protiviti®
Global Business Consulting

Open Banking Data and Information Security Specification

Central Bank of Oman

21 12 2023

Table of Contents

1.1 Background and purpose 3

1.2 Objectives 3

1.3 Principles of Data and Information Security Specification 4

2 The Framework 5

3 Control Domains for Open Banking 5

3.1 System Security Guidelines 6

3.1.1 Authentication and Authorization 6

3.1.2 Application Security and Secure Development Practices 7

3.1.3 Security Hardening 8

3.1.4 Vulnerability Management and Security 8

3.1.5 Event Logging 9

3.1.6 Data and Security Breach Incident Management Reporting 9

3.1.7 Disaster Recovery 10

3.2 Data Security Guidelines 11

3.2.1 Data Privacy 11

3.2.2 Information Sharing 11

3.2.3 Data encryption 12

3.3 API Security Guidelines 12

3.3.1 Input Validation 12

3.3.2 Controls over Excessive Data Exposure 13

3.3.3 Resources and Rate Limiting 13

3.3.4 Controls against Injection Risks 14

3.3.5 API Lifecyle Management 14

4 Appendix 15

1 Introduction

1.1 Background and purpose

Open banking enables financial services organizations such as banks, fintechs etc. to allow their retail and corporate customers to share account data securely with third party providers through the usage of Application Programming Interfaces (APIs).

Data and Information Security is a core focus area in open banking domain and is essential to foster continued customer trust in the safety and resilience of the open banking ecosystem. Hence, this specification document is developed to support the participant organizations in developing and maturing robust controls to safeguard themselves, their customers and the broader ecosystem from relevant threats.

These specifications should be read in conjunction with the all the existing guidelines and mandates published by the Central Bank of Oman (CBO) on cybersecurity. All ecosystem participants must ensure compliance to all the relevant existing cybersecurity standards applicable in Oman. All participants are solely responsible for their security compliance with the relevant regulations applicable to their service offering.

1.2 Objectives

Effective technology risk management and robust cybersecurity are essential to the operation and resilience of the Open Banking ecosystem. The consequences of compromises are significant, such as:

- Customer harm, including fraud or data compromise
- Loss of customer trust and reduced uptake of open banking products and services
- Loss of revenue
- Reputation damage

Ensuring the confidentiality, integrity, and availability of information and data within the open banking ecosystem necessitates a heightened focus on cybersecurity across all participating organizations. This encompasses establishing a dedicated cybersecurity function, the formulation of well-defined cybersecurity strategy, policy, and governance frameworks, and the development of capabilities for continuous monitoring and reporting of threats.

Cybersecurity should be at the core of open banking products and services. Robust controls have to be put in place throughout product development, deployment and production to ensure risks are appropriately mitigated.

These good practice guidelines will be maintained in line with CBO's processes and will be updated regularly – following any material changes to industry standards and to the threat landscape.

In addition to the guidance provided below, it is recommended that all stakeholders in the open banking ecosystem consult relevant regulatory documentation and guidelines, industry standards, threat intelligence and online resources for the very latest security guidance and regulatory requirements.

The objectives of the Data and Information Specification are to:

- Stipulate safeguards for financial system stability under an open banking regime;
- Provide clear responsibilities and expectations related to cybersecurity for the various participants;
- Outline pragmatic cybersecurity requirements for the open banking participants in Oman.

1.3 Principles of Data and Information Security Specification

This data and information security specification document is developed to ensure a fit-for purpose control framework that adequately addresses risks the Open Banking operational context and business environment.

This specification has been developed by considering the following Security Principles:

- **Layered Security** is a principle that security controls are layered to reduce the security risk.
- **Zero Trust** is a principle that security design that assumes asset compromise to minimize uncertainty in enforcing accurate, least privilege per-request access decisions in information systems and services.
- **Least Privilege** is a principle that restricts the access privileges of authorized personnel (e.g., program execution privileges, file modification privileges) to the minimum necessary, to perform their jobs.
- **Need to Know** is a principle that ensures that only authorized recipients are provided access to specific classified information to perform or assist in a lawful and authorized function.

