WHITE PAPER – MAY 2021

RED HAT OPENSHIFT CONTAINER PLATFORM FOR PCI DSS V3.2.1

APPLICABILITY GUIDE TO ASSIST PAYMENT CARD CUSTOMERS IN PCI DSS 3.2.1 DEPLOYMENTS

COALFIRE OPINION SERIES – VERSION 2.0

BYRON ESTRADA, MSC CYBERSECURITY
CHRIS KRUEGER, CISSP
JASON MACALLISTER

Red Hat

COALFIRE

North America | Europe
877.224.8077 | info@coalfire.com | Coalfire.com

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Red Hat, Inc. (Red Hat) delivers a comprehensive portfolio of products and services built from open-source software components using an affordable, predictable subscription model. Red Hat engaged Coalfire Systems, Inc. (Coalfire), a respected Payment Card Industry Qualified Security Assessor (QSA) company, to conduct an independent technical assessment of Red Hat® OpenShift Container Platform (OpenShift) on Red Hat Enterprise Linux (RHEL) and/or Red Hat Enterprise Linux CoreOS (RHEL CoreOS). The purpose of this product applicability guide is to identify the alignment of OpenShift and the underlying supported operating systems (OS) to the Payment Card Industry Data Security Standard (PCI DSS) v3.2.1 technical requirements to assist payment entities wishing to use the solution in a manner that supports compliance with PCI DSS v3.2.1.

This product applicability guide may be useful to any payment entity that is considering using OpenShift on RHEL CoreOS as part of their cardholder data environment (CDE). The opinion paper discusses the PCI DSS v3.2.1 requirements that are applicable to OpenShift and RHEL CoreOS. It further identifies and addresses PCI DSS v3.2.1 requirements that may be addressable by the built-in capabilities of OpenShift and RHEL CoreOS. It is understood that the payment entity must address every PCI DSS v3.2.1 requirement; however, for this opinion paper, Coalfire has primarily considered technical controls pertinent to and in alignment with OpenShift and RHEL CoreOS. Coalfire identified and aligned PCI DSS v3.2.1 technical requirements from requirement 1, requirement 2, requirement 6, requirement 7, requirement 8, and requirement 10 for applicability and/or supportability by the reviewed products.

COALFIRE OPINION

Security controls, features, and functionality that are built into OpenShift and RHEL CoreOS can support and address relevant technical PCI DSS v3.2.1 requirements as outlined in the compliance applicability detail section of this paper. It is also Coalfire's opinion that there are sufficient additional Red Hat and third-party solutions available in the market today that satisfy PCI DSS technical requirements where OpenShift and RHEL CoreOS do not alone support or meet the PCI DSS v3.2.1 technical requirements.

In general, the use of software-defined networking through OpenShift's implementation of Open vSwitch along with the OpenShift Ingress Operator and Egress Firewall are best aligned with requirement 1. The use of OpenShift's software-defined networking capabilities may be combined in coordination with the payment entity's overall network security architecture to sufficiently address control requirements. Additionally, the placement of OpenShift nodes on the entity's enterprise network should take into consideration the role that the containers hosted on them may play in the CDE. Strategic network placement and use may better facilitate security, control, and segmentation objectives for the payment entity.

OpenShift and RHEL CoreOS **can support** configuration parameters for requirements specific to the use of vendor default passwords and other default security parameters aligned with PCI DSS requirement 2. The scaled-down RHEL CoreOS helps payment entities align with least function controls, also in support of requirement 2. OpenShift provides mechanisms for application lifecycle management as well as the isolation of test and development environments in support of requirement 6. Access to the OpenShift platform is limited through the API or web console of the OpenShift Control Plane, and role-based access controls (RBAC) provide support for requirement 7 for both OpenShift and the underlying operating system, RHEL CoreOS. Through integration with third-party identity and access control sources, OpenShift provides partial support for requirement 8.

In the compliance applicability detail section of this document, Coalfire suggests consideration for additional non-technical requirements that may be relevant to the use of OpenShift and RHEL CoreOS for PCI DSS v3.2.1 regulated workloads. Also included are considerations for organizations using segmentation to minimize the scope of PCI DSS assessments and to reduce risk to cardholder data (CHD) in the

environment. In the absence of current guidelines for containerization, this opinion is formed based on existing guidance from the Payment Card Industry Security Standards Council (PCI SSC) Special Interest Group (SIGs) for virtualization and cloud computing. The opinion attempts to align with expectations that an assessor may have for addressing a payment entity's infrastructure. This white paper may also be useful to an assessor desiring to better understand the use of OpenShift and RHEL CoreOS in PCI DSS scope. Without specific guidance for the use of containerization technologies in infrastructures supporting a CDE, the payment entity risks that an assessor may find the segmentation controls and PCI DSS requirement controls insufficient.

Finally, no one product, technology, or solution is capable of fully addressing security and compliance requirements. Security is a design principle that must be addressed through carefully planned and implemented strategies. Entities seeking compliance are best able to obtain it through a governance, risk management, and compliance (GRC) program. For this reason, the introduction of new technologies such as OpenShift and RHEL CoreOS should include payment entity-defined security design principles to reduce risk and maintain or improve security. While Coalfire disclaims the generic suitability of any product for regulatory compliance, Coalfire can confirm that, with careful planning and implementation, OpenShift on RHEL CoreOS could be included as part of a payment entity's infrastructure in support of a CDE.

