalent, although these are unlikely to apply to the same accounts. Under PSD2, the banks must provide an interface with a specified level of uptime, which provides a specific amount of information,

Other banks operating in the UK can optionally implement the Open Banking APIs too, although they can alternatively choose to implement PSD2 in other ways, such as by providing access to a custom API. Metro Bank[5], for example, have stated that customers will be able to authorise third party company access, but note that it will be through the Open Banking APIs. The Co-operative Bank[6] has taken a more hands-off approach, simply stating that "we cannot hold you responsible for unauthorised transactions if you have shared your credentials with authorised AIS and PIS providers", but otherwise provides no details on how the PSD2 requirements will be met.

NCC Group has been working with a number of providers to ensure that appropriate security is both built into the specifications, and actively applied within specific implementations, both in the bank-specific and in the third-party facing sections.

Misconceptions: What Open Banking is not

There have been a number of misconceptions about what access Open Banking will provide

[2] https://www.yolt.com/
[3] https://www.moneydashboard.com/
[4] https://emma-app.com/
[5] https://www.metrobankonline.co.uk
[6] https://www.co-operativebank.co.uk/

to end users and to third parties. The key points here are that end users cannot sign up to directly access the bank-provided APIs, and that there is no obligation to provide access to any third party in this way.

Only registered businesses can be registered with the FCA, and hence with the Open Banking Directory. It is not possible for an individual to obtain access, even to their own accounts, without first setting up a business structure and following all the FCA reporting requirements. This is a non-trivial undertaking, with substantial ongoing requirements. Similarly, a business cannot register for access to the Open Banking Read/Write APIs without being FCA registered. This means that any access to Open Banking APIs for non-regulated businesses must involve a third party, which may limit the use for businesses in sensitive sectors. There is also a requirement that businesses with access to transaction details have insurance to cover any losses caused by their negligence.

Similarly, while it would in theory be possible for, for example, a supermarket chain to register as a third party provider and to send disapproving messages to mobile devices if they see a transaction for a rival store in a customer's banking history, this would require a number of steps from the user. First, the user would have to approve the supermarket's access to their detailed transaction history, then the supermarket would only be able to access the full details a limited number of times per day, unless the user was actively using whatever application to which the permission was linked. The supermarket would not be able to get this data as a result of receiving a payment from the user, or by making a request to the bank without the user's approval.

The "open" part of Open Banking is less about making the bank details themselves open, and more about making the process of obtaining information open. Rather than online banking details being passed to unknown processing partners, it should be possible to identify precisely which companies have been granted access to account details, both at the time of giving permission, and at any later point from a trusted bank interface.

It is also important to note that not every function of the full Open Banking system will be supported by all third party providers. The most obvious distinction here is the difference between payment providers and account information providers, but there are also distinctions between the levels of access that can be requested by information providers. Credit and debit transactions require distinct privileges: it is possible for a provider to request access to outgoing payments, without being able to view details of incoming salary payments, for example. The ability to tell who the recipient of a standing order is requires a higher level of privilege than the ability to tell the amount of a standing order, and there are other similar divisions for account details and beneficiaries.

The precise details that are requested are down to the third party provider, meaning that, just as with mobile app stores, it may be that providers request more permissions than are actually required, and the choice for the end user becomes "grant lots of permissions" or

“don’t use this application”. However, given that without using Open Banking APIs, the options tend to be not using the application, or granting full access to the online banking system, this is still an improvement.

The “open” part of Open Banking is less about making the bank details themselves open, and more about making the process of obtaining information open.

Section 2 – Security Implications

What are the risks?

In order to understand the potential risks of the Open Banking system, it is first important to understand what the system lets users do. In short, a user can visit a third party website, authorise it to view account details, which may include details on balances, previous transactions, future transactions, standing orders and direct debit payments, and payment recipients, and then authorise specific payments to the third party.

Depending on the specific details, this information could be considered as highly sensitive to some users, while effectively public for others – the Open Banking system is designed for both business accounts and personal accounts.

For example, if a publicly traded business exposes the account balance, this is likely to be roughly in-line with the published account figures for the previous accounting period. However, an individual may want to keep their savings balance private, and may not want details of some transactions to be exposed to any third parties – especially where this may cause embarrassment or result in harassment.

