CYBER STANDARDS DOCUMENT

NCSP Application Programming Interface (API) standard

ABSTRACT:
This standard defines the requirements and best practice for development, publishing and consumption of APIs.

ISSUED	November 2025
PLANNED REVIEW DATE	November 2026
DISTRIBUTION	Community Security Policy Framework Members

POLICY VALIDITY STATEMENT
This standard is due for review on the date shown above. After this date, this document may become invalid.

Cyber Standard users should ensure that they are consulting the currently valid version of the documentation.

CONTENTS

COPYRIGHT: Police Digital Service

Community Security Policy Commitment

National Policing and its community members recognise that threats to policing information assets present significant risk to policing operations. National Policing and its community members are committed to managing information security and risk and maintaining an appropriate response to current and emerging threats, as an enabling mechanism for policing to achieve its operational objectives whilst preserving life, property, and civil liberties.

This standard in conjunction with the National Policing Community Security Policy Framework and associated documents sets out National Policing requirements.

Introduction

APIs are essential for enabling communication between applications, allowing data and functionality to be shared seamlessly across platforms, devices and services. As policing environments increasingly rely on APIs to share data, trigger and execute actions and integrate with complex environments, securing these interfaces become critical to maintain trust, ensuring data privacy and preventing unauthorised access or abuse.

This API security standard is organised into three sections to address the full lifecycle of API usage. Section 1 outlines the security requirements for the development of APIs, focusing on secure by design and protection of data. Section 2 covers the requirements for publishing and managing APIs, including access control, monitoring, and versioning. Section 3 provides security guidelines for consuming commercial or third-party APIs, ensuring safe integration and risk management.

Owner

National Chief Information Security Officer (NCISO).

COPYRIGHT: Police Digital Service

Purpose

The purpose of this standard is to establish security requirements and best practices that policing communities can adopt to ensure consistent and robust approach to securing of APIs while developing, managing and consuming third-party APIs. As APIs are inherently insecure, security mechanisms defined in this document aim to harden its posture and secure consumption.

Audience

This standard is aimed at:

- Staff across PDS and policing who develop, publish and consume APIs across ICT systems, either on behalf of National Policing or at a local force level.
- Suppliers acting as service providers or developing products or services for PDS or policing.
- Auditors and penetration testers providing assurance services to PDS or policing.

Scope

1. The standard outlines the security requirements for anyone who is involved in API development, the maintenance and management of those APIs and the consumption of third-party APIs.
2. This standard is applicable to any infrastructure, system or application that is utilising APIs to communicate, perform an action and transfer data.

Definitions

Application programming interfaces (APIs) - are a set of rules and protocols that allows different software components to communicate with each other. An API defines how software components must securely interact. It provides a way for developers to access specific features or data of an application, service, or platform without needing to understand its internal workings.

Requirements

The API requirements section outlines the key considerations for the development, management, and consumption of the API. It defines the purpose and scope of the API from three perspectives: how it should be built securely (development), how it will be governed and maintained (management), and what must be considered when APIs, including third party, will be consumed by policing organisations.

1. API Secure Development

This section outlines security controls during the design and implementation of APIs.

1.1 API Secure Development

Reference	Minimum requirement	Control reference	Compliance Metric
1.1.1 Design principles	Adhere to common SDLC frameworks, API standards and API security guidance for secure development lifecycle (SDLC). Consider using the following API standards: • OWASP ASVS 5.0 • OpenAPI • OAuth 2.0 • OWASP API Security Top 10	NIST CSF PR.PS-06, ID.AM-08	Review the evidence of established processes for secure API development lifecycle.

COPYRIGHT: Police Digital Service

	Follow Secure by Design principles to ensure security is embedded right from the start.	NIST CSF PR.PS-06,	Review the evidence of established SbD process, documentation and produced artifacts when securely developing APIs.

