Security Standard – Microservices Architecture (SS-028)

Chief Security Office

Date: 21/11/2023

Department for Work & Pensions

This Microservices Architecture Security Standard is part of a suite of standards, designed to promote consistency across the Department for Work and Pensions (DWP), and supplier base with regards to the implementation and management of security controls. For the purposes of this standard, the term DWP and Department are used interchangeably.

Technical security standards form part of the DWP Digital Blueprint which is a living body of security principles, architectural patterns, code of practice, practices and radars, that aim to support Product Delivery Units (PDUs) and suppliers in delivering the DWP and HMG Digital Strategy. Security standards and policies considered appropriate for public viewing are published here:

https://www.gov.uk/government/publications/dwp-procurement-security-policies-and-standards

Technical security standards cross-refer to each other where needed, so can be confidently used together. They contain both mandatory and advisory elements, described in consistent language (see table below).

Table 1 – Terms

Term	Intention
must	denotes a requirement: a mandatory element.
should	should denotes a recommendation: an advisory element.
may	denotes approval.
might	denotes a possibility.
can	denotes both capability and possibility.
is/are	is/are denotes a description.

1. Contents

2. Revision History

Version	Author	Description	Date
1.0		First published version	25/10/2017
2.0		Full update in line with current best practices and standards; • Updated Intro, purpose, audience, scope; added reference to CIS v8 security controls • Added NIST CSF references 11.1.1 Threat model 11.1.2 In-support software versions 11.1.4 Microservices cannot run on the host network. 11.1.7 Separation of environments 11.1.10 Application footprint 11.1.11 data in multiple security contexts 11.2.1 Service registry **must** have validation checks 11.2.2 OAUTH and JSON Web Tokens 11.2.3 Approved algorithms 11.2.4 Removed 11.2.5 Differing TTL values 11.2.10 OAUTH; Shared library tokens; secrets not hardcoded 11.2.11 Hash-based message authentication 11.3.2 Including code-based 11.3.3 Internal services and mutual authentication pattern 11.4.1 UI/UX allowlisted ports 11.4.2 Rate limited procedure invocation 11.5.4 No direct communication 11.6.1 All nodes 11.6.2 Event types; logging of sensitive data	21/11/2023

3. Approval History

Version	Name	Role	Date
1.0		Chief Security Officer	25/10/2017
2.0		Chief Security Officer	21/11/2023

This document will be reviewed for continued completeness, relevancy and accuracy within 1 year of being granted "final" status, and at yearly intervals thereafter.

4. Compliance

Compliance with this standard will be verified through various methods, including but not limited to;

- controls tests performed by first-line teams and by 2nd line activities (e.g. security testing teams)
- security assurance activities to ensure that Architectural Design and delivery are appropriate and aligned to applicable Authority Security Standards. [See Security Assurance Strategy – Ref. I].
- independent external audit

Results of these will be fed back to the appropriate Authority Risk and System Owners.

5. Exceptions Process

In this document the term **"must"** is used in bold letters to indicate a mandatory security measure. Any exceptions to the application of this standard, or where specific security measures cannot be adhered to, **must** be presented to the Authority. This **must** be carried out prior to deployment and managed through the design caveats or exception process.

Such exception requests will invoke the Risk Management process to clarify the potential impact of any deviation to the configuration detailed in this standard.

Exceptions to the standard **must** be maintained on a risk register for accountability, traceability, and security governance reporting to senior management.

6. Audience

This document is intended for, but not necessarily limited to, technical architects, engineers, developers, security teams, project teams, including suppliers engaged in the design, development, implementation and operation of systems, services and applications.

7. Accessibility Requirements

Users of this standard **must** consider accessibility design requirements as appropriate. Further information on accessibility standards can be found in Appendix F.

8. Introduction

This Microservices Architecture Security Standard defines the minimum technical security measures that **must** be implemented for use within the Authority.

As this standard only provides minimum measures, they **should** be exceeded as appropriate depending on the threats and risks that need to be addressed, the sensitivity of the data, and in keeping with latest security enhancements.

The security measures are derived from industry best practice i.e. guidance published by NIST, CIS and OWASP (see Appendix C for full list external references) and support the implementation of appropriate security controls as selected by Authority or our third party providers, such as the CIS Critical Security Controls v8 controls set. [see External References]

Every effort has been made to ensure the security measures are vendor and technology agnostic as far as possible; this is to ensure greater applicability of the standard regardless of the technologies used. The security measures **may** be implemented in different ways, depending on the technology choices and business requirements in question.