2 The Framework

The primary goal of the Data and Information Security guidelines is to establish a standardized and a clear set of actionable security controls applicable to all participants involved within Oman's Open Banking ecosystem. To achieve this objective, it is crucial to employ a Framework approach during the development of security controls. The Framework serves as a structured and organized methodology for establishing, implementing, monitoring, and improving security measures. It provides a comprehensive and systematic way to address potential risks and vulnerabilities within the Open Banking environment.

The Open Banking Cybersecurity Framework (CSF) has been developed considering the four core capabilities; **Protect**, **Detect**, **Respond** and **Recover**.

These four widely understood and recognized terms, when considered together, provide a comprehensive view of the life-cycle for managing Open Banking cybersecurity and each capability entails key cybersecurity controls.

Further there are three control domains comprising of System, Data and API Cybersecurity Controls. Each of the control domains have required control statements that are to be implemented by the respective stake holders and are detailed in the subsequent section of the document.

The CSF is designed to encompass fundamental capabilities, each playing a pivotal role in guiding the development of actionable security controls and enhancing the overall security posture:

- **Protect**: Develop and implement safeguards to protect the organization's assets, data, and systems from unauthorized access, use, disclosure, disruption, modification, or destruction.
- **Detect**: Develop and implement activities to identify and report cybersecurity events.
- **Respond**: Develop and implement plans and procedures to respond to cybersecurity events.
- **Recover**: Develop and implement plans and procedures to restore systems and data that have been affected by a cybersecurity event.

By incorporating these four core capabilities within the Framework, the guidelines for Data and Information Security Controls aim to establish a resilient and adaptable security infrastructure within the Open Banking ecosystem. This approach ensures that all stakeholders adhere to a common set of practices, fostering trust, and enhancing the overall security posture of the financial services provided in the Open Banking landscape.

3 Control Domains for Open Banking

For ease of reference, this Data and Information Security Specification is structured around the below three control domains, which provides a common language to promote a shared understanding of security risks and controls applicable to Open Banking:

- System Security Guidelines
- Data Security Guidelines
- API security Guidelines

3.1 System Security Guidelines

3.1.1 Authentication and Authorization

There must be a security control mechanism to authenticate the identity of the user/device and or an Application programming Interface (API). After authentication, a security control mechanism must also ensure that the access rights to the application and data must be limited to only his authorized access level. And subsequently this authorized session should be controlled and terminated.

Organizations are required to adopt the following practices:

3.1.1.1 Authentication is mandatory for all resources and services, except those specifically intended to be public.

3.1.1.2 Users/devices and API consumers should be continuously authenticated and authorized.

3.1.1.3 Use modern authentication and authorization protocols (Like OpenID for authentication and OAuth 2.0 for authorization. Further details and resources on the OpenID/OAuth 2.0 specifications can be found on their website).

3.1.1.4 Certificate-based (Mutual TLS) authentication is to be used for machine-to-machine/API communication as it is not technically feasible to prompt for authentication inputs. Mutual TLS is to be used along with JSON web tokens for user session authorizations.

3.1.1.5 TLS should be used for transport layer security, ideally version 1.3 but 1.2 at a minimum.

3.1.1.6 All versions of SSL should be disabled due to the number of weaknesses in the protocol or related cipher suites.

3.1.1.7 Use Multi-Factor Authentication for transactional accounts.

3.1.1.8 Implement account auditing and enforce the disabling of unused accounts (e.g., After no more than 30 days from the expiration of an account’s password).

3.1.1.9 Enforce password complexity and length requirements established by policy or regulation.

3.1.1.10 User entry of Password should be obscured on the user's screen. (e.g., on web forms use the input type "password").

3.1.1.11 Enforce account disabling after an established number of invalid login attempts.

3.1.1.12 Concurrent logins with the same user ID must be prohibited.

3.1.1.13 The last use (successful or unsuccessful) of a user account should be reported to the user at their next successful login.

3.1.1.14 Disable "remember me" functionality for password fields.

3.1.1.15 Prohibit password re-use.

3.1.1.16 If the application manages a credential storage, it should ensure that only cryptographically strong one-way salted hashes of passwords are stored and that the table/file that stores the passwords and keys is writeable only by the application.