1.2 Authentication and Authorisation

Reference	Minimum requirement	Control reference	Compliance Metric
1.2.1 Zero trust	Authenticate and authorise every request, both external and internal.	NIST CSF PR.AA-01, PR.AA-03, PR.AA-05	Review designs and documented security mechanism confirming secure implementation and alignment with zero trust approach. Output from security testing.
1.2.1 Authentication	Use strong and modern authentication mechanisms such as OAuth 2.0, OpenID Connect, API keys or certificates for secure authentication. *** API keys-based authentication is considered secure with other security mechanisms such as HTTPS/TLS and should be used only for API client authentication***	NIST CSF PR.AA-03	Review designs and documented security mechanism confirming secure implementation of modern authentication mechanisms. Output from security testing. Alignment with the NCSP Cryptography standard.
	Use JWT (JSON Web Tokens) or OAuth tokens for secure and scalable session management. Ensure expiry times are set as per specific use case and risk position.	NIST CSF PR.AA-03	Review designs and documented security mechanism confirming secure implementation of modern authentication mechanisms. Output from security testing.

	Mutual TLS (mTLS) should be considered for additional layer of security by verifying both a client and a server.	NIST CSF PR.AA-03	Review designs and documented security mechanism confirming secure implementation of modern authentication mechanisms. Output from security testing.
	Ensure API credentials/secrets/ API keys are securely stored in a secured and centralised secrets manager (with a secure storage backend such as HSM or cloud KMS) and managed securely. Short-lived credentials (or for authenticating to low-value applications) may use software-backed storage.	NIST CSF PR.AA-01, PR.AA-04	Review of how credentials and secrets are stored and managed including process, documentation and logs.
	Credentials/token lifetime should be only set to appropriate amount of time to the use case, service and threat (for example, short-lived access token can be anything from 5 minutes to 24h whereas a long-lived refresh token could be up to 90 days). Credentials/tokens must be automatically rotated or renewed when they expire, if required, and the process of rotation should be automated	NIST CSF PR.AA-01	Review of design decision and risk assessment documentation justifying credentials lifetime, security testing. Evidence of credentials rotation.

	with no human involvement. Long-term access keys (i.e., 60 days) should be avoided.		
	Effective identity management for human and non-human identities (e.g., services, workloads, AI agents, API tokens) is critical for secure API access. Please refer to NCSP Identity and Access Management standard for details.		Alignment with the NCSP Identity and Access Management standard.
1.2.3 Authorisation	Implement Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) to ensure that access is granted based on the principle of least privilege.	NIST CSF PR.AA-01, PR.AA-05	Review designs and documented security mechanism confirming secure implementation of granular authorisation mechanism and policies. Output from security testing.
	Implement scope-based access controls to restrict the API access to specific actions, resources and properties depending on the user's role or service identity to ensure read and write granularity.	NIST CSF PR.AA-01, PR.AA-05, NIST 800-53 AC-5	Review designs and documented scope-based access confirming secure implementation of granular authorisation mechanism and policies. Output from security testing.
	Continuously validate permissions for each request.	NIST CSF PR.AA-01, PR.AA-05, DE.CM-09	Review API design and logs confirming that permissions are being tested as designed. Output from security testing.

1.3 Input Validation, Output Encoding and Data Sanitisation

Reference	Minimum requirement	Control reference	Compliance Metric
1.3.1 Input validation	Ensure that a service strictly validates all incoming requests via API gateways/WAF (e.g., query parameters, headers, body content) to match the API's definition, that all expected fields are present and if the correct type.	NIST CSF PR.PS-06, PR.DS-01, PR.DS-10, DE.CM-09, NIST 800-53 SI-10(6)	Review API design and documented security mechanism implemented to validate all requests. Output from security testing.
	Reject any data that does not conform to expected input types or validation rules.	NIST CSF PR.PS-06, PR.DS-10, DE.CM-09, NIST 800-53 SI-10(6)	Review API design and documented security mechanism implemented to validate all requests. Output from security testing.
1.3.2 Data Sanitisation	Sanitise input data to prevent injection attacks, such as SQL injection, XML injection, and cross-site scripting (XSS).	NIST CSF PR.PS-06, PR.DS-10, DE.CM-09, NIST 800-53 SI-10(6)	Review API design and documented security mechanism implemented to sanitise input. Output from security testing.
	Use security libraries or frameworks that automatically sanitize input to prevent injection-based attacks.	NIST CSF PR.PS-06,PR.DS-10, DE.CM-09, NIST 800-53 SI-10(6)	Review of selected and documented libraries or framework and justification process.