The aim of this standard is to:

- ensure security controls that are applicable to microservices architecture are implemented consistently across the Department and by third party providers where applicable.
- mitigate risks from common threats and vulnerabilities associated with microservices architecture, to an acceptable level for operation.
- support the achievement of security outcomes described in Appendix A.

Technical security standards ultimately support the achievement of security outcomes sought by the Department. They set the expectations for what needs to be done to achieve them and why, and provide an objective, measurable statement of the Authority's existing security posture in a number of important areas. The outcomes are based on the official NIST sub-categories where possible to ensure close alignment with the NIST Cyber Security Framework (CSF), and are enabled by the implementation of controls from the CIS Critical Security Controls v8 controls set. [see External References]. Those relevant to the subject of each standard can be found in Appendix A of every technical security standard.

9. Purpose

The purpose of this standard is to ensure that Authority systems and services are designed, configured, deployed, and managed consistently to protect against typical threats at the OFFICIAL tier.

This standard also serves to provide a baseline in which assurance and compliance activities can be carried out, so that the Department can be assured that security obligations are being met or exceeded.

10. Scope

This standard applies to all microservices architecture deployments within the Department and supplier base (contracted third party providers), for the purposes of delivering applications and services that handle Authority data.

Any queries regarding the security measures laid out in this standard **should** be sent to the Authority.

11. Minimum Technical Security Measures

The following section defines the minimum security measures that **must** be implemented to achieve the security outcomes described in Appendix A. For ease of reference, the official NIST sub-category ID is provided against each security measure e.g. PR.PT-3, to indicate which outcome(s) it contributes towards. Refer to Appendix A for full description of outcomes.

11.1 Microservices Architecture Attack Surface

Reference	Minimum Technical Security Measures	NIST ID
11.1.1	Microservices **must** run inside an approved application container technology. (As defined within SS-011 Containerisation Security Standard [Ref. A]). The Authority Architecture Blueprint Technology Radars [Ref. B] describes which specific technologies are approved for use within the Authority. As Microservices scale, the threat model **must** be reviewed.	PR.DS-5
11.1.2	Systems Administrators **must** ensure the container Runtime environment software is in vendor support, or is at least ‘n-1’ for open source software.	ID.RA-1
11.1.3	Containers **must** be configured according to Authority approved Security Best Practices. (see SS-011 Containerisation Security Standard [Ref. A]).	PR.DS-5
11.1.4	Microservices **must** not run on the host network, as they could access other services running on the same host.	PR.DS-5

11.1.5	Communication with a Microservice **must** be done via Gateway service to provide load balancing and a standard set of security capabilities for the Microservices to consume. As a minimum these services **must** provide, authentication, authorisation, logging and alerting.	PR.AC-1 PR.PT-1 DE.CM-1
11.1.6	Containers hosting Microservices **must** be limited to exposing a single port or the minimal number of ports required to provide the service. All other ports **must** be explicitly blocked.	PR.PT-3
11.1.7	The production environment **must** be clearly separated from development, testing or QA environments, with controls implemented to ensure separation.	PR.DS-7
11.1.8	The operation of the Microservice, its resource consumption and performance **must** be monitored and spikes in consumption addressed through capacity management activities.	PR.DS-4
11.1.9	Microservices **must** be protected by a Defence in Depth approach. This will include – Filtering of communication flows, Authentication and Authorisation of access to a Microservice and the use of encryption technologies.	PR.DS-1 PR.DS-2
11.1.10	The microservice architecture must have the smallest application footprint possible and designed using a single responsibility principle.	PR.PT-3
11.1.11	Ensure that a single deployed microservice does not process data in multiple security contexts e.g. one service that includes both low and high privileged operations that would traditionally be separated.	PR.AC-4

11.2 Authentication Requirements

Please refer to SS-001 pt.1 Access & Authentication Security Standard [Ref. E] for more information on API Authentication requirements.