It is also important that some terms used within the Open Banking specifications are defined:

- ASPSP – Account Servicing Payment Services Provider. Usually a bank or building society, but defined as any provider of accounts which can be used as the source of banking data.
- AISP – Account Information Service Provider. An API consumer that aims to provide information about the accounts to end users. A service that groups all bank accounts from multiple providers into a single dashboard would be an AISP, as would a service that sorts transactions into groups like “utility bills” and “eating out” for budgeting purposes, whether this applies to a single account or multiple accounts.
- PISP – Payment Initiation Services Provider. An API consumer that has the permission to request payments to be made from one account to another. This applies to both transferring money between accounts owned by a single end user, and to making payments to a third party account.
- TPP – Third Party Provider. Any business or person who is registered as an AISP, a PISP, or as both.

- PSU – Payment Services User. An actual person who is using a service built on the Open Banking APIs.

Clearly, the risks are different for ASPSPs, AISPs and PISPs, and can apply to both providers and end users of the system.

ASPSP risks

From the point of view of an ASPSP, implementing the Open Banking Read/Write APIs is effectively providing a wrapper over pre-existing functionality. Obviously, banks previously had a requirement to keep track and somehow display transactions – this pre-dates internet banking, with paper statements or transaction books. Similarly, internet-banking systems tend to expose details of direct debits, standing orders and other payment recipients. The ability to create a payment is also nothing new, but the Payment Initiation API changes the flow, as it is no longer the account holder requesting the payment directly.

As a result of this, the first set of risks for ASPSPs are almost entirely equivalent to those for internet banking providers: ensuring that only the account holder and designated third parties can access the account details, ensuring that anyone who can access the account cannot make unauthorised changes (e.g. increasing the balance of the account arbitrarily), and keeping track of all actions performed for auditing purposes. However, since the aim of the system is to allow third parties to access specific details, these risks need to take the potential for third party interactions into account. For example, if a TPP has permission from multiple PSUs to view account details, it is important that the correct details are provided for each request. The impact of this could range from confusion, should an account balance shown to an end user be wildly different to that they are expecting, to a full breach of confidence, if the wrong account details were shown to an end user, exposing another user's personal information.

Including the Payment Initiation API functions further complicates this. While the currently defined specification aims to allow a PISP to initiate a single, immediate, domestic payment made in GBP, the plan for future expansion of the APIs allows for more complex payment methods. These range from payments in other currencies, recurring payments (e.g. subscription payments), future dated payments (e.g. setting a payment to go out after salary has been paid into an account), and even making payments from multiple accounts to a single recipient, as a single transaction. The risks for immediate payments include both financial risks to the ASPSP (what happens if multiple payments are initiated simultaneously, but where the combined amount exceeds that available in the account?) and to the end user (if a payment is made to an incorrect recipient due to a mistake in the application logic), and get more complicated when further options are provided. If a future payment to be made in another currency is scheduled, but the currency exchange rate changes dramatically for instance, how should this be handled?

In terms of handling mistakes and fraudulent transactions, the banks have the majority of the responsibility, as long as the third party is fully regulated. Upon being notified of a fraudulent transaction, the banks are supposed to refund the end user immediately, and then attempt to recover funds from the involved third party, rather than requiring that the customer wait until the bank has recovered the funds from the third party. This changes the process from a customer chasing a third party for return of their funds to a bank chasing the third party, with the corresponding change in funding and legal awareness, which should help with fund recovery. The maximum individual liability for transactions before telling the bank also drops from £50 to £35. This also presents a risk to the bank in the case of a fraudulent claim of fraud – where an end user claims that a third party has not provided a service, or has caused loss, when this was not actually the case.

The banks also face a risk in the form of the CMA and FCA wanting to ensure that security concerns are not used as an excuse for delaying the roll out of Open Banking APIs[7]. While the FCA does not want insecure implementations released, there is also limited patience in terms of delays caused by failed security assessments. This may result in some implementations needing to change internally over time, while maintaining the outward facing endpoints. In turn, this may result in complications due to interactions between functional and security related testing – in theory, there should not be any changes to the data returned following this type of change, but in practice, this is not always the case.

