OneSpan
Be bold. Be secure.

OPEN BANKING APIS UNDER PSD2 SECURITY THREATS AND SOLUTIONS

WHITE PAPER

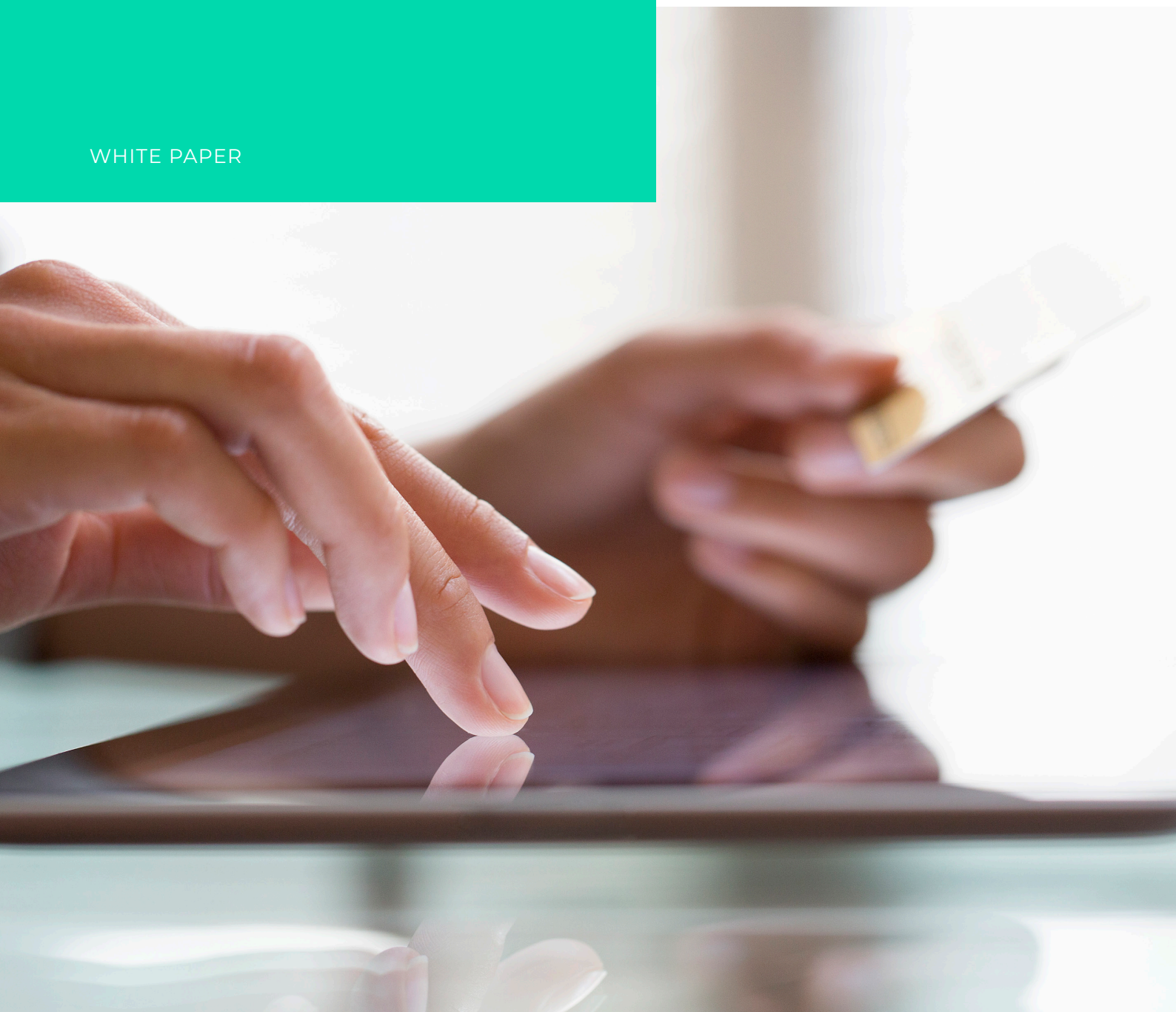


TABLE OF CONTENTS

About the Author

Frederik Mennes is Senior Manager Market & Security Strategy at OneSpan. He regularly advises financial institutions about the security and compliance aspects of online and mobile banking applications.

You can reach him at frederik.mennes@OneSpan.com.