Reference	Minimum Technical Security Measures	NIST ID
11.2.1	The Authentication model for Microservices **must** be defined early in the software development lifecycle. This includes the use of federated identity (i.e. where authorised users access multiple applications and domains using a single set of credentials)where appropriate. The gateway **must** authenticate to an approved and assured authentication service. Service registry **must** have validation checks to ensure that only legitimate services are allowed.	PR.AC-7
11.2.2	The latest versions of OAUTH or JSON Web Token (for public facing services), using security best practices **must** be used.	PR.AC-7
11.2.3	The Token based authentication mechanism **must** use an approved Authority algorithm to generate the security token that is in line with SS-007 Use of Cryptography Security Standard [Ref. C].	PR.AC-1 PR.AC-7
11.2.5	Authentication Tokens **must** have an associated TTL (values for which may differ from system to system) to prevent replay attacks.	PR.AC-6 PR.AC-7
11.2.6	TTL's for Authentication Tokens **must** be updated with a new expiration time each time a token is re-validated.	PR.AC-6 PR.AC-7
11.2.7	Expired Tokens **must not** be allowed to be replayed as a legitimate authentication request. Also a suitable HTTP error code (401) should be returned to the user.	PR.AC-1
11.2.8	Expired Tokens **must** be regularly purged from memory.	PR.AC-1
11.2.9	It **must not** be possible to (maliciously) replay a previously delegated request.	PR.AC-1
11.2.10	Public Key Infrastructure best practices **must** be adhered to for certificate exchange when using OAUTH or JSON web tokens for authentication. (see SS-002 PKI & Key Management Security Standard [Ref. D]). In addition, when shared libraries are used; • Token expiry times **must** be as short as possible • Token secrets **must** not be part of the library code • Token secrets **must** not be hard coded	PR.AC-1

11.2.11	Authentication tokens **must** be cryptographically signed, or protected by a hash-based message authentication code (HMAC) scheme, see (see SS-002 PKI & Key Management Security Standard [Ref. D]).	PR.AC-1 PR.DS-2
11.2.12	Where the use of passwords is required they must be created and managed in accordance with SS-001 pt.1 Access & Authentication Security Standard [Ref. E].	PR.AC-1

11.3 Deployments and Best Practice

Reference	Minimum Technical Security Measures	NIST ID
11.3.1	Policy configuration **must** be applied in an automated manner to enforce segmentation.	PR.AC-5
11.3.2	Underlying software, certificates, services and infrastructure upon which Microservices are reliant (including those that are code-based) **must** be patched (or updated) in line with SS-033 Security Patching Standard (ref. G].	PR.IP-12
11.3.3	All traffic between endpoints in the microservices architecture (including internal services) that carry authentication credentials **must** utilise TLS 1.2. SS-007 Use of Cryptography Security Standard [Ref. C] provides further guidance on the use of TLS, as does SP-006 Channel Encryption & Mutual Authentication Security Design Pattern [Ref. J].	PR.DS-2

11.4 Microservices Layer and UI/UX

Reference	Minimum Technical Security Measures	NIST ID
11.4.1	The security controls between the UX functions and the microservices layer **must** only permit connections on necessary and allowlisted TCP and UDP ports.	PR.PT-3
11.4.2	Each logic function triggering a stored procedure on the application components **must** limit the rate of procedure invocation based on user ID.	PR.DS-5

11.5 Data and Messaging Privilege Restriction

Reference	Minimum Technical Security Measures	NIST ID
11.5.1	Services **must** only be able to access messaging channels required for their function.	PR.PT-3
11.5.2	Access to any given messaging channel **must** be limited to functionality required (such as read only, write, etc).	PR.PT-3
11.5.3	Messaging credentials **must** be protected appropriately at rest and in transit.	PR.DS-1 PR.DS-2
11.5.4	Clients **must** communicate to the target service via a single gateway URL, rather than communicating directly to the service. Keep-alive TLS connections **must** be in use for frequently interacting services. (See also section 11.1.5)	PR.DS-2

11.6 Monitoring and Logging

Reference	Minimum Technical Security Measures	NIST ID
11.6.1	Individual Microservices and the Microservice gateway **must** produce appropriate logs in compliance with SS-012 Protective Monitoring Security Standard. [Ref. H]. Security monitoring **must** cover all nodes.	PR.PT-1
11.6.2	All API requests **must** be logged to a centralised logging and monitoring system, in compliance with SS-012 Protective Monitoring Security Standard [Ref. H]. Input validation errors, extra parameters errors, crashes and core dumps **must** be logged. Care **must** be taken to prevent sensitive and/or personal data being logged.	PR.PT-1

12 Appendices

Appendix A Security Outcomes

The minimum security measures defined in this standard contribute to the achievement of security outcomes described in the table below. For consistency, the official NIST Sub-category IDs have been carried through to the standards.

