# Security Standard - Microservices Architecture (SS-028)

## Chief Security Office

**Date: March 2020**

Department for Work & Pensions

## 1. Revision History

| Version | Author | Description | Date |
|---|---|---|---|
| 0.0a | | First Draft | 17/01/2017 |
| 0.0b | | Inclusion of additional controls | 28/02/2017 |
| 0.0c | | Uplift to new template | 29/02/2017 |
| 0.0d | | Review and update of template and content. | 16/06/2017 |
| 0.0e | | Updated following internal review | 09/07/2017 |
| 0.0f | | Minor update | 31/08/2017 |
| 0.0g | | Minor update following review | 10/10/2017 |
| 1.0 | | First Published version | 25/10/2017 |
| 1.1 | | Version for external publication | 30/03/2020 |

## 2. Distribution

| Version | Role/Area | Role | Date |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

## 3. Approval History

| Version | Approver Title | Role | Date |
|---|---|---|---|
| 1.0 | | Chief Security Officer | 25/10/2017 |
| 1.1 | | Chief Security Officer | 30/03/2020 |

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

# Contents

## 4. Introduction

4.1. This document describes the best practices and security control requirements for the deployment of Microservices within the Authority estate. As this area of technology matures so shall the controls in this standard to reflect this developing technology space.

## 5. Purpose

5.1. The purpose of this standard is to list the security requirements in relation to how to deploy, implement and control the usage of Microservices within the Authority.

5.2. Secondly, this standard provides a reference to conduct compliance based technical security audits against.

5.3. Suppliers should use this documentation to ensure that the security best practises for their system are being adequately and accurately addressed.

## 6. Exceptions

6.1. Any exceptions to the application of this standard, or where controls cannot be adhered to, MUST be presented to the Authority where appropriate. This activity MUST be carried out prior to deployment and managed through the design caveats or exception process.

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

6.3. Exceptions to this standard MUST be maintained on a risk register for accountability, traceability and security governance reporting to the Authority.

## 7. Audience

7.1. This standard is intended for Suppliers, system administrators, security groups, and IT staff involved in securing environments for Authority systems and applications and provided the security requirements on how to manage, implement and configure Micro Services Architecture.

## 8. Scope

8.1. This standard is applicable to systems handling data within the OFFICIAL tier of the Government Security Classification Policy (GSCP). This includes OFFICIAL information that attracts the SENSITIVE handling caveat. All

implementations of Microservices MUST meet all the requirements in this standard or gain authorization from the Authority.

8.2. The security control requirements laid out in this standard are product agnostic and applicable for all implementations of Microservice architecture that are provisioned for Authority use.

8.3. In the event of uncertainty on the controls laid out in this standard please contact the Authority for guidance and support on items which require clarification.

## 9. Security Controls Assurance

9.1. Controls presented in this standard or referred to via this standard may be subjected to a formalised IT Health Check penetration test to provide evidence of adequacy and effectiveness.

## 10. Technical Security Control Requirements

In this document the term MUST in upper case is used to indicate an absolute requirement. Failure to meet these requirements will require a formal exemption (see section [6. Exceptions] above).

## 10.1. Micro services Architecture Attack Surface

| Reference | Security Control requirement |
|---|---|
| 10.1.1 | Microservices MUST run inside an approved application container technology. (As defined within the SS-011 Containerisation Security Standard). As Microservices scale, the attack surface of the application inevitably increases. A method of reducing the overall attack surface is to use approved containers which expose the minimum attack surface |
| 10.1.2 | Systems Administrators MUST ensure the container Runtime environment is maintained in accordance with the current policy on S/W versions. |
| 10.1.3 | Containers MUST be configured according to Authority Approved Security Best Practices. (see SS-011 Containerisation Security Standard). |
| 10.1.4 | IP Filtering technologies MUST be applied to the Microservices host environment as a minimum to control connectivity to the Microservices executing on the host. |
| 10.1.5 | 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. |
| 10.1.6 | 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. |
| 10.1.7 | A tool to monitor and visualise inter-service communication MUST be deployed as part of the management capabilities of the Microservices architecture. |
| 10.1.8 | A baseline of normal communication activities MUST be created and thresholds on monitoring and logging tools MUST be configured to trigger when events such as traffic spikes, or unusual traffic flows are detected. |
| 10.1.9 | Production, development and QA environments MUST be isolated, running on logically separate networks. |
| 10.1.10 | A catalogue of API’s and their characteristics MUST be published so that developers of consuming software are clear on the function, URI, and other specific requirements for using the service. |
| 10.1.11 | The operation of the Microservice, its resource consumption and performance MUST be monitored and spikes in consumption addressed through capacity management activities. |
| 10.1.12 | 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. |

| Reference | Security Control requirement |
|---|---|
| 10.1.13 | Except during traffic inspection, the Microservices API gateway, must not retain OFFICIAL/OFFICIAL SENSITIVE information in memory. |