AISP and PISP risks

Third party providers wanting to offer account management and advice services have previously been limited to either requesting internet banking details from prospective users, then effectively screen-scraping the sites, or building distinct methods specifically handing each account provider (or utilising a third party supplier which handles one of these options). Until recently, providing internet banking login details was against most bank's terms and conditions, and the quality of bank specific APIs has been highly variable. As a result, most providers stuck to gathering limited data, restricting the potential impact of a data breach.

With the ability to collect all transactions from a given account, it is likely that third parties will have more data stored, which could increase the risk of data breaches. However, there is a requirement that AISPs wanting to use the Read/Write APIs are registered with a Competent Authority to perform services under the Payment Services Regulations or under the Electronic Money Regulations. This is the Financial Conduct Authority at the time of writing, but the guidelines have been written to take changes of name or function into account. Part of the requirements for being registered with the FCA is a commitment to periodic systems testing by a competent third party, but even with regular penetration testing it is possible for vulnerabilities to be introduced in software deployment or via systems that were excluded

[7] https://uk.reuters.com/article/uk-britain-fca-bailey-banks/uk-watchdog-tells-banks-not-to-use-security-concerns-to-stifle-competition-idUKKBN1FR2TL

from the testing scope.

AISPs also need to be aware of information leakage to other ASPSPs. Where a consolidation function is being used, it is important that account details are not exposed to other ASPSPs, even if data from multiple providers is being shown in a single view. This could be a particular problem where multiple end users are allowing a single AISP access to accounts, such as for a household budgeting application, where details from individual and joint accounts may be shown together.

Payment initiation is another complex area. To date, most online payments in the UK have either been through credit card processing gateway services, through direct credit card payments, or via services such as PayPal or Amazon Payments, which act as a buffer between the bank and the payment recipient. For the majority of smaller businesses, this is likely to remain an attractive option, since it avoids the requirement to register as a PISP, even if the intermediary services start to support Open Banking as a payment method.

For those businesses that decide to register as a PISP in their own right, however, this may involve increased interaction with banks. Since each ASPSP implements the same APIs, this should not massively increase the complexity of the system, but is likely to involve an increased number of participants compared to current solutions.

TPPs must also ensure that any log files stored as part of normal website activity do not include sensitive details, and that certificates used for authentication with banks are stored securely. These should not be new requirements, but there is always risk with adding new functionality to existing applications, and the potential impact of losses relating to banking data may be greater. Implementing a secure software development lifecycle is one way to minimise the risks of mistakes in application development, but is not sufficient as the only protection.

PSU risks

The Open Banking ecosystem has been designed to be safe for end users, with Open Banking Limited being intent on showing that security has been one of the key points with documents such as their “Background to Open Banking” document[8]. This emphasises the requirement for TPPs to be registered with the FCA, the requirements for all companies involved to comply with Data Protection laws, and the ability for end users to provide explicit consent for access, which can be revoked at any time.

However, there are always potential dangers when relying on end user consent. Users may not read the full descriptions of what permissions they are giving to the third party, or provide access to excessive permissions because they are requested, even when they may not be required – an end user does not have the option to deselect specific elements requested.

[8] https://www.openbanking.org.uk/background-to-open-banking/

There are also some discrepancies in the documentation between the behaviour of the APIs and the descriptions of the behaviour shown: most notably that the consent model guidelines show the “transactions” permission as providing details of transactions for the last twelve months, while the API is defined as providing all transactions for a given account, with no time limit.

It may also be possible for phishing style attacks to be performed, with the aim of obtaining end user banking credentials. This would be a particular risk for mobile apps, where the URL of visited pages is not always clearly displayed. This risk could be partially mitigated with multi-factor authentication, although an attacker mimicking an Open Banking authentication process may be able to pass valid 2FA details supplied by the user to the legitimate site as they are provided.

While third parties in the Open Banking ecosystem have to be registered with the FCA, there is no obligation that they are also impartial. Therefore, end users have to be aware that third parties may not always have their best interests in mind. This is not a specific risk introduced by Open Banking, although since users have to interact with their banking provider during the initial process, there is a risk that more trust is given to TPPs than would be for other providers. As with any other commercial transaction, though, users should be aware of details such as whether particular suppliers have commercial relationships with advisors, or whether there are financial incentives for an advisor service to recommend a specific supplier.