INTRODUCTION

The Revised Payment Services Directive, also known as PSD2, requires European banks to provide communication interfaces to Third Party Providers (TPPs). These interfaces, generally referred to as APIs, will allow TPPs to build innovative financial services on top of the services of the banks. The requirements for these interfaces are defined in the Regulatory Technical Standards (RTS) on Strong Customer Authentication (SCA) and Common and Secure Communication (CSC), of which the European Banking Authority (EBA) published a draft version in February 2017.

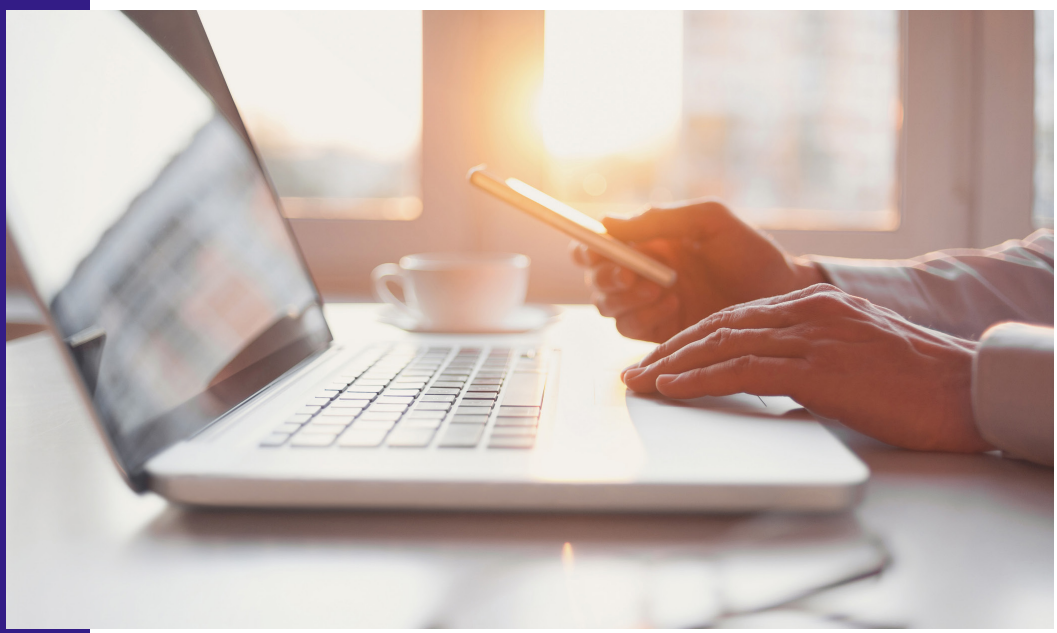
In this whitepaper we analyze the requirements for the communication interface as defined in the draft RTS, with a special emphasis on security requirements. Subsequently we identify the most important security threats against these interfaces. Finally we discuss various solutions that can help banks to protect their interfaces against these security threats.

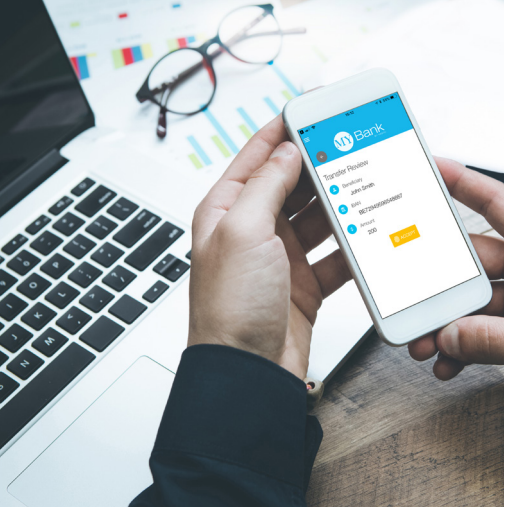
Background

In recent years, Open Banking has received a lot of attention in the financial services sector. Open Banking means that banks provide communication interfaces to Third Party Providers (TPPs), so that these companies can have access to data about the bank's customers, initiate financial transactions, and consult the transaction history of the customer. As such it promises to unlock innovation that will profoundly transform and improve the banking experience of consumers, and that will bring new financial services to consumers. For instance, TPPs will provide applications allowing consumers to consult multiple bank accounts from a single application and look for a mortgage more easily, allowing banks to find customers matched to a new product, and allowing businesses to share data more easily with their accountants.