3.1.1.17 Create an access control policy to document covering access authorization criteria and/or processes so that access can be properly provisioned/modified and deprovisioned.

3.1.1.18 Restrict access to resources, services, URLs, objects, to only authorized users and must be as per acceptable usage policy of the organization.

3.1.1.19 Enforce authorization controls on every request, including those made by server side scripts and requests from client-side technologies like AJAX.

3.1.1.20 Limit the number of transactions - a single user or device can perform in a given period of time. The transactions/time should be as per the business requirement, but low enough to deter automated attacks.

3.1.1.21 Establish a session inactivity timeout that is as short as possible, based on balancing risk and business requirements. (e.g.: Idle session timeout set at 15 minutes).

3.1.1.22 User's authorization in a long authenticated session should be periodically revalidated to ensure that their privileges have not changed.

3.1.1.23 Logout function should fully terminate the associated session or connections.

3.1.1.24 Disallow persistent logins and enforce periodic session terminations. Termination times should support business requirements and the user should receive sufficient notification to mitigate negative impacts.

3.1.1.25 Generate a new session identifier and deactivate the old one periodically to mitigate certain session hijacking scenarios.

3.1.1.26 Notifications shall be provided, to customers indicating any changes to permissions.

3.1.1.27 Comprehensively log authentication and authorization activities.

3.1.1.28 Implement monitoring tools to identify attacks to detect abuse of authentication and authorization.

3.1.2 Application Security and Secure Development Practices

Software development activities related to Open Banking needs to explicitly address software security in detail to ensure that the software being developed is well-secured.

Organizations are required to adopt the following practices:

3.1.2.1 Define a Secure Systems Development Lifecycle (SDLC) or DevSecOps policy that embeds the principles of "security and privacy by design and default" and supports appropriate security engagement throughout the design and development lifecycle. Embed appropriate change control processes, including formal processes for managing configuration changes.

3.1.2.2 Automated security scanning and testing controls should be performed throughout the SDLC/DevOps, for example:

- Static Application Security Testing (SAST), to identify vulnerabilities in application source code and ensure conformance to coding guidelines and standards.
- Dynamic Application Security Testing (DAST), to identify vulnerabilities in running web applications.

- Software Composition Analysis (SCA), to identify vulnerabilities associated with open-source components.

3.1.2.3 Production data should not be used in non-production deployments.

3.1.2.4 Implement a software change control system to manage versions and record changes to the code both in development and production.

3.1.2.5 Isolate development environments from the production network and provide access only to authorized development and test groups.

3.1.3 Security Hardening

Secure configuration and maintenance of Open Banking environment is critical and should be carefully planned ahead of deployment in production and securely managed through its life cycle. Participants should:

3.1.3.1 Develop configuration standards (e.g.: server hardening baseline) for all system components.

3.1.3.2 Ensure that these standards address known security vulnerabilities and are consistent with industry-accepted system hardening standards.

3.1.3.3 Remove/disable unnecessary software, system services, and drivers.

3.1.3.4 Restrict administrative access to limited group of individuals who require it for executing their job responsibilities.

3.1.3.5 Unused services and open ports should be removed or disabled.

3.1.3.6 Disable or change the password of default accounts.

3.1.3.7 Test and apply vendor supplied patches.

3.1.3.8 Strict control over system modifications must be maintained through a structured approach to change controls. All system changes and alterations shall be implemented exclusively through the organization's approved change and release management process.

3.1.4 Vulnerability Management and Security

It is critical to ensure timely identification, prioritization and treatment of technical vulnerabilities of the open banking environment.

Organizations are required to adopt the following practices:

3.1.4.1 A vulnerability management process is defined, approved, implemented, monitored, measured, periodically reviewed, and updated. The process shall include vulnerability discovery, prioritization, and remediation.

3.1.4.2 Security assessment activities such as vulnerability assessment, baseline security configuration reviews, application security testing, penetration testing and source code reviews of network, systems and applications shall be conducted on periodic basis or whenever significant changes occur to the environment, to identify the existence of vulnerabilities and classify them based on their impact.