Newspapers have also been keen to highlight the potential risks to end users, with headlines such as “’Open Banking’ revolution will leave account holders at mercy of ‘hackers and thieves’, banks warn”[9] and “Open banking? I think I’ll be keeping my door shut”[10]. In some cases, these articles appear to stem from a fear of third parties being able to get account information without user interaction – a problem that may have roots in branding, rather than reality. Other potential problems include those mentioned already: phishing sites, malicious third parties gaining access to the system, and insecure development processes leaving potential holes in TPP applications.

Third party TPPs

As mentioned previously, account aggregation services that existed before the launch of Open Banking have mostly been restricted to either accessing data by pulling it from the normal customer online banking systems, which, depending on the bank, could require end user interaction to be able to update balances or transaction details, or building bank specific interfaces. Other providers have out-sourced the data collection to third parties such as

[9] http://www.telegraph.co.uk/news/2018/01/12/open-banking-revolution-will-leave-account-holders-mercy-hackers/
[10] https://www.theguardian.com/money/blog/2018/jan/13/open-banking-app-data

Yodlee[11], Eurobits[12] or OpenWrks[13], who specialise in handling any changes to the data provided from the banks and wrangling it back to the expected format for end users, usually providing a standardised API for all responses, no matter which bank provided the original data.

Since these providers already allow for access to more banks than the Open Banking APIs, it seems likely that many services may stick with the third party providers, rather than becoming Open Banking AISPs in their own right. Where these providers merge data obtained from Open Banking APIs and legacy approaches, there is potential for consumer confusion: why do some accounts need the full online banking login details to be provided to this third party, while others direct the end user to the bank's own site? In some cases, there may also be data protection issues with using third party providers. Yodlee, for example, is a US company, so may not meet all requirements for holding data which EU companies are subject to.

These services may also be a source of confusion for users. Since the aggregation service would be the TPP, from the point of view of the bank, it is important that the interface shown to end users makes it clear that the aggregation service is acting on behalf of the company that the user was interacting with. This potential risk has been identified within the Open Banking specifications, with detailed guidelines available on the consent model that should be used[14]. However, it may not always be fully clear to end users precisely which entities they are dealing with, or which of those entities are fully regulated with the FCA. This information could be important if the user wants to revoke permissions from a specific TPP. It is currently unclear how an end user could revoke permission from a regulated TPP that acts as an intermediary for multiple end services: imagine revoking access from one shop that uses PayPal, without affecting other payments going to other shops from the same PayPal account.

Existing risks

There are also security risks which are not new to Open Banking, but which potentially gain potency when combined with the type of information being exposed. These could include the use of externally hosted scripting libraries on third party sites. While the third party provider would be registered with the FCA and with the Open Banking Directory, and hence subject to regular security assessment, hosts of external script files would not be. Therefore, should the external host be compromised, details could potentially be extracted about end user accounts. Obtaining the bank account number and sort code, along with the name and address of the account holder, would be sufficient to set up Direct Debit payments, for example. While it is

[11] https://www.yodlee.com/yodlee/emea/
[12] https://eurobits.es/
[13] https://www.openwrks.com/
[14] https://openbanking.atlassian.net/wiki/spaces/DZ/pages/23429879/Consent+Model+Guidelines+-+Part+1+Implementation

unlikely to result in a direct benefit to the attacker, due to the requirements for accepting Direct Debit payments, it could certainly cause disruption to the victim. Similar attacks could occur without the need for an external hosting partner if cross-site scripting vulnerabilities exist within the TPP application, or if the storage of retrieved data is insufficiently secure.

While banks have been subject to regular security assessments for a long time, and tend to have incorporated the requirement into development processes, companies wanting to follow "release early, release often" type deployment strategies may find this difficult to incorporate.

Section 3 – Process Flow

How should an Open Banking transaction work?

The intended flow for an Open Banking account information transaction involves at least three parties: an end user (PSU), a third party provider (AISP) and a banking provider (ASPSP). The third party provider prompts the end user to allow access to account information, explaining what information will be accessible if permission is granted. The third party also needs to identify the bank from which the end user wishes to provide information.