In many countries and regions around the world, banks can decide for themselves whether they want to participate in the Open Banking paradigm. However in the European Union the Revised Payment Service Directive, also known as PSD2, requires banks to provide open communication interfaces. In this regard Article 98 of PSD2 tasks the European Banking Authority with the development of detailed requirements for the communication interface that banks will need to offer to TPPs. In line with this mandate, the EBA submitted its final draft RTS on February 23rd 2017 to the Commission for adoption, and they are now subject to scrutiny by the European Parliament and the Council, before being published in the Official Journal of the European Union. The RTS is expected to be published in September or October 2017 and will be applicable 18 months after its entry into force, which suggests an application date of the RTS in March or April 2019 at the earliest.

While similar regulations do not yet exist in the United States, there is growing pressure from FinTech lobbying groups and the Consumer Financial Protection Bureau to make financial data more accessible. In Asia, the Monetary Authority of Singapore (MAS) supports APIs as part of its Smart Financial Centre initiative. Finally the Economics Committee of Australia and the Financial Services Commission in South Korea also promote the idea of APIs.





PSD2 requirements for the APIs of banks

Section 2 of the draft RTS specifies that banks must offer an interface allowing TPPs to communicate with banks, and provides a number of high-level requirements for this interface. In this section we discuss the most important requirements.

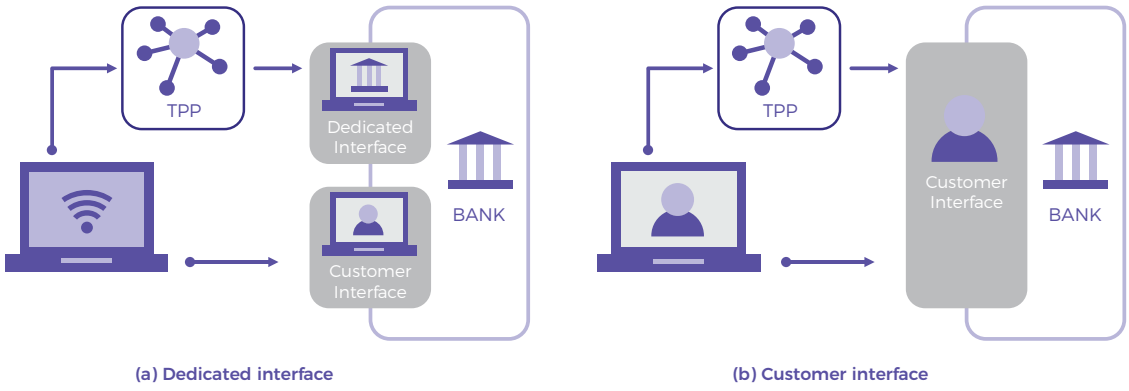
Type of communication interface: dedicated vs. customer-based

Article 27(1) requires banks to provide an online communication interface that allows TPPs to identify themselves to the bank, to request and receive information about bank accounts and financial transactions of customers of the bank, and to initiate a financial transaction from an account of a customer of the bank. The TPP needs consent from the user to perform the latter two functions.

According to Article 27(2) banks can offer this interface in following two ways, as illustrated in Figure 1:

a) Via a dedicated interface. In this case the bank develops a new interface specifically to support TPPs. Messages exchanged via this interface should be structured according to the ISO 20022 standard, which is already heavily used for electronic data interchange between financial institutions. Although not mandatory, the dedicated interface is generally expected to be based on RESTful APIs or similar technology. Many banks are expected to opt for this approach based on dedicated interfaces.

b) Via the customer interface. In this case the bank reuses an interface that it already offers to its users. For instance, the bank could reuse its existing web or mobile banking applications in this context. An important difference is that the TPP rather than the user needs to identify himself to the bank.



(a) Dedicated interface

(b) Customer interface

Figure 1: Options for the communication interface which banks have to offer to TPPs. The user can communicate directly with the bank, or via a TPP. Banks expose either dedicated interfaces (a) or customer interfaces (b) to TPPs.

Banks must provide TPPs with a test version of their communication interface, so that they can test the connectivity and functionality of the interface, as specified in Article 27(6).

Integration of authentication into communication interface

The RTS requires banks and TPPs to perform Strong Customer Authentication (SCA) when a user accesses a payment account, initiates a financial transaction, or performs any other action via a remote channel that might imply payment fraud. An important topic of the RTS is therefore how the Strong Customer Authentication mechanism can be integrated into the communication interface of the bank.

According to Article 27(3), the online communication interface of the bank – whether it's a dedicated or existing customer interface – must allow TPPs to rely on the mechanisms used by banks to authenticate its customers. For instance, if the bank offers a hardware token, card reader or mobile authentication app to its users, then TPPs must have the option to use these mechanisms to authenticate their users as well.