Table 1 – List of Security Outcomes Mapping

NIST Ref	Security Outcome (sub-category)	Related Security measure
ID.RA-1	Asset vulnerabilities are identified and documented	11.1.2
PR.AC-1	Identities and credentials are issued, managed, verified, revoked, and audited for authorized devices, users and processes	11.1.5, 11.2.3, 11.2.7, 11.2.8, 11.2.9, 11.2.10, 11.2.11, 11.2.12
PR.AC-4	Access permissions and authorizations are managed, incorporating the principles of least privilege and separation of duties	11.1.11
PR.AC-5	Network integrity is protected (e.g., network segregation, network segmentation)	11.3.1
PR.AC-6	Identities are proofed and bound to credentials and asserted in interactions	11.2.5, 11.2.6
PR.AC-7	Users, devices, and other assets are authenticated (e.g., single-factor, multi-factor) commensurate with the risk of the transaction (e.g., individuals' security and privacy risks and other organizational risks)	11.2.1, 11.2.2, 11.2.3, 11.2.5, 11.2.6
PR.DS-1	Data-at-rest is protected	11.1.9, 11.5.3
PR.DS-2	Data-in-transit is protected	11.1.9, 11.2.11, 11.3.3, 11.5.3, 11.5.4
PR.DS-4	Adequate capacity to ensure availability is maintained	11.1.8

PR.DS-5	Protections against data leaks are implemented	11.1.1, 11.1.3, 11.1.4, 11.4.2
PR.DS-7	The development and testing environment(s) are separate from the production environment	11.1.7
PR.IP-12	A vulnerability management plan is developed and implemented	11.3.2
PR.PT-1	Audit/log records are determined, documented, implemented, and reviewed in accordance with policy	11.1.5, 11.6.1, 11.6.2
PR.PT-3	The principle of least functionality is incorporated by configuring systems to provide only essential capabilities	11.1.6, 11.1.10, 11.4.1, 11.5.1, 11.5.2
DE.CM-1	The network is monitored to detect potential cybersecurity events	11.1.5

Appendix B Internal References

Below, is a list of internal documents that **should** be read in conjunction with this standard.

Table 2 – Internal References

Ref	Document	Publicly Available*
A	SS-011 Containerisation Security Standard	Yes
B	DWP Architecture Blueprint (Technology Radars)	No
C	SS-007 Use of Cryptography Security Standard	Yes
D	SS-002 PKI & Key Management Security Standard	Yes
E	SS-001 pt.1 Access & Authentication Security Standard	Yes
F	SS-003 Software Development Security Standard	Yes
G	SS-033 Security Patching Standard	Yes
H	SS-012 Protective Monitoring Security Standard	Yes
I	Security Assurance Strategy	No
J	SP-006 Channel Encryption & Mutual Authentication Security Design Pattern.	No

Requests to access non-publicly available documents **should be made to the Authority.*

Appendix C External References

The following publications and guidance were considered in the development of this standard and **should** be referred to for further guidance.

Table 3 – External References

External Documents List
CIS Critical Security Controls v8 controls set
NIST SP 800-204: Security Strategies for Microservices-based Application Systems
OWASP Top 10 Web Application Security Risks

Appendix D Abbreviations

Table 4 – Abbreviations

Abbreviation	Definition
CA	Certificate Authority
DDA	Digital Design Authority
DWP	Department for Work and Pensions (DWP)
Port	TCP/IP port number
Runtime	Software providing the execution environment for applications within an Application Container.

Appendix E Definition of Terms

Table 5 – Glossary

Term	Definition
Cryptographic Items	All logical and physical items used to achieve confidentiality, integrity, non-repudiation and accountability; including, but not limited to: devices, products, systems, key variables and code systems.
Cryptographic Key Material	Any parameter passed to an encryption cipher which influences the output of the algorithm (with the exception of the message itself).
Decoupling	It's an approach in IT development and operations where two or more systems work or are connected without being directly connected. In theory, this means that a change can be made to one service without the developer having to worry about how the change will impact other services.
Service registry	The service registry service is used by microservices that are coming online to publish their locations in a process called service registration and is also used by microservices seeking to discover registered services.

Appendix F Accessibility artefacts

A variety of accessibility guidance is available from the below URL, that includes:

DWP Digital Accessibility Policy | DWP Intranet

https://accessibility-manual.dwp.gov.uk/

https://www.gov.uk/guidance/guidance-and-tools-for-digital-accessibility

https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps