Security Standard – Containerisation (SS-011)

Chief Security Office

Date: 22/08/2022

Department for Work & Pensions

This Containerisation 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 Authority 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-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. Table of Contents

2. Revision history

Version	Author	Description	Date
1.0		First published version	18/09/2017
2.0		Full update in line with current best practices and standards; • Updated Intro, purpose, audience, scope • Shortened overview of containerisation • Written to be vendor and technology agnostic as far as possible to increase applicability • Replaced use of technical control requirements to minimum security measures • Re-formatted document to categorise security measures under five headings to correspond with each core containerisation component • Added new requirements (security measures) under each category • Added NIST sub-category references against each security measure • Added new table in Appendix A which list security outcomes the measures support the achievement of • Updated references and included links to external publications etc. 9.3.4 Added reference to build team. 9.4.4 Replaced CMDB with code repository or container registry.	22/08/2022

3. Approval history

Version	Name	Role	Date

1.0		Chief Security Officer	18/09/2017
2.0		Chief Security Officer	22/08/2022

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

4. Compliance

Security assurance teams will verify compliance with this standard through various methods, including but not limited to, internal and external audits, and feed back to the appropriate Authority Risk and System Owner.

4.1 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.

5. Audience

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

6. Accessibility statement

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

7. Introduction

This standard defines the minimum technical security measures that **must** be implemented to secure systems providing services to the Authority utilising application containerisation technology.

For the purposes of this standard, containerisation can be described as an operating system (OS) level virtualisation method where applications run in isolated user spaces, called containers, while using the same shared host. A container is essentially a stand-alone, all-in-one package for a software application. They contain everything that an application needs, such as its libraries, binaries, configuration files and software dependencies, all encapsulated into an independent, self-contained unit. The container itself is abstracted from the host OS, with only limited access to underlying resources (if configured securely).

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 the 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 core containerisation components are implemented consistently across the Authority and by third party providers where applicable.
- mitigate risks from common threats and vulnerabilities associated with containerisation technology, 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 Authority. They set the expectations for what needs to be done to achieve them, and why. The outcomes are based on the official NIST sub-categories where possible to ensure close alignment with the NIST Cyber Security Framework (CSF) and can be found in Appendix A of every technical security standard.

7.1 Purpose

The purpose of this standard is to ensure systems and services utilising application containerisation technology to process Authority data 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 Authority can be assured that security obligations are being met or exceeded.

8. Scope

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

All forms of virtualisation other than technologies that support containerisation are outside the scope of this document.

Also, this standard only addresses the core components of containerisation technology - platform (host OS), orchestrators, images, registries (repositories) and containers. Because this standard only looks at the core components, the measures should be applicable to most container deployments regardless of the container technology or vendor, host OS platform, or location i.e. public or private cloud.

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

9. 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.

9.1 Platform Hardening (host)

Reference	Minimum Technical Security Measures	NIST ID
9.1.1	The OS's supporting the containers **must** be in vendor support and hardened in accordance with SS-008 - Server Operating System Security Standard [Ref. A] or an approved 'Gold Build' (where applicable) to reduce the attack surface of the host as much as possible. Industry benchmarks **must** be used where available e.g. CIS Docker Benchmarks.	PR.PT-3
9.1.2	Container-specific OS's **should** be used whenever possible instead of general-purpose ones to reduce the attack surface. These are specifically designed to host containers and have other services and	PR.PT-3

	functionality disabled by default, providing mitigation against typical risks and hardening activities associated with general purpose operating systems (OS).	
9.1.3	Containers **must** not be used as a method to separate data or services that have different security profiles.	ID.AM-5
9.1.4	Hosts **must** be set up such that, by default, network stacks within the containers on the host cannot inter-communicate. When containers are run, they **must** obtain their own individual network stack.	PR.AC-P5
9.1.5	Hosts that run containers **must** only run containers and not run other applications, like web servers or databases outside of containers. The host OS **should** not run unnecessary services, such as a print spooler, that increase its attack and patching surface. For the avoidance of doubt, this measure does not apply to services/agents deployed on hosts as part of the OS build approved by the Authority.	PR.AC-P5, PR.PT-3
9.1.6	Hosts **must** be continuously scanned for vulnerabilities and updates applied in accordance with the DWP Technical Vulnerability Management Policy [Ref. B] and SS-033 – Security Patching Standard [Ref. C], not just to the container runtime but also to lower level components such as the kernel that containers rely upon for secure compartmentalised operation.	DE.CM-8, RS.MI-3, RS.AN-5

9.1.7	The OS and any associated Gold Build images **must** be kept up to date in accordance with SS-033 - Security Patching Standard [Ref. C], not only with security updates, but also the latest component updates recommended by the vendor. This is particularly important for the kernel and container runtime components as newer releases of these components often add additional security protections and capabilities beyond simply correcting vulnerabilities.	PR.IP-12, RS.MI-3
9.1.8	Host OS's **must** be used solely for hosting containers. There **must** also be no application level dependencies provided by the host, instead all components and dependencies should be packaged and deployed in containers.	PR.PT-3, DE.CM-3
9.1.9	All authentication to the hosts **must** be audited, and any login anomalies monitored, and any escalations to perform privileged operations **must** be logged in accordance with SS-012 - Protective Monitoring Standard [Ref. I]. This will make it possible to identify anomalous access patterns such as an individual logging on to a host directly and running privileged commands to manipulate containers.	DE.CM-7
9.1.10	Access to the host OS **shall** be based on the need-to-have and least privilege principle.	PR.AC-4

9.2 Orchestrator

Reference	Minimum Technical Security Measures	NIST ID

9.2.1	The principle of least privilege **must** be implemented for Orchestrators, in which users are only granted the ability to perform specific actions on the specific hosts, containers, and images their role requires.	PR.AC-4, PR.PT-3
9.2.2	Access to cluster-wide administrative accounts **must** be tightly controlled and monitored in line with SS-012 -Protective Monitoring Standard [Ref. I], as these accounts provide the ability to affect all resources in the environment. Strong authentication methods **must** be considered as appropriate, such as requiring multi-factor authentication instead of just a password.	PR.AC-4
9.2.3	Single sign-on **must** be implemented where possible, as this will simplify the orchestrator authentication experience, make it easier for users to use strong authentication credentials, and centralise auditing of access, making anomaly detection more effective. Any elevated access e.g. for administrative purposes, must force a re-authentication of user credentials, which must include a valid MFA interaction utilising a hard-token.	PR.AC-7, PR.PT-1, DE.CM-7
9.2.4	Orchestrators **must** be configured to separate network traffic into discrete virtual networks based on security profiles where possible. For example, public-facing apps with increased threat exposure could share a virtual network.	PR.AC-5

9.2.5	Orchestrators **must** be configured to isolate deployments to specific sets of hosts by sensitivity levels in accordance with the DWP Security Classification Policy [Ref. H]. This particular approach will vary depending on the Orchestrator in use, but the general model is to define rules that prevent high sensitivity workloads from being placed on the same host as those running lower sensitivity workloads.	ID.AM-5
9.2.6	Orchestrators **must** ensure that nodes are securely introduced to a cluster, have a persistent identity throughout their lifecycle, and can provide an accurate inventory of nodes and their connectivity states.	ID.AM-2
9.2.7	Orchestrators used for managing the build, distribution and run phases of the application container lifecycle **must** be supported by a CMDB or asset inventory.	PR.IP-3
9.2.8	Clusters **must** be configured to monitor resource consumption patterns of individual containers to aid detection of unanticipated spikes in resource usage that could lead to non-availability of critical resources.	DE.CM-1, DE.AE-1
9.2.9	Containers **must** be grouped according to their purpose, sensitivity, and threat posture on a single host OS kernel to allow for additional defence in depth.	ID.AM-5

9.2.10	All default settings for dashboards, clusters and endpoints **must** be reviewed and appropriately hardened using available benchmarks i.e., CIS, STIG etc., this is to minimise vulnerabilities due to misconfigurations. Under no circumstance **must** nodes and clusters be deployed using default configuration without assessing the risk implications.	PR.IP-1

9.3 Images

Reference	Minimum Technical Security Measures	NIST ID
9.3.1	Image configurations **must** be reviewed against secure configuration best practices where available e.g., CIS Benchmarks, to reduce the attack surface.	PR.PT-3
9.3.2	Images **must** be configured to run as non-privileged users where technically possible. Where this cannot be achieved, functionalities such as user namespace remapping **must** be used to map the container user to a non-privileged user on the host OS.	PR.PT-3
9.3.3	Secrets **must** be stored outside of images and provided dynamically at runtime as needed.	PR.AC-1
9.3.4	All images regardless of where they are sourced, **must** be vetted, tested, and validated. All images **must** also be digitally signed in accordance with SS-002 - PKI and Key Management Standard [Ref. D] before being added to the image registries. Separation of duties **must** be maintained between	PR.DS-6

	the build team and those approving the criteria for acceptance of an image.	
9.3.5	Images **must** be scanned for embedded malware and vulnerabilities, when acquired, before deployment and following significant changes.	DE.CM-4
9.3.6	Offline (stored) images **must** be kept up to date, and all runtime images re-created using the latest images. Furthermore, images **must** be regularly assessed or tested for compatibility with the wider ICT estate as to minimise the risk of 'breaking' applications that are still dependent on older software versions. Where updating runtime images with the latest images causes incompatibility issues with a given application, use of older images **must** be subject to a risk assessment and formal risk owner approval.	RS.MI-3
9.3.7	When any changes are made to the base image or dependent image (e.g., patching a vulnerability), the corresponding image **must** be recreated, and the container re-launched using the modified image. This ensure a single master, or gold image is maintained for any service.	PR.IP-1
9.3.8	Routine checks **must** be carried out (at least every 2 weeks as a minimum) to ensure the latest images available are being used. This process **should** be automated where possible.	PR.IP-1
9.3.9	Only approved container images **must** be used as a source (see 9.3.4).	PR.IP-1

9.3.10	Code base (ideally within source code management) used for image builds **must** be backed up in accordance with SS-035 – Secure Backup and Restore Security Standard [Ref. K].	PR.IP-4, RC.RP-1

9.4 Registry (Repository)

Reference	Minimum Technical Security Measures	NIST ID
9.4.1	Connection to Production registries from tools, orchestrators and container runtimes **must** be over encrypted channels in accordance with SS-007 - Use of Cryptography [Ref. E].	PR.DS-2
9.4.2	To mitigate against inadvertently using out of date and potentially vulnerable images, container registries **must** be regularly pruned of images that are no longer required (at least once a quarter). This process **should** be automated where possible.	PR.IP-12
9.4.3	Operational teams **must** access images using immutable names that specify the discrete version of images to be used. As such, deployment tasks **must** specify the exact versions to be used.	PR.IP-1, PR.IP-3
9.4.4	Identification of all images and versions **must** be maintained at all times in a code repository or container registry.	PR.IP-1, PR.IP-3
9.4.5	All access to registries **must** be authenticated and authorised in accordance with SS-001 (part 1) - Access and Authentication Controls Standard [Ref. F] and SS-001 (part 2) - Privileged User Access Controls Standard [Ref. G].	PR.IP-1, PR.IP-3
9.4.6	Any write access to registries **must** be authenticated to ensure that only images from trusted entities can be added to it.	PR.AC-4, PR.PT-3