3.1.4.3 New technology deployments must undergo security testing prior to go-live to confirm the compliance.

3.1.4.4 Regular Penetration testing should be conducted covering the critical web applications and API infrastructure serving the open banking services.

3.1.4.5 Penetration tests should realistically evaluate organizational ability to defend against the attack by considering methods and tools employed by attackers to attempt to circumvent the security features of a system.

3.1.5 Event Logging

Attackers often take advantage of lack of logging and monitoring to abuse systems without being noticed. Without logging and monitoring, or with insufficient logging and monitoring, it is almost impossible to track suspicious activities and respond to them in a timely fashion.

Organizations must adopt the following practices:

3.1.5.1 Establish consistency in logging practices by ensuring the adoption of a unified logging format and approach across the organization.

3.1.5.2 Logging mechanisms should record both successful and unsuccessful occurrences of security events.

3.1.5.3 Sensitive information should be excluded from logs, eliminating recording unnecessary system details, session identifiers, or passwords.

3.1.5.4 Access to log data should be strictly limited to authorized individuals.

3.1.5.5 Ensure that a mechanism exists to conduct log analysis to identify anomalies and detect potential security incidents.

3.1.5.6 Logging should cover critical details such as input validation failures, authentication attempts, access control failures, potential tampering events, use of invalid or expired session tokens, administrative functions, including changes to the security configuration settings, TLS connection failures, cryptographic events.

3.1.5.7 Implement secure and centralized system (e.g.: SIEM) for comprehensive analysis, detection, alerting, and escalation.

3.1.6 Data and Security Breach Incident Management Reporting

Any security breach resulting in the inadvertent or unlawful loss, alteration, unauthorized disclosure or access to data, or any incident related to cybersecurity, such as malware, ransomware, or denial-of-service attacks, must be promptly reported to the relevant authorities. These occurrences should undergo an assessment to determine their impact, which may be categorized as high, medium, or low. Subsequently, appropriate reporting timelines should be adhered to.

Organizations must adopt the following practices:

3.1.6.1 Conduct periodic cyber risk assessments and risk monitoring in order to anticipate potential data threats, hazards and impacts. Risk assessments must be conducted at least annually, while risk monitoring must be conducted quarterly with defined Key risk indicators (KRIs).

3.1.6.2 Document incident response procedures, clearly outlining roles and responsibilities, defined lines of escalation, and communication protocols for all parties involved.

3.1.6.3 Security incidents should be analyzed and prioritized to determine their severity, scope, and potential impact. A structured process should be established to escalate incidents based on well-defined criteria.

3.1.6.4 Incident response playbooks must be documented to guide effective responses to specific types of cybersecurity incidents. These playbooks should outline clear and actionable steps for isolating and containing the incident to prevent further damage or system abuse. Additionally, they should detail the process for identifying and remediating the root cause of the incident to ensure that the organization is not susceptible to similar attacks in the future.

3.1.6.5 Incident response plans should include details on how to communicate internally and externally during an incident, including reporting mechanisms and external notification procedures.

3.1.6.6 Procedures must be in place for conducting a thorough post-incident analysis or debriefing to identify lessons learned, improve incident response processes, and enhance overall cybersecurity resilience.

3.1.6.7 Incident management should also include instructions on how to restore affected systems and data to normal operation.

3.1.6.8 Cyber incident drills/exercises or simulations should be conducted annually to test and improve an organization's response to cybersecurity incidents. These drills are a proactive measure to assess the effectiveness of incident response plans, identify areas for improvement, and enhance the overall cybersecurity posture.

3.1.7 Disaster Recovery

It's critical for the open banking technology infrastructure to continue to operate correctly and predictably in the presence risks that affects its service availability.

Organizations must follow the following practices:

3.1.7.1 Maintain a documented and approved Disaster Recovery Plan (DRP).

3.1.7.2 Plan should address both transactional and reporting systems of the open banking environment.

3.1.7.3 Establish clearly defined and approved Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO).

3.1.7.4 Disaster recovery plans must incorporate redundancies to address the possibility of in-flight data loss to maintain data integrity.