1.4 Error Handling

Reference	Minimum requirement	Control reference	Compliance Metric
1.4.1 Standardised Error Responses	Responses must be designed to prevent leaking sensitive system details such as stack traces, database errors, internal	NIST CSF PD.DS-10, PR.PS-04 NIST 800-53 SI-11	Review API design, documented error responses.

	paths or exception messages and providing only the necessary generic information.		Output from security testing confirming error responses is as expected.
	Any technical errors must be simplified, consistent and made more user-friendly, providing enough detail in error messages and with relevant status codes.	NIST CSF PD.DS-10, PR.PS-04 NIST 800-53 SI-11	Review of documented errors and expected responses and process to validate it.
	Full range of errors an API can produce, must be well understood, documented and security tested and verified.	NIST CSF PD.DS-10, PR.PS-04 NIST 800-53 SI-11	Review of documented errors and expected responses and process to validate it.
1.4.2 Error Testing	API error handling testing must be evaluated to identify and address potential issues with valid/invalid requests and edge cases.	NIST CSF PD.DS-10, PR.PS-06 NIST 800-53 SI-11, SA-3	Review of documented errors and expected responses and process to validate it. Output from security testing confirming error responses is as expected.

1.4 Data Security and Encryption

Reference	Minimum requirement	Control reference	Compliance Metric
1.5.1 API security	Always use strong and appropriate algorithms and protocols to secure APIs. See Cryptography standard for details.	NIST CSF PR.DS-1, PR.DS-2, PR.DS-10, NIST 800-53 SC-13	Penetration testing, configuration check, vulnerability assessment. Alignment with the NCSP Cryptography standard.
	All API communications must use TLS (Transport Layer Security) with strong cipher suites (e.g., AES-256) to protect data integrity and	NIST CSF PR.DS-1, PR.DS-10, NIST 800-53 SC-13	Review of design decisions, penetration testing, configuration check, vulnerability assessment.

	confidentiality during transmission.		Align with the NCSP Cryptography standard.
	APIs must be configured to only allow encrypted communication (e.g., avoid using HTTP and only allow HTTPS).	NIST CSF PR.DS-1, PR.DS-10, NIST 800-53 SC-13	Review of design decisions, penetration testing, configuration check, vulnerability assessment.
	Sensitive data, including API encryption keys, should be encrypted at rest using industry-standard encryption algorithms such as AES-256.	NIST CSF PR.DS-2, NIST 800-53 SC-13	Review of design decisions, penetration testing, configuration check, vulnerability assessment.
	API URLs must never include sensitive information such as user credentials, keys or tokens.	NIST CSF PR.DS-10	Review of design decisions, penetration testing, configuration check, vulnerability assessment.
1.5.2 Minimal Data Exposure	Only return necessary data fields and never rely on the client side to filter sensitive data.	NIST CSF PD.DS-10, PR.PS-04 NIST 800-53 SI-11	Review API design and documented security mechanism implemented to filter sensitive data and logs. Output from security testing.

1.6 Logging and Monitoring

Reference	Minimum requirement	Control reference	Compliance Metric
1.6.1 Logging	Log all API requests and responses to enable comprehensive visibility into the API's operational landscape.	NIST CSF DE.CM-09, PR.PS-04	Review of logging configurations and log audit.
	Log important security-related events such as: • Authentication ○ successful and failed logons	NIST CSF DE.CM-09, PR.PS-04	Review of logging configurations and log audit.

	○ token generation, refresh and revocation • Authorisation ○ access denied errors ○ privileged actions and modification of access • Input validation failures ○ requests with malformed or suspicious payloads • Rate limiting/throttling events • Error and exceptions logs • Sensitive resource access		
	Sensitive data such as API keys, access tokens and PII information should never be logged.	NIST CSF DE.CM-09, PR.PS-04	Review of documented logging configurations and log audit.
1.6.2 Format	Logs should be written using a format appropriate for the consuming log management solution, providing enough detail to identify any malicious risks. API call logs must include: • Timestamps • IP addresses • HTTP method and endpoint information	NIST CSF PR.PS-04	Review of documented configurations and policies.

COPYRIGHT: Police Digital Service

	• Request and response details		
1.6.3 Monitoring	Ensure continuous and real-time monitoring of APIs to identify suspicious activities and anomalies.	NIST CSF DE.CM-01, DE.CM-09	Review of documented configurations and policies.
	Use Security Information and Event Management (SIEM) tool to centralise logging, detection and alerting of APIs.	NIST CSF DE.CM-01, DE.CM-09, PR.PS-04	Review of design decisions, documented configurations and policies.