The third party then connects to the bank to create an "account-request", which informs the bank that one of their customers wants to grant access to the third party, although not, at this stage, which customer. It does tell the bank what permissions are being requested, and potentially an expiration date for the access. The bank then responds with an AccountRequestId, which acts as an identifier for the request through the subsequent steps.

The third party then redirects the user to the bank's authentication system, including the AccountRequestId in the redirection. The end user logs into the authentication system provided by the bank, using whatever methods the bank determines as being necessary. This might include logging in with normal internet banking credentials, a distinct set of Open Banking credentials, multi-factor authentication, or even the use of a dedicated application supporting custom features like biometrics or hardware tokens. The key thing here is that the connection at this point is between the end user and the bank. The third party is not involved directly, which is intended to increase the end user security since they never directly provide details to the third party. The end user is again shown the permissions that they are granting, and can choose to reject the request. It is not possible to reject parts of the request, however – either all permissions must be accepted, or the whole request must be rejected.

Assuming the end user accepts the permissions, they can then select which specific account or accounts they want to allow the third party to access. Having done this, the bank then redirects the end user back to the third party site or application. The third party then exchanges the AccountRequestId token for an access token, by sending another request to

the bank.

Now the third party can make requests directly to the bank using the access token. The first request the third party makes should retrieve the specific accounts that the end user selected on the bank system. Subsequent requests to the bank from the third party will relate to a specific account, and include an AccountID returned from this initial request.

The access token remains valid for future requests, meaning that the third party is able to access updated details relating to the end user accounts, without requiring the end user to grant consent again. An end user can revoke consent from a given third party at any point either from the bank interface (where the third party may not be informed of the revocation until they next make a request to the bank), or from the third party (where the bank may not be informed, despite provision being made within the APIs to allow third parties to inform banks of revoked consent). An access token remains valid for up to 90 days, with end users required to reauthorise specific TPPs if they want to allow ongoing access.

There are also some regulations which limit the ability of a third party provider to access account details without the end user "actively requesting information". Specifically, by default, a third party can only access an end user's account details up to four times in a 24 hour period, unless the end user is actively requesting the information. What counts as "actively requesting information" is not strongly defined, however, and the mechanism by which this is indicated to the bank (passing the end user IP address) is optional for third parties. Banks may also choose to limit the validity period for access tokens, requiring a third party to reobtain consent from the end user for continued access.

The process for a payment request is similar, although the current specifications require a new consent step for each payment, rather than a lasting permission. This is likely to change as recurring payment features are added.

What could go wrong and what prevents it?

The three-way flow is intended to minimise the risks to all three parties. The end user never provides their account details directly to the third party, instead only interacting with their bank, with whom they have a pre-existing trust relationship. The third party provider does not have to rely on the end user supplying financial details, but can instead rely on the bank to provide accurate information; this may be a particular concern for suppliers of loans or credit agreements. The bank can limit the information provided to third parties, and verify with their customer that they are happy for the information to be shared.

Nevertheless, it is still possible for steps to go wrong. The end user must trust the third party to request the correct permissions, and to redirect them correctly to the valid bank endpoint. Should a third party application be compromised, it may be possible for a malicious actor to direct end users to a fake bank endpoint, resulting in the loss of credentials. In theory, the bank must also trust that the redirection target supplied by the third party is valid. However,

according to the specification this must match the original redirection URL as provided when the third party registered for access to the bank APIs, which should prevent this from being practical.

A third party can claim that a customer is present by supplying an IP address in an “x-fapi-customer-ip-address” header, although there is no way for the bank to verify this for requests beyond the initial consent process. This may allow third parties to make additional requests, with little oversight from the end user. Detecting this would be difficult, although the potential reputational damage would act as a deterrent for businesses that rely on having FCA approval.

Once the access has been granted, the account details are available to the third party and can therefore be stored in systems under their control. At this point, the Open Banking element has no further impact, but the data could still be accessible in the event of problems with the TPP application, the infrastructure on which this sits, or while in onward transit from the TPP to other systems. There are also form-monitoring libraries designed to help with UI problems which will store all data submitted in forms on sites which are using them. If a TPP misconfigured one of these libraries, the data might not stay where it should.