PAYMENT CARD INDUSTRY DATA SECURITY STANDARD 3.2.1

The Payment Card Industry Data Security Standard (PCI DSS) version 3.2.1 is a proprietary information security standard that was developed by the Payment Card Industry Security Standards Council (PCI SSC) to encourage and enhance cardholder data (CHD) security by stipulating a set of requirements regulating the use of information systems that handle CHD. PCI DSS is not an optional standard. As stated, all entities who process, store, or transmit CHD, regardless of geographic location, must comply with the standard, or they can be fined and refused access to the card brand's payment system. To address OpenShift's usability by payment entities in PCI DSS relevant use cases, Coalfire utilized additional guidance provided by the PCI SSC, including the Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation May 2017, Information Supplement: PCI SSC Cloud Computing Guidelines April 2018, Information Supplement: PCI DSS Virtualization Guidelines June 2011, PCI DSS v3.2.1 Template for Report on Compliance, and other similar materials.

UNDERSTANDING PCI DSS SCOPE

A reliable approach for determining where PCI DSS is required to be applied begins with identification and definition of scope for review. Per the PCI DSS Requirements and Security Assessment Procedures document, "PCI DSS security requirements apply to all system components included in or connected to the CDE. The CDE is comprised of people, processes, and technologies that store, process, or transmit cardholder data and sensitive authentication data." (PCI Security Standards Council, 2016). PCI DSS recommends that an assessed entity confirm the accuracy of their PCI DSS scope at least annually or prior to the annual assessment. To help identify scope, the payment entity should evaluate their systems to identify all locations and flows of CHD and identify all systems that are connected to or, if compromised, could impact the CDE. These systems should be included in the scope for review.

For this assessment, the PCI DSS v3.2.1 requirements were limited primarily to technical requirements pertaining to the Red Hat OpenShift Container Platform (OpenShift) on Red Hat Enterprise Linux CoreOS (RHEL CoreOS), including "network devices, servers, computing devices, and applications." (PCI Security Standards Council, 2016). Some additional consideration was made for PCI DSS v3.2.1 operational requirements because these requirements may need to address nuances of the technical platform. The

technical evaluation assumed the use of the OpenShift on RHEL CoreOS for developing, testing, building, managing, monitoring, orchestrating, deploying, and hosting a payment entity's CDE applications. Categorizing the infrastructure and application components helps to identify the role that systems play within the CDE with respect to the CHD. When a system is directly engaged with storing, processing, or transmitting CHD or sensitive authentication data (SAD), it is considered a Tier 1 system. Tier 1 systems may include payment applications, systems that host payment applications, networks that provide transport for CHD or SAD, and so forth. Systems that are adjacent to Tier 1 systems that do not have any direct interactions with CHD or SAD but may provide an avenue to gain access to Tier 1 systems are considered Tier 2 systems. Tier 2 systems, as an example, may include infrastructure monitoring, management, logging, identity, and authentication systems. At a minimum, the OpenShift control plane, when used in this manner, may be considered a Tier 2 system, while OpenShift cluster application nodes may be considered a Tier 1 system when used to host containers that handle CDE. Also included as a Tier 1 system would be any system that provides software-defined networking (SDN) capabilities for the transmission of CHD or SAD, where this data is transmitted through the device providing SDN services.

Network Segmentation

Network segmentation, per PCI DSS v3.2.1, states that isolating (segmenting) the CDE from the remainder of an entity's network is not a PCI DSS requirement; however, it is strongly recommended as a method that may reduce:

- The scope of the PCI DSS assessment
- The cost of the PCI DSS assessment
- The cost and difficulty of implementing and maintaining PCI DSS controls
- The risk to an organization (reduced by consolidating CHD into fewer, more controlled locations) (PCI SSC, 2016)

To evaluate the possibility for the use of segmentation to reduce scope, the payment entity must have a clear understanding of business needs and processes related to the storage, processing, or transmission of CHD. There is an assumption that risk to an organization is reduced through the reduction and consolidation of CHD into fewer and more controlled locations as it pertains to storage, processing, and transmission. This is a fair assumption given that the reduction in scope decreases the surface area for attack and minimizes the number of entry points needed to be controlled.

The payment entity will need to consider the presence and placement of OpenShift internal components, nodes, application pods, and containers within their infrastructure when defining segmentation, establishing DMZs, and isolating secure internal zone assets. Deployment of CDE and non-CDE containers on a single cluster node (host) may impact the payment entity's designed segmentation by making all hosted and connected workloads in scope for assessment, regardless of network segmentation. Each payment entity can benefit from understanding controls that are present within OpenShift on RHEL CoreOS for isolation of workloads to determine the risk associated with commingling CDE and non-CDE. Current guidance from PCI SSC SIGs on cloud and virtualization strongly recommends the separation of workloads representing different zones of trust onto separate hosts.

Wireless Technology

Where used to store, process, or transmit cardholder data, the wireless network and everything attached to it would be considered in scope for application of PCI DSS v3.2.1 requirements and testing procedures. OpenShift is designed to be a part of the enterprise data center infrastructure and would typically be deployed on a wired network.

Use of Third-Party Service Providers/Outsourcing

PCI DSS describes where scope may be impacted using third-party service providers or payment entity outsourcing of services. Specifically, a payment entity that uses a third-party service provider that is used to store, process, or transmits cardholder data on the payment entity's behalf or to manage components that are part of the payment entity's infrastructure, including routers, firewalls, databases, physical security, or servers must consider the impact that the use of the service provider may have on the security of the CDE. The expectation is that there will be a clear understanding of responsibilities from each party for the protection of CHD. While OpenShift supports multi-tenancy and may be used by service providers for delivery of services to payment entities, the applicability of PCI DSS v3.2.1 to OpenShift pertaining to delivery by third-party service or hosting providers was not thoroughly evaluated for this document. However, some of the findings in this document may be useful to third-party service or hosting providers for architecture and design of implementation of OpenShift on RHEL CoreOS implementations in support of PCI DSS regulated workloads.