2 API Management and publishing

This section covers practices for securely managing and publishing of APIs.

2.1 API Management

Reference	Minimum requirement	Control reference	Compliance Metric
2.1.1 API management and governance.	API policies, standards, principles and process must be documented to define how APIs are designed, secured and managed.	NIST CSF PR.PS-01, NIST 800-53 CM-1	Review of documented API designs, polices and lifecycle processes.
	A centralised API platform should be used for effective management and governance of organisation APIs throughout their lifecycle (design, development, deployment, deprecation) and/or inherited in mergers and acquisitions.	NIST CSF PR.PS-01, PR.PS-06, ID.AM-08	Review of documented API designs, polices and lifecycle processes.
2.1.2 API Discovery and documentation	APIs should be automatically discovered, identified, catalogued, classified and	NIST CSF PR.PS-01, PR.PS-02, ID.AM-02, ID.AM-05, ID.AM-08	Review of documented API designs, polices and lifecycle processes.

COPYRIGHT: Police Digital Service

	documented for both production and lower environments, including what data the API has access to. Ensure consistent approach to APIs documentation is used.		
2.1.3 API versioning	Use consistent and clear API versioning strategy to ensure backward compatibility and minimise risks when updating or deprecating APIs. API documentation must be updated with information about the release when a new API version is published. Consideration should be given to capture the following API information: introduction and how API works, endpoint descriptions, authentication setup, folder structure, requests and headers, sample responses (success and error cases).	NIST CSF PR.PS-01, PR.PS-02, PR.PS-06	Review of documented processes, polices and configurations.
2.1.4 API deprecation	Any deprecated APIs (replaced or suspended) must not be used.	NIST CSF PR.PS-01, PR.PS-02, PR.PS-06	Review of documented processes, polices and configurations. Penetration testing of deprecated APIs to ensure they are not reachable.

2.2 API Gateway

Reference	Minimum requirement	Control reference	Compliance Metric
2.2.1 API Gateway	Use appropriate gateway patterns (centralised, hybrid or distributed) for the required use cases, to enforce access control, execute rules, request validation and file content inspection/scanning, throttling, and logging of all API connections.	NIST CSF PR.IR-01, PR.IR-03, PR.DS-2, PR.DS-10, DE.CM-01, DE.CM-09, PR.PS-04	Review of documented design decision, policies and configurations. Penetration testing, configuration check, vulnerability assessment.
	API gateway should include features like IP allowed geo-blocking and bit mitigation to prevent abuse or malicious use of APIs.	NIST CSF PR.IR-01, DE.CM-01, DE.CM-09, PR.PS-04	Review of documented design decision, policies and configurations. Penetration testing, configuration check, vulnerability assessment.

2.3 API Rate limiting and Throttling

Reference	Minimum requirement	Control reference	Compliance Metric
2.3.1 Rate limiting	Implement rate limiting and throttling to control the number of API requests a client can make within a defined period. Individual assessment per API, baselining and fine tuning based on business needs will be required to set the appropriate rate limit.	NIST CSF PR.DS-10, PR.IR-01, PR.IR-03, DE.CM-01	Review of documented design decision, policies and configurations. Penetration testing, configuration check, vulnerability assessment.
	Consider using global and endpoint-specific limits. For example: GET/products:1000 req/min	NIST CSF PR.IR-01, PR.PS-01, PR.IR-03	Review of documented design decision and configurations. Penetration testing, configuration check, vulnerability assessment.