Section 4 – Technical Controls

The Open Banking ecosystem builds upon the pre-existing OAuth2 framework for Financial APIs created by the OpenID Foundation. Specifically, some elements of the original framework are extended or established as firm requirements, while others are reduced due to other technical provisions making them less relevant.

For example, Open Banking specifications require that authorisation servers only support communication from confidential clients which have specific requirements regarding authentication processes. However, the requirements for a confidential client in Open Banking specifications allow more methods to be supported than the original OpenID specifications did.

Additionally, the Open Banking ecosystem includes a directory of third party providers. This ensures that only entities who are authorised by the current competent authority are able to register with banks as providers. A number of businesses registered on the initial launch date, mostly wanting to offer business account functions, and the process to allow further registrations is now in place.

For a third party provider to be able to act as an AISP or PISP for a given bank, they must first register for access to the bank’s API implementation. As part of this per-bank registration process, the bank checks with the Open Banking Directory for the status of the third party.

The directory can indicate whether a third party is registered with the competent authority and whether they have been suspended or removed.

Third parties must register with each bank from which they wish to access account details, and banks cannot refuse access to third parties who are appropriately registered with the FCA. The third party must provide a software statement, which defines what permissions they want to be able to access, and a set of callback URLs, where the bank should redirect users who want to authorise the TPP, following the authorisation process. These details can only be changed by specific named contacts from the TPP.

Once registered with the bank, the TPP can allow end users to trigger authorisation requests, using the flow mentioned previously.

While it would be theoretically possible for a TPP to make automated or fake authorisation requests, in order for these to be successful the TPP would have to correctly guess the password for the end user and bypass any second factor required by the bank.

Section 5 – Alternatives

The PSD2 directives that Open Banking APIs implement do not require that these specific APIs be used. In fact, they just make it so that banks must allow users to be able to provide their account information to a third party. This is the reason for changes to terms and conditions for bank accounts over the last year or so, where wording such as “you must not provide your online banking details to anyone else” has been changed to “you may use thirdparty providers who offer account information services or payment initiation services as long as they are authorised by law”.

This means that providers are free to ignore the Open Banking APIs, and continue to use screen scraping (at least until a ban on it comes into force in September 2019[15]) or custom-built parsing applications. There are also companies which effectively provide an alternative to the whole Open Banking ecosystem, such as Teller.io[16], who offer a reverse-engineered API to (at time of writing) 11 banks, including some who are not part of the CMA9. Teller have to keep updating their interfaces to preserve access to the account information whenever the mobile banking applications (or, specifically, the back-end calls they make) are modified, and are not FCA registered, so may not be allowed under the terms of some banks. Nevertheless, the Teller API provides similar levels of access to the official Open Banking APIs, and in some

[15] https://www.out-law.com/en/articles/2017/november/psd2-screen-scraping-ban-confirmed-in-finalised-standards/
[16] https://teller.io/

cases allows more access than is available from Open Banking. Teller also allows individuals to gain access to their account information.

The legality of reverse engineering in this way is subject to dispute. The banks include statements prohibiting reverse engineering in their terms and conditions of use for mobile applications, but the developers of Teller point to an EU directive which allows for adaptation of code for interoperability purposes. As a result, there is an uneasy relationship between some banks and the service, although others have agreed on official methods of access. There are also questions about who would be liable should problems arise: Teller, the application developer, the bank, or the end user.

Using a service like this may not be covered by as vigorous fraud protection as the official Open Banking standards. For example, some banks specify that end users are responsible for checking that companies they grant account access to are regulated. This is simple with the official Open Banking APIs: if the company can use them, they are regulated. For other methods, it becomes a matter of checking on the Financial Services Register[17] and ensuring that the company is listed with a suitable status. If there were a problem with an alternative like this, the banks would be able to reject claims for refunds more easily, although it seems unlikely that this was the intention of the regulations.

Another alternative API is provided by TrueLayer[18]. As with Teller they provide a unified API for multiple banks, which utilises a mix of scraping, bank specific access and, as soon as the APIs are fully available, the Open Banking APIs. Unlike Teller, TrueLayer are authorised with the FCA to provide both payment initiation and account information services, so would benefit from the full fraud protection required under the PSD2 directives, even if using legacy access methods. They also offer a different set of ASPSPs to the CMA9, including some providers who only offer credit cards. Teller and TrueLayer are not the only providers of unifying APIs, but both are concentrating on the UK market initially rather than aiming to gain global coverage as a first step.