9.4.7	Images **must** be approved by authorised personnel and only pushed to a registry after they have passed a security assessment process i.e. passed a vulnerability scan. This process **should** be automated where possible.	ID.RA-1, PR.IP-1, PR.IP-3
9.4.8	The number of accounts accessing the registry **must** be limited to mitigate against the threat of account hijacking.	PR.AC-1
9.4.9	Image Registries **must** support signed images.	PR.DS-6
9.4.10	Image Registries **must** not allow unrestricted network access.	PR.PT-4
9.4.11	Logging and Alerting **must** be enabled on Image Registries where supported to detect anomalous activity.	PR.PT-1
9.4.12	Threat detection capability **must** be utilised for Image Registries where supported.	RS.AN-1
9.4.13	Logs **must** be forwarded from Image Registries to security monitoring tools to support threat detection in accordance with SS-012 - Protective Monitoring Standard [Ref. I].	PR.PT-1
9.4.14	Artefacts maintained within image registries and associated meta data **must** be backed up in accordance with SS-035 – Secure Backup and Restore Security Standard [Ref. K].	PR.IP-4, RC.RP-1

9.5 Containers

Reference	Minimum Technical Security Measures	
9.5.1	Mandatory access control (MAC) technologies **must** be considered as appropriate, to provide enhanced control and isolation for containers.	PR.AC-4
9.5.2	Separate environments **must** be used for development, test, production, and other scenarios, each with specific controls to provide role-based access control for container deployment and management activities.	PR.DS-7
9.5.3	Container runtimes **must** be monitored for vulnerabilities, and when problems are detected, they **must** be remediated (in accordance with SS-033 – Security Patching Standard [Ref. C]). A vulnerable runtime exposes all containers it supports, as well as the host itself, to potentially significant risk. Security tools **should** be used to look for CVE vulnerabilities in runtimes deployed, to ensure that Orchestrators only allow deployments of properly maintained runtimes.	DE.CM-8, RS.MI-3
9.5.4	Systems administrations **must** apply the default deny rule to all container capabilities, and only allow those capabilities needed through an explicit 'Allow List'.	PR.PT-3

9.5.5	All container creation **must** be associated with individual user identities and logged to provide a clear audit trail of activity.	PR.AC-1, PR.PT-1, DE.CM-3
9.5.6	SSH / RDP and other administration tools designed to provide remote shells to host **must** be disabled within containers.	PR.AC-3
9.5.7	Although created inside a container, all logs **must** be managed by a process executing outside the container and **must** not be managed by a process running inside the container.	PR.PT-1
9.5.8	As part of the container configuration, commands and capabilities not required to support the service provided by the container **must** be removed or disabled.	PR.PT-3
9.5.9	Containers **must** externally present only the necessary ports and services required by the consuming business or administrative services.	PR.PT-3, PR.PT-4
9.5.10	Network specific operations **must** be disabled inside containers. Network configuration **must** be applied to the container at start-up and not be dynamically modified.	PR.PT-3, PR.PT-4
9.5.11	Under no circumstance **must** containers be able to mount sensitive directories on a host's file system, especially those containing configuration settings for the operating system.	PR.PT-3

9.5.12	To mitigate malicious network activity related to packet spoofing, access to raw sockets **must** not be allowed within the container.	PR.PT-3
9.5.13	Run File systems in containers **must** be read only to prevent malicious scripts being saved or files being overwritten.	PR.AC-4
9.5.14	Containers **must** not be allowed to load modules dynamically. All code that is required to execute within the container must be within the container image.	PR.IP-1
9.5.15	During the build process, the identity of all dependencies **must** be verified and authenticated using code signing and signatures.	PR.DS-6
9.5.16	Build processes **must** enforce the use of the most up to date image dependencies, where appropriate.	PR.IP-1
9.5.17	Members of the test team **must** only be given access to images in a test environment and the hosts used for running them and **should** only be able to manipulate the containers they created.	PR.AC-4
9.5.18	If an application has multiple components that need to run distinctly from one another, then each component **should** be deployed in its own container.	PR.PT-3
9.5.19	Services between containers or groups of containers, **must** be exposed only via port binding, with ports explicitly opened in a container configuration file, specifying that the only permitted	PR.PT-3

	connection to a given application is from another container.	
9.5.20	Containers **must** be configured in accordance with the requirements set out in SS-012 - Protective Monitoring Standard [Ref I].	DE.CM-7
9.5.21	All diagnostics in production **should** be done via log files or other approved tooling which negates direct access to running containers.	PR.AC-4
9.5.22	If supported, the container runtime **must** be configured to enforce running signed images only, this will prevent images from external, un-vetted sources from being used.	PR.DS-6