	POST /orders: 100 req/min POST /login: 5 req/min		
	Apply a multi-layered rate limiting approach: • Per API key/Client ID to limit requests per authenticated user or app. • Per IP address to block unauthenticated traffic, Tor exit nodes and well-known proxies. • Per endpoint or method to set stricter limits on sensitive operations (e.g., /login, /checkout). • Per resource to rate limit access to critical resources such as database or specific service.	NIST CSF PR.IR-01, PR.PS-01, PR.IR-03	Review of documented design decision and configurations. Penetration testing, configuration check, vulnerability assessment.
	Consider using algorithm-based rate limiting to handle API requests.	NIST CSF PR.IR-01, PR.PS-01, PR.IR-03	Review of documented design decision and configurations. Penetration testing, configuration check, vulnerability assessment.
	Apply rate limits on error-returning endpoints to mitigate attacks.	NIST CSF PR.IR-01, PR.PS-01, PR.IR-03	Review of documented design decision and configurations. Penetration testing, configuration check, vulnerability assessment.
2.3.2 Circuit breaking	Consider implementing circuit breaking pattern to enhance system resilience and fault tolerance.	NIST CSF PR.IR-01, PR.PS-01, PR.IR-03	Review of documented design decision and configurations. Penetration testing, configuration check, vulnerability assessment.

2.3.3 Abuse Detection	Monitor and detect anomalous API usage patterns, such as unusually high request volumes or suspicious access from unauthorized locations.	NIST CSF PR.IR-01, PR.IR-04, DE.CM-01, DE.CM-09	Review of documented configurations, policies. Output from a tool monitoring traffic.

2.4 API Key and Secret Management

Reference	Minimum requirement	Control reference	Compliance Metric
2.4.1 Secrets Management	Secure vaults must be used for secure storage of secrets and API keys, using hardware security modules (HSMs) or cloud key management services (KMS).	NIST CSF PR.AA-1, PR.PS-01, PR.DS-5 NIST 800-53v5 IA-5(1)	Documented design decisions, enforced system policies and configurations. Alignment with NCSP Cryptography standard.
	Principle of least privilege must be followed to protect access to API keys. Only approved account role must be used to access Secrets Management service and with restricted access to an IP/specific range.	NIST CSF PR.AA-05, PR.PS-01	Documented design decisions, enforced system policies and configurations.
	Lifespan of secrets and API keys must be restricted by default. Lifespan of secrets and keys rotation should be set to appropriate amount of time to the use case and threat.	NIST CSF PR.PS-01, PR.AA-05	Documented design decisions, enforced system policies and configurations. Review of logs from secrets management service.

	Immediate rotation should be also triggered upon significant change or an incident.		
	Regular review and revocation of access rights must be conducted. Process should be automated.	NIST CSF PR.AA-05	Review of system access polices and logs.
	API keys must never be hard-coded.	NIST CSF PR.PS-01, PR.AA-01, PR.PS-06	Review of policies and secure configuration. Output from security testing.

2.5 API Security Testing and Vulnerability Management

Reference	Minimum requirement	Control reference	Compliance Metric
2.5.1 Secure Software Development Lifecycle (SDLC)	Integrate API security testing into the secure software development lifecycle.	NIST CSF PR.PS-06, ID.RA-01, ID.IM-02	Review of SDLC lifecycle process and output from security testing throughout different stages of the SDLC lifecycle.
2.5.2 Security testing	Use automated tools to perform static and dynamic security testing to identify risks and vulnerabilities across all APIs, dependencies and any other associated components.	NIST CSF PR.PS-06, ID.RA-01, ID.RA-05, ID.IM-02	Review of SDLC lifecycle process and output from security testing throughout different stages of the SDLC lifecycle.
	Conduct regular penetration testing focused on API endpoints to identify vulnerabilities.	NIST CSF PR.PS-06, ID.RA-01, ID.IM-02	Review of SDLC lifecycle process and output from security testing throughout different stages of the SDLC lifecycle.
	Prioritise testing against known OWASP API Security Top 10 threats, such as broken	NIST CSF PR.PS-06, ID.RA-01, ID.IM-02	Review of SDLC lifecycle process and output from security testing throughout different stages of the SDLC lifecycle.

	authentication, excessive data exposure, and improper rate limiting, others.		
	Test for error handling vulnerabilities.	NIST CSF PR.PS-06, ID.RA-01, ID.IM-02	Review of SDLC lifecycle process and output from security testing throughout different stages of the SDLC lifecycle.
	Establish vulnerability management process for quickly addressing and patching any vulnerabilities identified in APIs.	NIST CSF PR.PS-06, ID.RA-01, ID.IM-02	Review of vulnerability management process and output from security testing throughout different stages of the SDLC lifecycle.
	Ensure timely deployment of security patches for underlying libraries, dependencies, frameworks, or platform components.	NIST CSF PR.PS-06, ID.RA-01, ID.IM-02	Review of vulnerability management process and evidence of vulnerability mitigation.