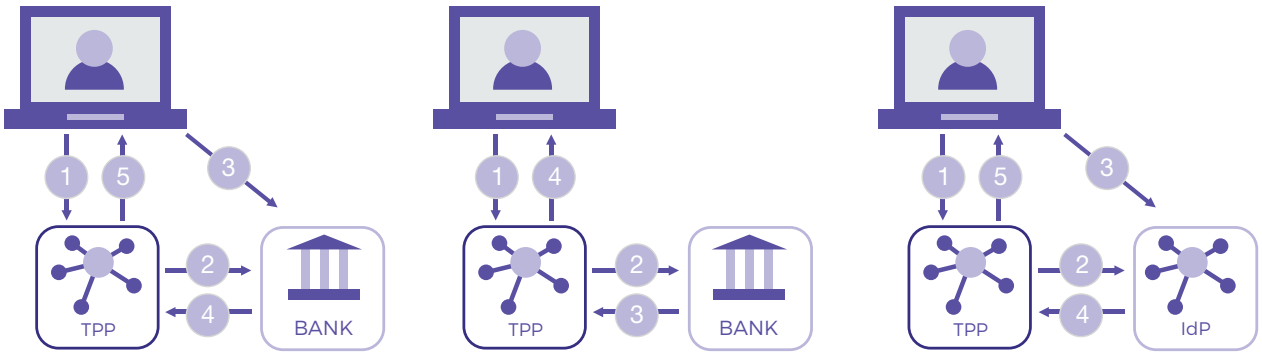
Article 27(3c) furthermore states that it must be possible to submit personal security credentials and authentication codes to the bank via the TPP. In other words, the user might enter his bank-provided credentials into a webpage or mobile app provided by the TPP, and the TPP would then forward these to the bank for verification.

These requirements give rise to a number of possible approaches for the flow between a bank and TPP, which are currently being developed in more detail by several European standards organizations, such as the Euro Retail Payments Board (ERPB), the NextGenPSD2 task force of the Berlin Group, and Convenient Access to PSD2 Services (CAPS). More specifically the following models are being considered:

a) **Redirection:** the user is redirected from the TPP's application to the bank's application for the sole purpose of authentication, and is then redirected back to the TPP's application. The user enters his authentication credentials (e.g. user ID, one-time password) into the bank's authentication application. This model is used, for instance, by the OAuth authorization protocol and 3-D Secure protocol for card payments.

b) Embedded: the user enters his authentication credentials into the TPP's application, and the TPP passes them on to the bank for verification. This approach is the one allowed under Article 27(3c) described above.

c) Decoupled: this is similar to the approach based on redirection, but now the user is directed to a third party – also called Identity Provider (IdP) – of authentication services instead of a bank.



(a) Redirection (b) Embedded (c) Decoupled

Figure 2: Possible models for integration of authentication into the flow between user, TPP and bank or identity provider (IdP)

Security requirements for communication interface

In line with best security practices, one could expect that the RTS requires banks and TPPs to communicate over a secure channel, providing mutual authentication of both the bank and TPP towards each other, and protecting the confidentiality and integrity of data exchanged over this channel. Additionally, the bank should verify whether a certain TPP is indeed authorized to perform a certain function. For instance, Account Information Service Providers (AISPs) may have the right to obtain information from a bank, but should not have the right to initiate a payment.

In practice the draft RTS contains the following high-level security requirements for the communication between a bank and TPP:

1) Mutual authentication. The draft RTS contains very limited provisions about the authentication of the bank and TPP towards each other. As mentioned above, Article 27(1) requires TPPs to identify themselves to banks. More specifically TPPs need to identify themselves towards banks using certificates issued by a Qualified Trust Service Provider (QTSP) in accordance with the European eIDAS regulation. This is nonetheless different from TPPs authenticating themselves, as authentication provides stronger security guarantees than mere identification. Furthermore the draft RTS does not explicitly require banks to identify or authenticate themselves to TPPs.