3.1.7.5 Plan should cover both physical and logical redundancies, well-defined replication intervals, failover and fail-back processes with clearly assigned responsibilities for individuals or roles.

In the event of fail over to the alternate site, it should not fail into an insecure state (i.e., its failure should not leave the service itself, its data, or its environment vulnerable to being compromised, subverted, or exploited to compromise something else).

3.2 Data Security Guidelines

3.2.1 Data Privacy

To ensure protection from information breaches and to create trust by responsible use of data, participants shall establish data privacy measures, to cover the below. The policy shall ensure that all aspects of the data is well managed and fulfils legal and regulatory requirements applicable in Oman.

3.2.1.1 A data privacy policy shall be documented, approved and implemented. The following privacy principles shall be included in the policy at a minimum:

- Collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further processing for archiving purposes shall not be considered to be incompatible with the initial purposes (purpose limitation);
- Collected by obtaining consent from the data subject for the purpose specified. Such consent shall be clear, explicit, unambiguous and involve a clear affirmative action and should be separate from other terms and conditions and should not generally be a precondition of signing up to a service (consent); and
- Processed in a manner that ensures appropriate security of the personal data, including protection against unauthorized or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organizational measures (‘integrity and confidentiality’).

3.2.1.2 Data Protection Impact Assessments (DPIAs) must be conducted regularly in accordance with the organization's policy and whenever significant changes occur in the environment to proactively identify potential risks to sensitive information and ensure that adequate technical safeguards are in place to protect such information.

3.2.1.3 Data classification requirements and privacy by design principles must be integrated into the design of new applications from the initial stages of development.

3.2.2 Information Sharing

Organizations have various business needs and requirements for sharing data, including collaboration with third-party vendors and enhancing operational efficiency. Nevertheless, without implementing adequate data-sharing security measures, organizations expose themselves to the potential violation of data privacy laws and an elevated risk of cyberattacks and data breaches.

Participants are required to ensure:

3.2.2.1 Data sharing is compliant with existing privacy laws and regulations of Oman applicable to the financial services sector namely the Financial Consumer Protection Framework (FCPRF).

3.2.2.2 Data privacy and protection requirements must be clearly communicated to third-party data processors and explicitly outlined in a formal agreement.

3.2.2.3 Enforce technical controls for data sharing, encompassing end-to-end encryption, stringent data access controls, and the establishment of audit trails to monitor data usage.

3.2.3 Data encryption

To protect from unintentional information disclosures, organizations shall implement appropriate information protection measures, using cryptography techniques, and follow the below controls:

3.2.3.1 Enforce robust encryption for data protection, both during transmission and while at rest on the server.

3.2.3.2 Encryption measures should encompass both TLS for secure network connections as well the encryption of payloads or sensitive files.

3.2.3.3 Ensure the integrity of TLS certificates by verifying their validity, accuracy with the correct domain name and non-expiration status.

3.2.3.4 Utilize a single standard TLS implementation to avoid the risk of misconfigurations and potential vulnerabilities.

3.3 API Security Guidelines

3.3.1 Input Validation

Open banking API's must perform input validation to ensure only properly formed data is entering the workflow, preventing malformed data from persisting or triggering malfunction of various downstream components. Input validation must happen as early as possible in the data flow, preferably as soon as the data is received from the external API.

The APIs validation, should cover the following:

3.3.1.1 Data from all sources must be subject to input validation, including not only Internetfacing API endpoints but also backend feeds over portals, from suppliers, partners or vendors

3.3.1.2 Input validation must include checks to prevent the program from executing malicious commands, scripts and codes.

3.3.1.3 Checks for acceptable and unacceptable data types, length criteria, data range and potentially hazardous characters.

3.3.1.4 Checks for preventing the use of eval () or other dynamic code execution features.

3.3.1.5 Checking for redirects – The redirection process should remain controlled within the application's functionality. Prior to redirects, validation should be performed to restrict attackers from submitting malicious content directly to the target, thereby circumventing application logic.