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 2 – List of Security Outcomes Mapping

Ref	Security Outcome (sub-category)	Related Security measure
ID.RA-1	Asset vulnerabilities are identified and documented	9.4.7
ID.AM-2	Software platforms and applications within the organization are inventoried	9.2.6
ID.AM-5	Resources (e.g., hardware, devices, data, and software) are prioritized based on their classification, criticality, and business value	9.1.3, 9.2.5, 9.2.9
PR.PT-1	Audit/log records are determined, documented, implemented, and reviewed in accordance with policy	9.2.3, 9.4.11, 9.4.13, 9.5.5, 9.5.7
PR.PT-3	The principle of least functionality is incorporated by configuring systems to provide only essential capabilities	9.1.1, 9.1.2, 9.1.5, 9.1.8, 9.2.1, 9.3.1, 9.3.2, 9.4.6, 9.5.4, 9.5.8, 9.5.9, 9.5.10, 9.5.11, 9.5.12, 9.5.18, 9.5.19

PR.PT-4	Communications and control networks are protected	9.4.10, 9.5.9, 9.5.10
PR.IP-1	A baseline configuration of information technology/industrial control systems is created and maintained incorporating security principles (e.g. concept of least functionality)	9.2.10, 9.3.7, 9.3.8, 9.3.9, 9.4.3, 9.4.4, 9.4.5, 9.4.7, 9.5.14, 9.5.16
PR.IP-3	Configuration change control processes are in place	9.2.7, 9.4.3, 9.4.4, 9.4.5, 9.4.7
PR.IP-4	Backups of information are conducted, maintained, and tested	9.3.10, 9.4.14
PR.IP-12	A vulnerability management plan is developed and implemented	9.1.7, 9.4.2
PR.AC-1	Identities and credentials are issued, managed, verified, revoked, and audited for authorized devices, users and processes	9.3.3, 9.4.8, 9.5.5
PR.AC-3	Remote access is managed	9.5.6
PR.AC-4	Access permissions and authorizations are managed, incorporating the principles of least privilege and separation of duties	9.1.10, 9.2.1, 9.2.2, 9.4.6, 9.5.1, 9.5.13, 9.5.17, 9.5.21
PR.AC-5	PR.AC-5: Network integrity is protected (e.g., network segregation, network segmentation)	9.1.4, 9.1.5, 9.2.4
PR.AC-7	Users, devices, and other assets are authenticated (e.g., single-factor, multi-factor) commensurate with the risk of the transaction	9.2.3

	(e.g., individuals' security and privacy risks and other organizational risks)	
PR.DS-2	Data-in-transit is protected	9.4.1
PR.DS-6	Integrity checking mechanisms are used to verify software, firmware, and information integrity	9.3.4, 9.4.9, 9.5.15, 9.5.22
PR.DS-7	The development and testing environment(s) are separate from the production environment	9.5.2
DE.AE-1	A baseline of network operations and expected data flows for users and systems is established and managed	9.2.8
DE.CM-1	The network is monitored to detect potential cybersecurity events	9.2.8
DE.CM-3	Personnel activity is monitored to detect potential cybersecurity events	9.1.8, 9.5.5
DE.CM-4	Malicious code is detected	9.3.5
DE.CM-7	Monitoring for unauthorized personnel, connections, devices, and software is performed	9.1.9, 9.2.3, 9.5.20
DE.CM-8	Vulnerability scans are performed	9.1.6 9.5.3
RS.MI-3	Newly identified vulnerabilities are mitigated or documented as accepted risks	9.1.6, 9.1.7, 9.3.6, 9.5.3