Any service that stores sensitive credentials such as bank details is potentially a high value target for attackers. Therefore, it is important that security has been built into this type of system at a core level. Even if financial losses were covered, the requirement that original credentials can be accessed means that state-of-the-art password hashing methods cannot be used for the bank access passwords. Despite public awareness campaigns many users are still using the same passwords for multiple accounts, which could even potentially expose accounts that are not entered into these services directly to compromise.

Additionally, unless the bank has implemented read-only alternative credentials, or other direct access methods, any service that receives online banking credentials could technically

[17] https://register.fca.org.uk/
[18] https://truelayer.com

perform any action that the account holder could, subject to any requirements for 2FA. Typically, sending money to a new recipient, configuring standing orders, and sending large amounts of money are considered by banks as risky operations that require a second factor to be used, but more subtle actions could equally be used to cause problems for account holders. These could include making additional payments to previously configured recipients, or transferring money between different types of account held with a specific bank (sending money from a current account to an ISA, for example, could cause problems with payment limits on ISAs or make it difficult to return the money to a current account). In these cases, it may be difficult to prove that the actions were unauthorised.

Some of the smaller banks operating in the UK have also provided their own API interfaces. Starling Bank[19] and Monzo[20] each provide APIs which give access to similar details as the Open Banking APIs, but which are not currently part of the Open Banking Directory. While these are of limited use to companies wanting to maximise customer accessibility, both of these APIs can be accessed by individuals wanting to monitor their own account activity. In some ways, these bank-specific APIs are closer to what some technical users were expecting from Open Banking, allowing for individuals who want to implement their own scripts for monitoring accounts.

Conclusion

Open Banking is a complex ecosystem of standards and software built by a mixture of banking giants, regulated third parties, and standards boards reacting to government decrees. As a result, at times it appears to be haphazard and incomprehensible, with companies being forced to rush through implementation or face censure by the FCA.

The key elements, however, are less haphazard, although the specific benefits or aims of these have not always been communicated clearly to end users, creating a climate of uncertainty. Despite what newspaper reports may suggest, the Open Banking system is not designed to expose account details to any business that takes payments from individuals, and there is no need to opt out of using it. Only specific types of businesses can access the data, and even then, only with the consent of the account holder.

It is not all plain sailing however. The launch delays and limited functionality in the initial release may make selling the benefits difficult, with smaller players taking the first steps towards using the APIs. Early predictions of large entities such as Google or Amazon moving into the banking advice sector have not yet come true, although they both have histories of

[19] https://developer.starlingbank.com/docs
[20] https://monzo.com/docs/

acquiring companies that have products in markets in which they want to operate. Take up of the Open Banking APIs by banks other than the CMA9 is also an unknown.

In security terms, the use of APIs that do not require that online banking details are shared with third parties and which limit third party access to specific actions has to be a good thing. The three-way process, where end users authorise on the bank infrastructure, should help reinforce the security aspects, especially if banks integrate multi-factor authentication into the process.

There is clearly a demand for a robust, unified API platform allowing secure access to bank account details, as the number of companies looking at entering this market shows. However, even among these companies, there is a very mixed view as to whether the Open Banking APIs should be welcomed as a secure method of access, or derided as “too little, too late”.

It is also unclear whether the original stated objective of Open Banking will be met: many of the challenger banks have their own APIs which provide similar or greater levels of access as the Open Banking ones. For early adopters these may prove more attractive, giving access to their own accounts for custom functions, while for the 30% of the British population who have resisted online banking even the services offered by their own banks are likely to be ignored.

About NCC Group

NCC Group is a global expert in cyber security and risk mitigation, working with businesses to protect their brand, value and reputation against the ever-evolving threat landscape.

With our knowledge, experience and global footprint, we are best placed to help businesses identify, assess, mitigate & respond to the risks they face.

We are passionate about making the Internet safer and revolutionising the way in which organisations think about cyber security.

Headquartered in Manchester, UK, with over 35 offices across the world, NCC Group employs more than 2,000 people and is a trusted advisor to 15,000 clients worldwide.