Inputs must be validated on the server side as well. Client-side validation is encouraged but not meant to be the sole control for input validation.

3.3.2 Controls over Excessive Data Exposure

Systems should keep details hidden. Attackers often craft specific requests that causes systems to display addition sensitive information and use common reverse engineer front-end code and sniff API traffic directly to see what data is actually being transmitted.

Developed APIs are required to follow:

3.3.2.1 Should not expose sensitive information in error messages and headers, etc.

3.3.2.2 Any system internal information should be hidden from the request.

3.3.2.3 Should not share authentication details such as passwords and cryptographic keys.

3.3.2.4 Transmission of sensitive or private data through API calls should be restricted to the minimum necessary

3.3.2.5 Data stored or cached at the client-side should be limited to minimize potential security risks.

3.3.2.6 Implement server-side filtering mechanisms to enforce data integrity and security, eliminating the reliance on client-side filtering.

3.3.2.7 Enforce schemas for all API responses to guarantee the preservation of data integrity and enhance the reliability of data exchange.

3.3.2.8 Enforce response checks to prevent accidental data and exception leaks.

3.3.3 Resources and Rate Limiting

Rate limits are necessary for restricting consumption of APIs to a smaller set of API consumers or partners to avoid risks such as Denial of Service.

Rate limiting:

3.3.3.1 Should be enforced using traffic management functionalities / API gateway to place limits or maintain IP address allow and deny lists.

3.3.3.2 Implementation of rate limits should be tailored specific to API methods, clients, and addresses.

3.3.3.3 Implementation should include both dynamic as well static rate limiting techniques.

3.3.3.4 Monitoring tools should be implemented to study and analyze actual API consumption patterns and further adjust rate limits. (Setting blanket static rate limits can result in DOS as attackers will throttle their requests.)

3.3.3.5 Restrict the Payload Size limits to prevent Denial of Service (DoS) and other types of attacks.

3.3.3.6 Network behavioral analysis should be performed to detect Low and Slow or slow rate DDoS attacks.

3.3.4 Controls against Injection Risks

A popular method used by an attacker is to constructs API calls that include OS, Directory Services or DB and other commands that the API or back-end behind it blindly executes. Injection can lead to information disclosure and data loss. It may also lead to DoS, or complete host takeover. It's essential that open banking APIs are protected against malicious injections.

API Injection controls should address the below:

3.3.4.1 Strictly define schema and enforce them at runtime.

3.3.4.2 Ensure that data selection or database queries use parameterized queries and also ORMs or entity frameworks are protected.

3.3.4.3 Ensure that specific security controls to prevent LDAP and OS Injection have been implemented.

3.3.4.4 Limit the number of returned records to prevent mass disclosure in case of injection.

3.3.4.5 Web application firewalls (WAF) should be deployed for runtime protections against attacks. WAF logs should be collected and analyzed to detect and mitigate injection attempts.

3.3.5 API Lifecyle Management

Overtime participants of open banking ecosystem have too many APIs and run the risk of lack of standardization and missing controls. Hence its' critical to carefully govern the life-cycle of APIs including development, publishing including retirement whether its newer APIs or newer versions of APIs.

The API lifecycle controls should:

3.3.5.1 Include clearly defined process for API design, development, modification and retirement to deliver secure API.

3.3.5.2 Ensure the API is release-ready through functional and security testing.

3.3.5.3 Incorporate repeatable process leading to review and update configurations across the entire API stack.

3.3.5.4 Regularly monitor APIs for use and proactively identify and retire unused APIs.

3.3.5.5 Ensure APIs are sunset as per clear documentation, approvals and communication.

4 Appendix

Normative References

#	Framework/Best Practice	Notes
1	CBO CS&RF	
2	IS 27001	
3	Draft NISTIR 8389:Cybersecurity Considerations for Open Banking Technology and Emerging Standards	
4	Bahrain Open Banking Framework - Security Standards and Guidelines	
5	UK Open banking Security & Counter-Fraud good practice guidance	
6	NIST SP 800-95, Guide to Secure Web Services	
7	EBA Guidelines on ICT and security risk management	
8	OWASP API Security Project	