RS.AN-5	Processes are established to receive, analyse and respond to vulnerabilities disclosed to the organization from internal and external sources (e.g. internal testing, security bulletins, or security researchers)	9.1.6
RS.AN-1	Notifications from detection systems are investigated	9.4.12
RC.RP-1	Recovery plan is executed during or after a cybersecurity incident	9.3.10, 9.4.14

Appendix B. Internal references

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

Table 3 – Internal References

Ref	Document	Publicly Available*
A	SS-008 – Server Operating System	Yes
B	DWP Technical Vulnerability Management Policy	Yes
C	SS-033 – Security Patching	Yes
D	SS-002 – PKI and Key Management	Yes
E	SS-007 – Use of Cryptography	Yes
F	SS-001 (part 1) – Access and Authentication Controls	Yes
G	SS-001 (part 2) – Privileged User Access Controls	Yes
H	DWP Security Classification Policy	Yes
I	SS-012 - Protective Monitoring Standard	Yes
J	SS-015 – Malware Protection	Yes
K	SS-035 – Secure Backup and Restore	tbc

Requests to access non-publicly available documents **should be made to an assigned DWP Security Architect or DWP Contracts/Supplier Manager.*

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 4 – External References

External Documents List
Amazon Elastic Container Service Best Practices Guide
NIST – Cyber security Framework – 2018-04-16
NIST – 800-53 – Rev 5 – Security and Privacy Controls for Information
NIST Special Publication 800-190 – Application Container Security Guide (September 2017)
NIST 8176 – Security Assurance Requirements for Linux Application Container Deployments
CIS v8 Critical Security Controls
CIS Docker Benchmark (Version 1.3.0)
ISO/IEC 27002:2013
Cloud Security Alliance Cloud Controls Matrix Version 4
OWASP Application Security Verification Standard (ASVS)
OWASP Container Security Verification Standard (Version 1.0)
UK Government Technical Standard
National Security Agency – Kubernetes Hardening Guidance (Version 1.0)

Appendix D. Abbreviations

Table 5 – Abbreviations

Abbreviation	Definition	Owner
CIS	Centre for Internet Security	Industry body
CMDB	Configuration Management Database	Industry term
CVE	Common Vulnerabilities and Exposures	Industry term
DWP	Department of Work and Pensions.	UK Government
GSCP	Government Security Classification Policy	UK Government
ISO	International Organization for Standardization	Industry term
MAC	Mandatory Access Control	Industry term
NIST	National Institute of Standards and Technology	US Government
NIST – CSF	National Institute of Standards and Technology – Cyber Security Framework	US Government
OS	Operating System	Industry term
OWASP	Open Web Application Security Project	Open source
OWASP ASVS	(OWASP) Application Security Verification Standard	Open source
RDP	Remote Desktop Protocol	Industry term
SSH	Secure Shell	Industry term

Appendix E. Glossary

Table 6 – Glossary

Term	Definition
Image	A package that contains all files required to run a container.
OFFICIAL	Information classification mark, identified in the Government Security Classification Policy.
Container	A method for packaging and securely running an application within an application virtualisation environment. Also known as an application container or a server application container.
Container runtime	The environment for each container; comprised of binaries coordinating multiple operating systems components that isolate resources and resource usage for running containers.
Container specific OS	A minimalistic host operating system explicitly designed to only run containers.
Gold Build	A detailed build document and associated master image that has been evaluated for security issues, which then forms a template used to deploy replicated instances across the network.
Namespace isolation	A form of isolation that limits which resources a container may interact with.
Orchestrator	A tool that enables DevOps personas or automation working on their behalf to pull images from registries, deploy those images into containers, and manage the running of containers.
Registry	A service that allows developers to easily store images as they are created, tag and catalogue images for identification

	and version control to aid in the discovery and reuse and find and download images that others have created.
Virtualisation	The simulation of the software and/or hardware upon which other software runs.

Appendix F. Accessibility artefacts

A variety of accessibility guidance is available from the below URLs:

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

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