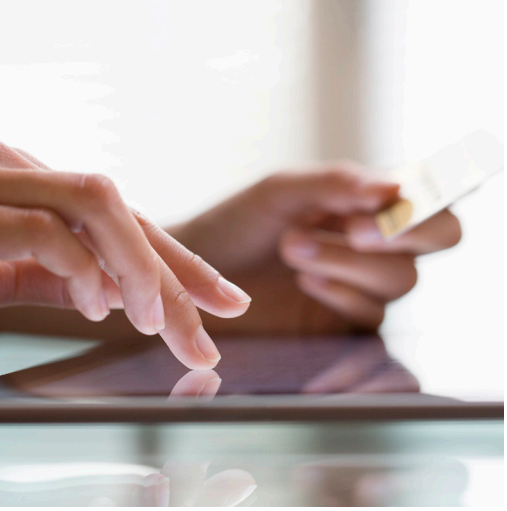
2) Confidentiality and integrity. Article 27(3c) as well as Article 30 require banks and TPPs to protect the confidentiality and integrity of the data they exchange using encryption.

3) Authorization. The basic principles of authorization are laid out in Article 27, which says that AISPs (but no other types of TPPs) should be able to request and receive information about bank accounts and financial transactions. Similarly, Payment Initiation Service Providers (PISPs) – but no other types of TPPs – should be able to initiate a financial transaction from a user's bank account and receive information from the bank on the initiation and execution of financial transaction. PSD2 requires the European Banking Authority to maintain a central register listing all TPPs and the rights they would have, which is currently under development.

4) Other security requirements. Article 30 also requires TPPs to terminate the session with the bank's communication interface as soon as the requested action is completed. Finally, the communication sessions between the user and TPP on one hand, and between the TPP and bank on the other hand, need to be linked to each other.

As such many details of the security requirements are not present in the draft RTS and are left to the European organizations developing API standards. Important open questions include:

- Are banks required to authenticate or merely identify TPPs?
- Which security checks need to be performed to validate a certificate issued by a QTSP?
- Are TPPs required to authenticate banks, and if so, how?
- Is a PISP authorized to request and receive bank account information?



Security and privacy threats against the APIs of banks

In this section we discuss the most important security and privacy threats that banks need to consider when providing communication interfaces to TPPs. We assume that banks opt for the dedicated interface, based on APIs.

Overview

The table below provides a summary of the most important security and privacy threats against the APIs provided by banks to TPPs. We discuss these threats in more detail in the next sections.

Table 1: Overview of security and privacy threats against APIs of banks

RISK	EXAMPLES OF THREATS	POTENTIAL BUSINESS IMPACT
Leakage of financial information of users	· API vulnerabilities, resulting in injection attack causing dump of personal information of bank's users · Compromised or malicious TPP leaking financial information obtained from bank	· Legal liability (e.g. GDPR fines) · Reputational damage
Fraudulent financial transactions via API	· API vulnerability leading to man-in-the-middle attack manipulating transaction data · Compromised or malicious TPP issuing fraudulent transaction request	· Financial loss · Reputational damage
Unavailability of API	· Flooding of API affecting quality of service for users · Compromised or malicious TPP locking out users with invalid authentication requests	· Contractual liability · Reputational damage · Negative impact on users

Security threats originating from the authentication model

The different authentication models discussed above have different security characteristics, which banks should take into account when deciding how to provide their APIs:

1) The embedded authentication model requires the user to enter his authentication credentials into a webpage or mobile app provided by the TPP rather than the bank, which might give rise to phishing or man-in-the-middle attacks. Additionally it breaks the end-to-end encryption for the communication channel between the user and bank. Finally, since banks are liable for payment fraud, they may prefer to perform authentication of the user so that they can collect irrefutable proof of authentication events.

2) The redirection and decoupled approaches ensure that authentication credentials are only entered into webpages or mobile apps provided by the bank (or IdP), which ensures the bank directly obtains evidence of authentication events. However, models based on redirection need to be properly designed to avoid security flaws. For instance, the 3-D Secure protocol for cardholder authentication performs redirection using an iframe or pop-up window without an address bar, which means there is no easy way for a customer to verify who is asking his credentials [2]. This might subject the user to phishing attacks as well.

Security and privacy threats originating from incidents at TPPs

The introduction of APIs by banks means that their security and privacy becomes more dependable on the security of TPPs using these APIs. Examples of threats resulting from this dependency are:

1) Leakage of personal data about bank's users from compromised TPP. TPPs will store financial data from the users of banks, hence security incidents at TPPs might impact the confidentiality and integrity of data about a bank's users, which in turn might impact the reputation of the bank. In light of the upcoming European General Data Protection Regulation (GDPR), banks need to make sure TPPs implement appropriate technical and organizational measures to protect the data they share with TPPs.