PCI DSS REQUIREMENTS

The PCI DSS standard is composed of six "control objectives" with twelve "requirements." The following is a listing of control objectives and their associated requirements. These technical and operational requirements were evaluated for applicability with the use of OpenShift on RHEL CoreOS by a payment entity in a PCI DSS regulated environment. The burden for implementing PCI DSS requirements is on the payment entity, regardless of the present capabilities of the technology to provide control.

PCI Data Security Standard – High Level Overview

Build and Maintain a Secure Network and Systems	**1.**	Install and maintain a firewall configuration to protect cardholder data
	2.	Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	**3.**	Protect stored cardholder data
	4.	Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	**5.**	Protect all systems against malware and regularly update anti-virus software or programs
	6.	Develop and maintain secure systems and applications
Implement Strong Access Control Measures	**7.**	Restrict access to cardholder data by business need to know
	8.	Identify and authenticate access to system components
	9.	Restrict physical access to cardholder data
Regularly Monitor and Test Networks	**10.**	Track and monitor all access to network resources and cardholder data
	11.	Regularly test security systems and processes
Maintain an Information Security Policy	**12.**	Maintain a policy that addresses information security for all personnel

Figure 1: PCI DSS Security Standard High-Level Overview (PCI SSC, 2016)

NEW AND EMERGING TECHNOLOGY CHALLENGES TO COMPLIANCE

Many of the new and emerging technologies, such as containerization, software-defined data center, software-defined networking, software-defined storage, and others, provide more efficient, scalable, extensible, and expedient means to support and deliver services for business and their customers. However, often the challenge with the early adoption of new technologies is to understand and properly address the impact that new technology has on security and compliance. PCI SSC SIGs are sometimes formed to provide guidance on the use of new technology; however, new technologies often emerge and evolve faster than regulating bodies are capable of addressing them. For organizations wishing to gain

early benefits of new technology, the level of risk must be evaluated and mitigated, especially as it may impact compliance requirements and the security of protected data.

OPENSHIFT CONTAINER PLATFORM

OpenShift is an enterprise-grade container application platform by Red Hat based on open-source container technologies and the Kubernetes container cluster manager. It is an integrated platform to run, orchestrate, monitor, and scale containers. It is also a multitenant solution that provides build automation, deployment automation, and a rich graphical user experience. Much of the information found in this section can be found on Red Hat's website with the product documentation for OpenShift.

OPENSHIFT CONTAINER PLATFORM AND ROLES

To better understand OpenShift, it is useful to explain the components that make up the OpenShift Container Platform. The following is a listing of components and the roles that the components play, starting with underlying components that support OpenShift.

Operating System – OpenShift can be deployed on either RHEL or Red Hat Enterprise Linux CoreOS (RHEL CoreOS). RHEL CoreOS is a container-optimized, minimal footprint OS powered by Red Hat Enterprise Linux. The OpenShift control plane requires RHEL CoreOS.

Operating Environment – OpenShift can be deployed on bare-metal physical hardware, VMware vSphere, Red Hat Virtualization (RHV), OpenStack, or other major cloud providers. It can be deployed on private or certified public cloud environments, depending on the organization's specific use cases.

Open Container Initiative (OCI) Runtime – Container Runtime Interface-Orchestration (CRI-O) is an OCI-compatible container engine that is installed on every Red Hat CoreOS host. As such, CRI-O enables the use of OCI-compatible containers. OCI has an open governance structure used to create open industry standards around container formats and runtimes. The CRI-O engine focuses on features needed by Kubernetes platforms, such as OCP, and offers specific compatibility with different Kubernetes versions.

Kubernetes – Kubernetes is an open-source container orchestration engine for automating the deployment, scaling, scheduling, and management of containerized applications across the cluster. Kubernetes provides orchestration for complex multi-container services, serving as the de facto standard for orchestrating containers.

Containers – End-user application instances, application components, or other services are run in Linux containers. A container should only include the necessary libraries, functions, elements, and code required to run the application.

Pods – While application components run in containers, OCP orchestrates and manages pods. A Kubernetes pod is a group of containers that are deployed together on the same host. A pod is an orchestrated unit in OCP made up of one or more containers. A pod should only contain a single function, such as an application server or web server, and should not include multiple functions such as both a database and application server.

Operators – An Operator is a method of packaging, deploying, and managing a Kubernetes-native application. Operators automate the lifecycle management of containerized applications within Kubernetes. A Kubernetes-native application is an application that is both deployed on Kubernetes and managed using the Kubernetes APIs and kubectl tooling. A controller is a core concept of Kubernetes. It is implemented as a software loop that runs continuously, compares, and, if necessary, reconciles the expressed desired state and the current state of an object. Objects are resources like Pods, Services, ConfigMaps, or PersistentVolumes, but, through Custom Resource Definitions (CRDs), could be any object. Operators

apply the model of the controller at the level of entire applications and are, in effect, application-specific custom controllers.

OpenShift Nodes – Nodes are instances of RHEL CoreOS with the OpenShift software installed. Nodes are where end-user applications are run in containers. Nodes will contain the necessary OpenShift node daemon, the container runtime, and other necessary services to support the hosting of containers.

OpenShift Control Plane – The Control Plane maintains and understands the state of the environment and orchestrates all activity that occurs on the nodes. For the purposes of automation and reliability, the Control Plane only runs on RHEL CoreOS.

OCP adds developer-centric and operations-centric tools to Kubernetes that help enable rapid application development, simplified platform deployment, scaling, and long-term lifecycle maintenance. OCP also leverages integrated components to automate application builds, deployments, scaling, and health and lifecycle management. Included in the automation capabilities of OCP is the ability to configure and deploy Kubernetes container host clusters.

OPENSHIFT CONTAINER PLATFORM SECURITY

OpenShift runs on RHEL CoreOS and makes use of existing security features built into the operating system. Red Hat is committed to responsive action to security vulnerabilities. Red Hat proactively manages OpenShift, including the underlying OS. The security of OpenShift includes and utilizes hardened technologies such as SELinux; process, network, and storage separation with kernel and Kubernetes Namespaces; proactive monitoring of capacity limits (CPU, disk, memory, etc.) with cgroups; and encrypted communications for infrastructure transport, including SSH, TLSv1.2, etc. Additionally, OpenShift provides secure authentication and authorization options to support organization compliance requirements.

For initial implementation and continuing maintenance, OpenShift components, including RHEL CoreOS, can be acquired from Red Hat-controlled distribution points. Red Hat provides enterprise support for OpenShift Container Platform supported versions, including vulnerability patches and updates.

SCOPE AND APPROACH FOR REVIEW

The understanding of OpenShift on RHEL CoreOS and their combined capabilities was gained through product specification, installation, configuration, administration, and integration documentation provided by Red Hat and generally made available from Red Hat's public-facing website. Coalfire further conducted interviews and engaged in live product demonstrations with Red Hat personnel. For live product demonstration purposes, OpenShift was also implemented on RHEL CoreOS in the Coalfire lab environment to provide hands-on testing and analysis of the system's capabilities to support compliance.

Coalfire's review of OpenShift on RHEL CoreOS began with a general alignment of the applicability of the technology against the high-level PCI DSS control objectives. This was further narrowed down to specific requirements that were considered applicable to either OpenShift or RHEL CoreOS. An analysis of capability for the reviewed technology to address the applicable requirements was then conducted. This analysis primarily focused on what a PCI DSS QSA might review when following the PCI DSS testing procedures during an assessment of applicable requirements.

SCOPE OF TECHNOLOGY AND STANDARD TO REVIEW

Coalfire was tasked by Red Hat to review OpenShift 4 as deployed on RHEL CoreOS. The primary focus of review included the components, features, and functionality of OpenShift along with the supporting underlying OS features and functionality when the OpenShift components are deployed on RHEL CoreOS. Coalfire did not include in the assessment application pods or containers that a payment entity may deploy

on OpenShift. Containers that were deployed in the lab environment were used for the purposes of demonstrating the platform's orchestration, deployment, and management capabilities. Furthermore, Coalfire did not assess publicly available image registries or repositories that may be used for acquiring applications, services, dependencies, or other elements to be hosted on or used for the setup of OpenShift.

For this review, Coalfire included requirements from PCI DSS Requirements and Security Assessment Procedures Version 3.2.1 May 2018 publication available from https://www.pcisecuritystandards.org. For a broader understanding of the requirements and their applicability to technical solution implementation, Coalfire also reviewed supporting documentation provided by the PCI SSC, including the approach to assessment guidance, a sample report on compliance form, and self-assessment questionnaires. Applied understanding of PCI DSS requirements and recommendations was also made available from PCI DSS guidance documents on relevant topics such as scoping and segmentation, best practices for maintaining PCI DSS compliance, risk assessment guidance, cloud computing, and virtualization.

COALFIRE EVALUATION METHODOLOGY

Coalfire initially examined the PCI DSS requirements and identified them as either procedural or technical. Qualification of a requirement as procedural or technical was based on a review of the requirement narrative, testing procedures, and guidance.

"Non-technical" operational requirements that include definition and documentation of policies, procedures, and standards were not considered applicable to the technical solution. Likewise, "non-technical" requirements, including operational procedures that describe manual processes, were not assessed against the technology. Examples of this type of "non-technical" requirement included maintenance of facility visitor logs, verification of an individual's identity prior to granting physical or logical access, the performance of periodic physical asset inventories, or generation of network topology or flow diagrams.

Technical requirements were then assessed to determine applicability to the solution and/or solution components. Where the achievement of the requirement objectives was more likely to be met using an external and non-adjacent mechanism, the requirement was determined to be "not applicable" to the assessed technology. Examples of requirements that Coalfire determined not to be applicable to OpenShift or RHEL CoreOS included wireless networking, technical, and physical access controls.

Where the requirement was qualified as applicable, Coalfire further assessed the capability of the solution to address the requirement. For applicable requirements, Coalfire designated a qualitative category of capability, including whether the solution was determined to fully support the requirement, partially support the requirement, or was unable to support the requirement. In cases where the requirement was determined to be applicable but unsupported, additional thought for the use of third-party solutions could be considered.

Each applicable requirement is described in the table in the following section. This table includes the findings of applicability along with a short narrative describing the capability. The table uses Harvey Balls to describe the qualitative applicability of the technology to the requirement.
(see: https://en.wikipedia.org/wiki/Harvey_Balls)

The following key describes the meaning of the Harvey Balls.