3. Consumption of third-party APIs

This section provides guidance for safely integrating with and relying on third-party APIs.

3.1 Vendor Risk Assessment

Reference	Minimum requirement	Control reference	Compliance Metric
3.1.1 Vendor assessment	Perform due diligence and TPAP/security assessments before integrating third-party APIs.	NIST CSF GV.SC-06, GV.SC-07	Compliance with TPAP and output from security testing SCA/SAST tooling.
	Ensure detailed API documentation is available, maintained and reviewed.	NIST CSF PR.PS-01, GV.SC-07, ID.RA-05	Review of documentation, processes and standards.
	Ensure permissions to test APIs have been	NIST CSF GV.SC-05, ID.RA-05	Output from security tests.

	granted by 3rd party prior security testing or security testing reports are provided (high-level).		
	Ensure the vendor complies with relevant standards (e.g., ISO 27001, Cyber Essentials Plus, SOC2).	NIST 800-53 CA-0(1)	Evidence of compliance with standards and frameworks.
3.1.2 Compliance	Review third-party API terms of service and data handling policies.	NIST 800-53 CA-0(1)	Review of documented and up to date policies.
	Ensure third-party APIs meet policing privacy and compliance requirements.	NIST 800-53 CA-0(1)	Review of documented and up to date policies and compliance documentation.

3.2 Traffic and Data Monitoring

Reference	Minimum requirement	Control reference	Compliance Metric
3.2.1	Monitor API consumption for anomalies and changes in patterns and behaviours.	NIST CSF DE.CM-01, DE.CM-09	Output from monitoring tools.

3.3 Security Updates and Change Management

Reference	Minimum requirement	Control reference	Compliance Metric
3.3.1	Subscribe to vendor change notifications (e.g., API version changes, deprecations, dependencies).	NIST GV.SC-07, PR.PS-01, PR.PS-02	Review of documented process.
	Always verify changes do not break API	NIST GV.SC-07, ID.IM-01, PR.PS-01	Review of documented process.

	functionality or introduce security regressions.		

Communication approach

This document will be communicated as follows:

- Internal peer review by the members of the National Cyber Policy & Standards Working Group (NCPSWG), which includes PDS and representatives from participating forces.
- Presentation to the National Cyber Policy & Standards Board (NCPSB) for approval.
- Formal publication and external distribution to PDS community, police forces and associated bodies.

Measurables generated by adopting this standard can also form part of regular cyber management reporting.

For external use (outside PDS), this standard should be distributed within IT teams to help complete an initial gap analysis which can inform any implementation plan. This implementation plan can be shared with force SIROs / Security Management Forum. Consideration should also be given to raising awareness amongst force personnel of the implementation of this standard where it may affect them.

COPYRIGHT: Police Digital Service

Review Cycle

This standard will be reviewed at least annually (from the date of publication) and following any major change to Information Assurance (IA) strategy, membership of the community, or an identified major change to the cyber threat landscape. This ensures IA requirements are reviewed and that the standard continues to meet the objectives and strategies of the police service.

Document Compliance Requirements

(Adapt according to Force or PDS Policy needs.)

Equality Impact Assessment

Forces should consider local impacts as a result of this standard being applied.

Document Information

Document Location

https://knowledgehub.group/web/national-standards/policing-standards

Revision History

Version	Author	Description	Date
0.1	PDS Cyber	NCSP API Standard	5/08/2025
1.0	PDS Cyber	Incorporate NCPSWG suggestions	5/09/25

Approvals

Version	Name	Role	Date
1.0	NCPSB	National Cyber Policy & Standards Board	27/11/25

COPYRIGHT: Police Digital Service

Document References

Document Name	Version	Date
ISF - Standard of Good Practice (for Information Security)	v2024	03/2024
NIST Cyber Security Framework	V2.0	04/2024
10 Steps to Cyber Security - NCSC.GOV.UK	Web Page	05/2021
NCSC API Guidance	Web Page	04/2025
OWASP API Security Top 10	Web Page	04/2023
NIST Guidelines for API Protection for Cloud-Native Systems	Web Page	05/2025
OWASP REST Security	Web Page	07/2024
Open API Specification	Web Page	03/2025