2) Fraudulent requests from compromised TPP. A security incident at a TPP might result in fraudulent requests for information about a bank's users, or fraudulent payment initiation requests from the TPP to the bank. An incident might be caused, for instance, by vulnerabilities in the TPP's mobile app or web application. Since banks are the first party to be held liable for unauthorized financial transactions from a user's bank account, the security breach at the TPP may also have consequences for the bank.

3) Fraudulent requests from malicious TPPs. Similarly, a disgruntled employee at a TPP might issue fraudulent requests for information about a bank's users, or initiate fraudulent payments.

4) Account lock-out by compromised or malicious TPP. A security incident at a TPP might cause the TPP to send invalid authentication requests to accounts of users of the bank. As a consequence the bank needs to temporarily or permanently block the bank accounts of these users, in line with Article 4(3b) of the draft RTS. This would cause loss of service to users of the banks.

Security and privacy threats originating from API implementation vulnerabilities

Modern banking applications are increasingly composed of rich clients (browser, mobile, or even desktop) that connect to back-end RESTful APIs (e.g. micro-services, web services) over HTTP using JSON or XML to structure data. If not implemented properly, APIs can be vulnerable to many security threats. These threats might result in theft, corruption or destruction of sensitive user data, leakage of sensitive data, unauthorized financial transactions, or even complete take-over of the banking application.

The most important security threats against APIs are the following:

1) Injection attacks can occur when untrusted data is sent to an interpreter as part of an API function call. The attacker's hostile data can trick the interpreter into executing unintended commands. Injection flaws are usually the result of developers not properly sanitizing input to the APIs. An example of such an attack is SQL injection, whereby an adversary places malicious SQL statements into an entry field for execution. For example, the adversary could instruct the database to dump certain records. In this way the adversary could obtain sensitive financial or personal information from the database. SQL injections typically exploit incorrect filtering for string literal escape characters embedded in SQL statements. Another example are HTTP Parameter Pollution (HPP) attacks, whereby an adversary crafts a HTTP request in order to manipulate or retrieve hidden information. The HTTP request is crafted so that the attack vector is split across multiple parameters of the HTTP request (e.g. the URI, the request body, the request header) and recombined by the HTTP server upon receipt of the API function call, resulting in the execution of the attack vector. In this way the adversary could initiate a fraudulent transaction, whereby money is transferred to a bank account controlled by the adversary.

2) Authentication and session attacks. Application functions related to authentication and session management are often not implemented correctly, allowing adversaries to compromise passwords, cryptographic keys or session tokens, and assume other users' identities or obtain excessive privileges. For instance, an adversary could hijack the session of a user if a session token is exposed in a URL and the adversary obtains the hyperlink. As another example, a bank might not properly validate the X.509 certificate of a TPP using its API, resulting in a rogue TPP connecting to the bank's API. As a result of this, the rogue TPP could obtain financial information about users of the bank or even initiate fraudulent financial transactions.

3) Man-in-the-middle attacks allow adversaries to tamper with legitimate API requests/responses. These attacks usually occur because requests and responses are not exchanged via a secure channel (e.g. SSL/TLS), or because the secure channel is not properly authenticated. For instance, SSL/TLS might be implemented without certificates allowing the bank or TPP to be spoofed. SSL/TLS might also be implemented without HTTP Strict Transport Security (HSTS), allowing a redirection attack from HTTPS to plaintext HTTP.

4) Denial-of-Service (DoS) attacks can affect the availability of the API. APIs are potentially open to flooding and other types of DoS attacks that can bring back-end systems to a halt. A DoS attack cripples an API by overwhelming it with requests.

Protecting APIs against security threats

Banks should adopt a number of technical and organizational security measures to address security threats against their APIs. This section provides recommendations for these security measures.

Use transaction risk analysis (TRA) to detect fraudulent transactions and incidents at TPPs

In the context of Open Banking, banks can use transaction risk analysis in in multiple ways:

1) Detect fraudulent transactions and user behavior. The draft RTS mandates PSPs, including banks, to perform risk analysis of all financial transactions that they process. In this way PSPs can detect payment fraud, such as Card-Not-Present (CNP) fraud.

2) Detect security incidents at TPPs. Transaction risk analysis can be used to detect abnormal behavior in requests originating from TPPs, identify suspicious transactions from TPPs, detect atypical sequences of API calls, and all of this in real time.

3) Detect API implementation vulnerabilities. Weaknesses in the implementation of APIs might give rise to fraudulent transactions and unusual user behavior, which can be detected using advanced fraud monitoring.