DESIGNATION	KEY	MEANING
Fully Supported	●	The evaluated technology has the means, through configuration or by design, to fully support the requirement.
Partially Supported	◐	The evaluated technology has the means, through configuration or by design, to partially support the requirement. Additional technology or organizational policy, procedures, or standards may be required to fully meet the requirement objective.
Not Supported	○	The evaluated technology does not have the means, through configuration or by design, to support or meet the requirement objective. The technology relies on an external technology, system, or organizational procedure to meet the requirement objective.
Does Not Apply	N/A	The requirement, though it may be technical in nature, is not applicable to the evaluated technology. Meeting the requirement is completely achieved independently of the evaluated technology.
Non-Tech	Non-Tech	The requirement is purely operational, requiring a defined and documented policy, procedure, or standard. This may also include operational, non-technical procedures such as the implementation of training, maintaining of paper logs, or performance of personnel identity verification.

Table 1: Harvey Ball Applicability Key

OPENSHIFT APPLICABILITY TO PCI DSS V3.2.1

Regarding requirement 1, “install and maintain a firewall configuration to protect cardholder data,” it is expected that the payment entity will implement firewalls and routers at the edge of the network to protect the trusted network from untrusted external networks. The payment entity may consider the use of additional routers, firewalls, or layer 3 switching to provide additional segmentation or isolation of networks within the trusted realm of the entity’s network. Changes in router configurations would include changes to NetworkPolicy objects, as well as modifications to OpenShift ingress and egress routing methods to support OpenShift workloads. For assessment purposes, the payment entity should also be able to demonstrate that the network controls are sufficient to prevent unauthorized access from non-CDE systems to CDE systems. The design of the network should take into consideration the placement and use of OpenShift nodes. This includes the placement of cluster nodes with respect to the intended hosted pods or containers.

Pods or containers may represent elements of services that typically exist within the DMZ, such as a web service, portal, or website. Other pods or containers may represent elements that traditionally exist on internal networks, such as applications or databases. The placement and use of these nodes will be important regarding ensuring proper boundary protection between the untrusted networks, the DMZ, and trusted networks, including separation of CDE and non-CDE where microsegmentation is applied for scope reduction. It is recommended to place the OpenShift environment on the internal network protected by the payment entity’s firewalls at each internet connection and between any DMZ and the internal network zone. The expectation is that the choke router at the internet, the DMZ router, and firewall, or the perimeter firewalls and routers, will provide a layer of protection between the internet and the internal systems, including the OpenShift environment.

Additionally, NetworkPolicy rules allow for the restriction of traffic into applications. This allows for microsegmentation of workloads within the OpenShift environment to control network traffic between pods.

To maintain brevity, requirements that were designated as not applicable or non-technical were removed from the following detail in the requirements table.

PCI DSS 3.2.1 COMPLIANCE APPLICABILITY DETAIL

ID	REQUIREMENT	SUPPORT	NARRATIVE
1.2	Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment. Note: An “untrusted network” is any network that is external to the networks belonging to the entity under review and/or which is out of the entity’s ability to control or manage.	◐	The OpenShift router (HAProxy) supports ingress and routes to provide external access to services running on-cluster. The OpenShift router supports re-encrypt and passthrough policies: “re-encrypt” decrypts and re-encrypts HTTP traffic when forwarding it, whereas “passthrough” passes traffic through without terminating TLS. A cluster administrator can use an egress firewall to limit the external hosts that some or all pods can access from within the cluster. An egress firewall supports the following scenarios: - A pod can only connect to internal hosts and cannot initiate connections to the public internet. - A pod can only connect to the public internet and cannot initiate connections to internal hosts that are outside the OpenShift Container Platform cluster. - A pod cannot reach specified internal subnets or hosts outside the OpenShift Container Platform cluster. - A pod can connect to only specific external hosts.
1.2.1	Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment, and specifically deny all other traffic.	●	Restrictions for only necessary ingress/egress traffic and explicit deny all other traffic is made possible within OpenShift with the use of ingress configuration, network policy objects, and egress policy (egress firewall) rules.
1.2.2	Secure and synchronize router configuration files.	◐	Ingress into the OpenShift environment is handled via the use of routes or Ingress objects. The Ingress Operator’s controller function ensures that ingress configurations are synchronized throughout the cluster. OpenShift’s RBAC can be used to define which users or groups may make changes to configurations.
1.3	Prohibit direct public access between the internet and any system component in the cardholder data environment.	◐	Where a container may represent the DMZ element of a project, it will need to reside on a cluster node on the DMZ. While other project container may represent a secure internal network element of a service, where these containers will need to be placed on nodes residing on the internal network. Additionally, the payment entity can consider the integration of OpenShift Container Platform with proxy solutions to enhance the security and ensure adequate control for outbound connections to the internet.

ID	REQUIREMENT	SUPPORT	NARRATIVE
1.3.1	Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.	◐	Use dedicated router nodes in combination with Kubernetes Network Policies to create access zones within the cluster. Deploy and configure a router instance to manage DMZ traffic. Use Network Policies to create logical zones in the SDN.
1.3.2	Limit inbound internet traffic to IP addresses within the DMZ.	◐	Where a container may represent a DMZ element of a project or service, it will need to reside on a cluster node on the DMZ. The cluster node in the DMZ will contain an OpenShift router for ingress with routes that direct traffic to the DMZ elements served from the OpenShift cluster. Additionally, for inbound traffic with a load balancer in front of the routers, the load balancer virtual IP (VIP) must belong to the network zone designated by the customer for this requirement.
1.3.4	Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.	●	The OpenShift SDN supports the creation of an egress firewall that can prevent unauthorized outbound traffic from the CDE. https://docs.openshift.com/container-platform/4.7/security/container_security/security-network.html https://docs.openshift.com/container-platform/4.7/networking/openshift_sdn/configuring-egress-firewall.html
1.3.6	Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.	●	The OpenShift SDN supports the limitation of network access between pods via NetworkPolicy objects. . The default cluster configuration does not include any NetworkPolicy objects, and therefore no isolation is configured by default. A cluster admin can configure NetworkPolicy rules to appropriately segment traffic within the cluster. NetworkPolicy is used to control access between pods regardless of the hosting Node. Customers can configure zones and use network policies to segregate (microsegment) the CDE. https://www.openshift.com/blog/openshift-and-network-security-zones-coexistence-approaches https://docs.openshift.com/container-platform/4.7/networking/network_policy/creating-network-policy.html https://docs.openshift.com/container-platform/4.7/networking/network_policy/multitenant-network-policy.html