10.2 Authentication Requirements

| Reference | Security Control requirement |
|---|---|
| 10.2.1 | The Authentication model for Microservices MUST be defined early in the software development lifecycle. This includes the use of federated identity where appropriate. The gateway must authenticate to an approved and assured authentication service. |
| 10.2.2 | A well-known and secure open standard protocol for centralised authentication using tokens MUST be leveraged. |
| 10.2.3 | The Token based authentication mechanism MUST use an algorithm to generate the security token that follows the guidance in the SS-007 Use of Cryptography Security Standard |
| 10.2.4 | Multiple active authentications per user or process MUST be allowed, with a strict upper bound set to mitigate any potential Denial of Service attacks. |
| 10.2.5 | Authentication Tokens MUST have an associated TTL to prevent replay attacks. |
| 10.2.6 | TTL's for Authentication Tokens MUST be updated with a new expiration time each time a token is re-validated |
| 10.2.7 | 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. |
| 10.2.8 | Expired Tokens MUST be regularly purged from memory. |
| 10.2.9 | It MUST NOT be possible to (maliciously) replay a previously delegated request. |
| 10.2.10 | Public Key Infrastructure best practices MUST be adhered to for certificate exchange when using JSON web tokens for authentication. (see SS-002 PKI and Key Management Security Standard) |
| 10.2.11 | Authentication Tokens MUST be encrypted. This is to mitigate the exposure time should a token become compromised. |
| 10.2.12 | Every API endpoint MUST authenticate to the gateway. |
| 10.2.13 | The gateway Must implement a mechanism to restrict the number and alert on repeated authentication failures. |
| 10.2.14 | Where the use of a password is required and has been agreed, through the exceptions process, then the passwords must be created and managed in accordance with the SS-001 Access and Authentication Controls standard. |

10.3 API keys and Signed Requests

| Reference | Security Control requirement |
|---|---|
| 10.3.1 | Services MUST have integrity mechanisms in place to handle data that has been transmitted to them to ensure that it has not been altered in transmission. |
| 10.3.2 | The function call hierarchy MUST be configured in such a way that a threat actor is not able to intercept API calls across the network and leverage the data in such a way to launch a replay attack against the service. |
| 10.3.3 | Each service MUST have a totally unique API key for calling another service. This key MUST comprise of a unique Service ID and a User ID at minimum. |
| 10.3.4 | Communication of API keys MUST utilise encryption in transit (TLS) |
| 10.3.5 | All API requests MUST be cryptographically signed. Signatures MUST include request parameter data, service ID, API key, and originating time stamp as a minimum. |
| 10.3.6 | Functions MUST be created using an approved SDK. |
| 10.3.7 | Development of Microservices MUST follow the guidance published in the SS-003 Software Development Security Standard |

10.4 Deployments and Best Practices

| Reference | Security Control requirement |
|---|---|
| 10.4.1 | Policy configuration MUST be applied in an automated manner to enforce segmentation as part of the Orchestration processes. |
| 10.4.2 | Underlying software, certificates, services and infrastructure upon which Microservices are reliant MUST be patched in line with SS-033 Patching Security Standard. |
| 10.4.3 | Deployment and updates MUST be automated using an Orchestration service. |
| 10.4.4 | All traffic between endpoints in the microservices architecture including that carrying authentication credentials MUST utilise TLS. The SS-007 Use Of Cryptography security standard will give further guidance on the use of TLS. |

10.5 Data and Messaging Privilege Restriction

| Reference | Security Control requirement |
|---|---|
| 10.5.1 | API calls made by users and systems MUST be limited to only those necessary for those users or systems to perform their functions |
| 10.5.2 | Data available to services MUST be limited to the minimum required for them to function |
| 10.5.3 | Database credentials MUST provide access to only the minimum amount of data possible to discharge the function for which those credentials are issued |

| Reference | Security Control requirement |
|---|---|
| 10.5.4 | Database credentials MUST provide access only to functionality and operations required to discharge the function for which those credentials are issued |
| 10.5.5 | Services MUST only be able to access messaging channels required for their function. |
| 10.5.6 | Access to any given messaging channel MUST be limited to functionality required (such as read only, write, etc.) |
| 10.5.7 | Messaging credentials MUST be protected appropriately at rest and in Transit. |

10.6 Monitoring and Logging

| Reference | Security Control requirement |
|---|---|
| 10.6.1 | Individual Microservices and the Microservice gateway MUST produce appropriate logs in compliance with the SS-012 Protective Monitoring Security Standard |
| 10.6.2 | All API requests MUST be logged to a centralised logging and monitoring system, in compliance with the SS-012 Protective Monitoring Security Standard. |
| 10.6.3 | Microservices MUST log performance and throughput metrics<br>A baseline set of normal metrics MUST be established. |
| 10.6.4 | Thresholds MUST be configured on logging and monitoring system(s) such that abnormal metrics trigger alerts for investigation |

## 11. Compliance

Compliance with this standard MUST occur as follows:

| Compliance | Due Date |
|---|---|
| **On-going** | From the first day of approval |
| **Retrospective** | Within 6 months of the approval of the standard. |

## 12. Accessibility

No user interfaces are included in this standard and accessibility is not applicable as part of this standard. However, it is deemed that Suppliers implementing this standard are obliged to incorporate accessibility functions where necessary.

## 13. Definition of Terms

| | |
|---|---|
| **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). |

## 14. Glossary

| | |
|---|---|
| **AES** | Advanced Encryption Standard – defined in FIPS 197. Different modes of operation are covered in different documents. |
| **CA** | Certificate Authority |
| **DA** | Design Authority (DA) |
| **Authority** | The Authority refers to the Department for Work and Pensions |
| **Port** | TCP/IP port number |
| **Runtime** | Software providing the execution environment for applications within an Application Container. |
| **SUPPLIER** | Is inclusive of Contractor, their employees or any sub-contractors used |