Transaction risk analysis requires a comprehen-sive and intelligent platform that accurately detects and prevents fraud. Through sophisti-cated, real-time risk analysis, OneSpan Risk Analytics delivers strong and dynamic protection against fraudulent activities across multiple channels, including web, mobile, API and others. OneSpan Risk Analytics works silently in the background to collect and score activities and transactions based on intelligent analysis of behavioral, contextual, qualitative and quantitative data, and by challenging unusual patterns.

Choose the appropriate authentication model

As discussed above different authentication models have different security characteristics. From the perspective of security for banks, the redirection and decoupled models are the most attractive, since they allow banks to directly interact with the user during the authentication process. This, in turn, allows banks to collect following useful information:

1) Proof of authentication events. This is relevant to handle disputes about the liability of a fraudulent transaction.

2) Information about the security status of the authentication devices the users of TPPs. This information can be taken into account by the transaction risk analysis engine of the bank.

The OneSpan Mobile Security Suite is a comprehensive software development kit (SDK) for mobile apps that allows collecting security-critical information about the user's mobile device, such as the device's geolocation and jailbreak/root status. By collecting this information from the user's device and feeding it into OneSpan Risk Analytics, banks can ensure visibility into the security-level of the user's devices.

Protect the communication channel with TPPs

Banks should ensure they protect the communi-cation channel between their APIs and the TPPs using mutual authentication and secure encryption in order to minimize the risk of man-in-the-middle attacks, leakage of sensitive user data, and unauthorized financial transactions. Security protocols like SSL/TLS can be used for this purpose. However, SSL/TLS is still subject to misuse. It is easy to misconfigure at the server-side and many develo-pers do not properly validate certificates when using it. Hence care must be taken when deploying this.

Implementing a secure communication channel without the burden of certificate management, can be accomplished with OneSpan's DIGIPASS for Apps.

This software development kit (SDK) natively includes a Secure Communications functionality that allows TTPs to create an end-to-end encrypted channel between the app and the bank's APIs.

Request independent security audit reports from TPPs

The European Commission has anticipated security risks under PSD2 and therefore requires PSPs, including TPPs, to adequately manage operational and security risks. More specifically, Article 95(1) of PSD2 requires PSPs to establish a framework with appropriate mitigation measures to manage operational and security risks relating to the financial services they provide.

In relation to this, Article 95(3) requires the EBA to issue draft Guidelines with regard to the establishment, implementation and monitoring of the security measures. These Guidelines, which are expected to be published by the EBA in Q3 2017, cover the following:

- Governance, including the operational and security risk management framework, the risk management and control models, and outsourcing
- Risk assessment, including the identification, classification and risk assessment of functions, processes and assets
- Protection of the integrity of data, systems and confidentiality, physical security and asset control
- Monitoring, detection and reporting of security incidents
- Business continuity management, including testing of business continuity plans, incident management and crisis communication
- Independent testing of security measures.

Banks should request TPPs to provide independent security testing reports so that they can verify the maturity of their security practices.

Avoid security vulnerabilities in the API implementation

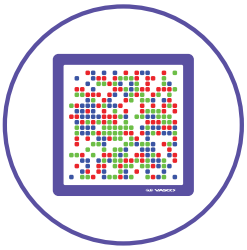
In order to protect against injection attacks, ensure that whatever data format the API uses, that the API software component parsing the data structures is hardened against attack. Also protect input sanitization to prevent injection of all forms. Be sure your security analysis and testing covers all your APIs and your tools can discover and analyze them all effectively.

Finally, a list of the Top 10 security threats against APIs is currently under development by the Open Web Application Security Project (OWASP) [3]. This list and the security best practices that come with it are a great source of knowledge for the bank's developers.

OneSpan's Solution Suite for PSD2 Compliance

OneSpan's Multi-factor Authentication Solution – Enables quick compliance through strong, customizable and easy-to-deploy, authentication options

- Complete multi-factor authentication solutions, offering fast, convenient & secure authentication for all users
- Multi-modal biometrics framework with next generation behavioural and contextual authentication options, driving stronger security and the best user experience
- Extensive hardware and software solutions to ensure full compliance across nearly every application



OneSpan's Secure Channel Technology – Ensures confidentiality, integrity and authenticity of every payment transaction

- Secure transaction data signing technology links the authentication code to the amount of the transaction,
  payee information and other key details
- Unique end-to-end protection of communication between user and the Payment Service Provider
- Unique visual validation feature provides users a quick and convenient check of each transaction
- Omni-channel user experience leveraging innovative transaction signing technology or secure push notifications

OneSpan's Mobile Application Security – Mitigates malicious attacks on mobile apps and reduces exposure to related fraud

- Provides extensive app protection features far beyond what's available through the OS or typical mobile app development security practices
- Wraps around the mobile app, enabling it to operate safely even on infected devices
- Proactively manages the real threat of sophisticated malware, preventing foreign code from altering the app, detecting real-time attacks and reporting malicious attacks to fraud management platforms

OneSpan's Secure Provisioning Tools – Dramatically reduces the risks of unauthorized use on authentication platforms

- Strong security technology for deployment, provisioning and activation of authentication solutions, preventing credential theft attacks
- Flexible and streamlined provisioning process across on- and-offline channels including: web, app, branch, IVR and ATM

OneSpan's Fraud Prevention Solution – Enables compliance with strict regulations, satisfying transaction monitoring and risk analysis requirements

- Real-time detection of sophisticated fraud drives down exposure and boosts the top line
- Processes 200+ data points to detect abnormal user behavior, suspicious transactions, and atypical navigation in the payment application
- Accurate risk scoring and threat mitigation across all common devices (e.g. mobile phones and tablets)
- Operates invisible to end users, mitigating fraud while providing the best possible user experience

Glossary

ACRONYM	TERM	MEANING
AISP	Account Information Service Provider	A type of TPP offering a service based on financial information collected from banks
API	Application Programming Interface	A set of clearly defined methods for communication between various software components
ASPSP	Account Servicing Payment Service Provider	A PSP such as a bank or card issuer that provides authorized access to bank account information
EBA	European Banking Authority	A European agency which works to ensure effective and consistent prudential regulation and supervision across the European banking sector
ERPB	Euro Retail Payments Board	The purpose of the ERPB, launched by the European Central Bank (ECB) is to contribute to and to facilitate the further development of an integrated, innovative and competitive market for euro retail payments in the EU.
PISP	Payment Initiation Service Provider	A type of TPP offering a service that allows initiation of payments without the customer needing to directly access their bank account or use a debit or credit card.
PSD2	Revised Payment Services Directive	A European directive that primarily aims to deliver open banking and reduce payments fraud.
PSP	Payment Services Provider	In the context of PSD2, this refers to entities that provide payment services through issuing credit or debit cards or offering payment mechanisms through accounts. Examples are ASPSPs and TPPs.
PSU	Payment Services User	A natural or legal (company) person making use of a payment service in the capacity of payer, payee, or both.
QTSP	Qualified Trust Service Provider	An entity allowed to issue qualified digital certificates which can be used to create qualified electronic signatures.
REST	Representational State Transfer	A set of architectural principles for designing web services
RTS	Regulatory Technical Standards	Regulatory Technical Standards define certain requirements of PSD2 in more detail. The European Banking Authority is responsible for the development of the RTS to meet the objectives of PSD2 as defined by the European Commission.
SCA	Strong Customer Authentication	A set of principles to verify the identity of a user of a banking or payment application, based on the usage of authentication codes generated by a two-factor authentication mechanism.
TPP	Third Party Provider	A company offering payment services in the context of PSD2. AISPs and PISPs are examples of TPPs for PSD2.

For more information on PSD2 compliance,

visit www.OneSpan.com/psd2

[1] European Banking Authority, Final Report on Draft Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2), https://www.eba.europa.eu/regulation-and-policy/payment-services-andelectronic-money/regulatory-technical-standards-on-strong-customer-authentication-andsecure-communication-under-psd2

[2] Steven J. Murdoch and Ross Anderson, Verified by Visa and MasterCard SecureCode: Or, How Not to Design Authentication, http://sec.cs.ucl.ac.uk/users/smurdoch/papers/fc10vbvsecurecode.pdf

[*] OWASP API Security Project, https://www.owasp.org/index.php/OWASP_API_Security_Project

OneSpan

OneSpan enables financial institutions and other organizations to succeed by making bold advances in their digital transformation. We do this by establishing trust in people's identities, the devices they use, and the transactions that shape their lives. We believe that this is the foundation of enhanced business enablement and growth. More than 10,000 customers, including over half of the top 100 global banks, rely on OneSpan solutions to protect their most important relationships and business processes. From digital onboarding to fraud mitigation to workflow management, OneSpan's unified, open platform reduces costs, accelerates customer acquisition, and increases customer satisfaction.

CONTACT US

For more information:
info@OneSpan.com
www.OneSpan.com