ID	REQUIREMENT	SUPPORT	NARRATIVE
1.3.7	Do not disclose private IP addresses and routing information to unauthorized parties. Note: Methods to obscure IP addressing may include, but are not limited to: Network Address Translation (NAT), placing servers containing cardholder data behind proxy servers/firewalls, removal or filtering of route advertisements for private networks that employ registered addressing, and internal use of RFC1918 address space instead of registered addresses.	●	By default, there is no public access to the pods directly from the internet. Routers (Ingress Controller) will use ports 80/443 on all public interfaces on the node where it runs. Routers will specify whether the routes may listen for all exposed services or only respond to requests for a subset of hostnames. Additionally, the ingress configuration can specify which services and/or hostnames can be serviced.
2.1	Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network. This applies to ALL default passwords, including but not limited to those used by operating systems, software that provides security services, application and system accounts, point-of-sale (POS) terminals, payment applications, Simple Network Management Protocol (SNMP) community strings, etc.).	◐	OpenShift and RHEL CoreOS do not come preconfigured with a default password. At the conclusion of the installation, a randomly generated password is provided for initial administrative access. Administrators are expected to integrate OpenShift with an Identity Provider (IDP) and subsequently disable the generated administrative account. SSH access is enabled on RHEL CoreOS, but it requires key-based authentication. Root access via SSH is disabled, and the SSH key for access must be either provided at the time of the installation or added via OpenShift configuration later.
2.2	Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards. Sources of industry-accepted system hardening standards may include, but are not limited to: • Center for Internet Security (CIS) • International Organization for Standardization (ISO) • SysAdmin Audit Network Security (SANS) Institute • National Institute of Standards Technology (NIST).	◐	See sub-requirements below.
2.2.1	Implement only one primary function per server to prevent functions that require different security levels from co-existing on the same server. (For example, web servers, database servers, and DNS should be implemented on separate servers.) Note: Where virtualization technologies are in use, implement only one primary function per virtual system component.	●	OpenShift provides the means to separate application workloads within the OpenShift cluster onto separate hosts within the cluster. This allows the payment entity to separate CDE from non-CDE pods and to target DMZ and internal pods to separate hosts.

ID	REQUIREMENT	SUPPORT	NARRATIVE
2.2.2	Enable only necessary services, protocols, daemons, etc., as required for the function of the system.	●	When an OpenShift Container Platform cluster is installed, the installation program is downloaded from the appropriate Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site. The installation only installs the components required to run the cluster, including RHEL CoreOS. RHEL CoreOS is purpose-built to support container-based workloads and comes preconfigured with only the necessary elements required to run containers. For more information, see https://docs.openshift.com/container-platform/4.7/installing/index.html The optional Compliance Operator can be used to audit an OCP cluster for the configuration of technical controls. For more information, see https://docs.openshift.com/container-platform/4.7/security/compliance_operator/compliance-operator-understanding.html
2.2.3	Implement additional security features for any required services, protocols, or daemons that are insecure.	●	Communication at the Management and Control planes is performed using TLS 1.2 encryption. Additionally, OpenShift Service Mesh supports mutual tls (mTLS)-enforcement capabilities that can be used to transparently upgrade inter-pod traffic. As of OpenShift 4.7, OVN-IPsec allows for encrypting all node-to-node traffic with IPsec when using Open Virtual Network (OVN).
2.2.4	Configure system security parameters to prevent misuse.	◐	OCP and RHEL CoreOS can be configured and hardened per best practices. Also, see a mention about OVN-IPsec in 2.2.3. The Compliance Operator can be used to audit an OCP cluster for the configuration of technical controls. The Compliance Operator uses OpenSCAP, a NIST-certified tool, to scan and enforce security policies provided by the compliance profiles. For more information, see https://docs.openshift.com/container-platform/4.7/security/compliance_operator/compliance-operator-understanding.html Additional configuration of routers (ingress controllers) and egress controls is also available. More information about OCP 4 security capabilities can be found at https://docs.openshift.com/container-platform/4.7/welcome/index.html, and selecting the Security and Compliance links on the left side of the page.

ID	REQUIREMENT	SUPPORT	NARRATIVE
2.2.5	Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems, and unnecessary web servers.	◐	RHEL CoreOS comes pre-hardened with the minimum necessary functionality to support running containers. All additional functionality available from OpenShift as layered on RHEL CoreOS is necessary for the function of the platform and is described in detail in the product documentation.
2.3	Encrypt all non-console administrative access using strong cryptography.	●	The only way to access the OpenShift Control Plane or other OpenShift cluster hosts for non-console administrative access is through SSH, which is enabled by default in RHEL CoreOS. Telnet is not enabled. Administrative access to the API or the Web User Interface occurs over HTTPS using TLS 1.2. Note: In cloud environments, nodes do not have public IPs by default. They also do not have SSH keys installed by default if a payment entity does not provide SSH keys, making SSH access unavailable.
2.6	Shared hosting providers must protect each entity's hosted environment and cardholder data. These providers must meet specific requirements as detailed in Appendix A1: Additional PCI DSS Requirements for Shared Hosting Providers.	◐	Accommodations are in place with OpenShift to support the use of the platform by shared hosting providers. This allows the shared hosting provider to protect each entity's hosted environment and cardholder data. There are multiple considerations for separating tenants in a multitenant environment, including separating tenants to their own OpenShift projects, the use of multitenant isolation with network policy, and the use of a multi-segmented network. Separate clusters can also be used for additional segmentation.

ID	REQUIREMENT	SUPPORT	NARRATIVE
4.1	Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks, including the following: Only trusted keys and certificates are accepted. The protocol in use only supports secure versions or configurations. The encryption strength is appropriate for the encryption methodology in use. Examples of open, public networks include but are not limited to: • The internet • Wireless technologies, including 802.11 and Bluetooth • Cellular technologies, for example, Global System for Mobile communications (GSM), Code division multiple access (CDMA) • General Packet Radio Service (GPRS) • Satellite communications	●	TLS 1.2 is supported in the HTTPS TLS configurations by configuring the Ingress Operator (HAProxy) with a tlsSecurityProfile of intermediate or modern. https://access.redhat.com/documentation/en-us/openshift_container_platform/4.7/html/networking/configuring-ingress#nw-ingress-controller-tls-profiles_configuring-ingress
5.1	Deploy antivirus software on all systems commonly affected by malicious software (particularly personal computers and servers).	◐	The user space in RHEL CoreOS is read-only. Use the optional File Integrity Operator to monitor the small set of writable directories for unexpected changes. For more information, see https://docs.openshift.com/container-platform/4.7/security/file_integrity_operator/file-integrity-operator-understanding.html Malicious software or vulnerabilities may be present in an image being deployed by a pod. OCP supports scanning for vulnerabilities in images with Red Hat Quay or Red Hat Advanced Cluster Security. The optional Container Security Operator (CSO) can be deployed to provide a view in the OpenShift console of known vulnerabilities in images deployed to the cluster from Quay. https://access.redhat.com/documentation/en-us/openshift_container_platform/4.7/html/security_and_compliance/pod-vulnerability-scan There are also a variety of vulnerability scanning solutions available from Red Hat certified partners. See sub-requirements below.

ID	REQUIREMENT	SUPPORT	NARRATIVE
5.1.1	Ensure that antivirus programs are capable of detecting, removing, and protecting against all known types of malicious software.	◐	Traditional anti-malware solutions, vulnerability scanning techniques, intrusion detection, and audit logging processes, for instance, may not be as effective in container runtime environments. The nature of how containers are executed on a container platform can limit the capabilities of traditional security tools to detect malware and identify vulnerabilities. However, with a proper understanding of container environments and the container lifecycle, new methods of applying security techniques to achieve the same or better outcome can be implemented. As an example, in most cases, containers that are deployed and executed in the container runtime are immutable. Containers are deployed from images which are stored in repositories or a registry waiting to be deployed and executed. With this knowledge, it seems reasonable to target the scanning of vulnerabilities and malware at the image repository. In addition, as part of the systems development lifecycle, entities can establish a gate to inspect and digitally sign images prior to their placement in the image repository.
5.2	Ensure that all antivirus mechanisms are maintained as follows: • Are kept current, • Perform periodic scans • Generate audit logs that are retained per PCI DSS Requirement 10.7.	◐	Malicious software or vulnerabilities may be present in an Image being deployed by a Pod. OCP supports scanning for vulnerabilities in images with Red Hat Quay or Red Hat Advanced Cluster Security. The optional Container Security Operator (CSO) can be deployed to provide a view in the OpenShift console of known vulnerabilities in images deployed to the cluster from Quay. The vulnerabilities in images will be reported by their maintainers; trusted images are maintained. Use the optional File Integrity Operator to schedule file integrity checks on the cluster nodes. The File Integrity operator deploys a daemon set that initializes and runs privileged advanced intrusion detection environment (AIDE) containers on each node, providing a status object with a log of files that are modified after the initial run. Red Hat Universal Base Images (UBI) are updated regularly to keep up with security patches. See the narrative in 5.1.1 for a complete description of the antivirus requirement.

ID	REQUIREMENT	SUPPORT	NARRATIVE
5.3	Ensure that antivirus mechanisms are actively running and cannot be disabled or altered by users unless specifically authorized by management on a case-by-case basis for a limited time period. Note: Anti-virus solutions may be temporarily disabled only if there is a legitimate technical need, as authorized by management on a case-by-case basis. If antivirus protection needs to be disabled for a specific purpose, it must be formally authorized. Additional security measures may also need to be implemented for the period during which antivirus protection is not active.	◐	Malicious software or vulnerabilities may be present in an Image being deployed to a Pod. OCP supports scanning for vulnerabilities in images with Red Hat Quay or Red Hat Advanced Cluster Security. The optional Container Security Operator (CSO) can be deployed to provide a view in the OpenShift console of known vulnerabilities in images deployed to the cluster from Quay. The vulnerabilities in images will be reported by their maintainers. Trusted images are maintained by Red Hat, and vulnerabilities should be quickly disclosed. Also, see the narrative in 5.1.1 and 5.2 for a complete description of the antivirus requirement, file integrity operator, and regular security patches.
6.4	Follow change control processes and procedures for all changes to system components. The processes must include the following:	◐	See sub-requirements below.
6.4.1	Separate development/test environments from production environments and enforce the separation with access controls.	●	Like tenant separation, OpenShift can be implemented in a way to allow for separation of development/test environments from production environments, including the use of RBACs, projects, and NetworkPolicy rules to enforce separation. Access controls can be applied to projects where separate projects can be created for each development and test environment. Alternatively, separate clusters can be deployed and managed for development/test and production.
6.4.2	Separation of duties between development/test and production environments	●	Access controls for the administration and management of OpenShift can be established to support the separation of duties; different levels of access can be granted to development and test projects and users as opposed to production projects and users. Alternatively, separate clusters can be deployed and managed for development/test and production.

ID	REQUIREMENT	SUPPORT	NARRATIVE
7.2	Establish an access control system(s) for systems components that restricts access based on a user's need to know and is set to "deny all" unless specifically allowed. This access control system(s) must include the following:	●	OpenShift on RHEL CoreOS can be configured to support the payment entity's standards for access control based on minimum access requirements. OpenShift includes built-in authentication and authorization controls. For users to interact with OpenShift Container Platform, they must first authenticate to the cluster. The authentication layer identifies the user associated with requests to the OpenShift Container Platform API. The authorization layer then uses information about the requesting user to determine if the request is allowed. As an administrator, you can configure authentication for OpenShift Container Platform. RBACs can be implemented to establish authorization with a granular level of control for interaction with and/or administration of OpenShift and the underlying OS. No initial access is granted, except for access initiated during setup. Although anonymous access cannot be disabled without impacting features (API/webhook integration, discovery, etc.), it is recommended that quick setup or a custom setup be performed, and anonymous access not be utilized. The narrative in ID 2.1 provides information for identity providers and the kubeadmin user to provide access controls. Though anonymous access does not authorize an individual to perform any actions, it is recommended to use a strict identity and access implementation. See sub-requirements below.
7.2.1	Coverage of all system components.	●	OpenShift authentication and authorization components support coverage of all OCP systems components. Control for the application or service running in the container is not covered by OpenShift and would be the responsibility of the payment entity or application developer to implement.
7.2.2	Assignment of privileges to individuals based on job classification and function.	●	Roles can be established that are in alignment with the payment entity's standards and definition for job classifications and functions.
7.2.3	Default "deny-all" setting.	●	The solution can be implemented to deny-all access except for that which is explicitly permitted during the initial install or configuration of the Identity Provider.

ID	REQUIREMENT	SUPPORT	NARRATIVE
8.1	Define and implement policies and procedures to ensure proper user identification management for non-consumer users and administrators on all system components as follows:	◐	A user in OpenShift Container Platform is an entity that can make requests to the OpenShift Container Platform API. An OpenShift Container Platform User object represents an actor which can be granted permissions in the system by adding roles to them or to their groups. Typically, this represents the account of a developer or administrator that is interacting with OpenShift Container Platform. Each user must authenticate in some way to access OpenShift Container Platform. Once authenticated, policy determines what the user is authorized to do. Administrators can configure authentication to specify an identity provider after the cluster is installed. See sub-requirements below.
8.1.4	Remove/disable inactive user accounts within 90 days.	◐	Revocation of access for terminated users would be performed with the identity provider. Users with revoked access would not be able to access OpenShift. Likewise, removal or disabling of inactive user accounts within 90 days would be handled with the identity provider. All user IDs, including those handled by third parties to access, support, or maintain system components via remote access, would be handled externally to OpenShift. The options for controlling remote access for third parties are the responsibility of the payment entity.
8.1.6	Limit repeated access attempts by locking out the user ID after no more than six attempts.	◐	Account lockout for failed attempts would be managed by the identity provider, as with all authentication attempts that occur prior to granting access from OpenShift. Establishing a threshold for limiting repeated failed attempts would be configured with the chosen identity provider. Likewise, the lockout duration for the account and mechanisms to unlock the account for use would be established with the identity provider.

ID	REQUIREMENT	SUPPORT	NARRATIVE
8.1.7	Set the lockout duration to a minimum of 30 minutes or until an administrator enables the user ID.	◐	Token timeouts can be enabled via integration options with the token identity provider to limit the amount of time that a token can be active. https://docs.openshift.com/container-platform/4.7/authentication/configuring-oauth-clients.html#oauth-token-inactivity-timeout_configuring-oauth-clients It is, however, recommended that the payment entity control idle session timeouts at the user or administrator endpoint by implementing controls on the local Operating System rather than at the OpenShift console.
8.1.8	If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session.	◐	PCI QSA assessment typically confirms that this session limit for administrative users is met by setting idle limits for authentication to the terminal or client (user) device via user endpoint policies or local settings. This prevents someone from walking up to an unoccupied terminal and hijacking the user's open session. Should additional enforcement be desired, OpenShift has an inactivity timeout. The default lifetime of an access token is 24 hours. While the lifetime of an access token can be reduced through configuration to fifteen (15) minutes, doing so may prove exceedingly burdensome to the user of the system as this would require reauthentication upon token expiration regardless of session idle state. https://docs.openshift.com/container-platform/4.7/authentication/configuring-oauth-clients.html#oauth-token-inactivity-timeout_configuring-oauth-clients
8.2	In addition to assigning a unique ID, ensure proper user-authentication management for non-consumer users and administrators on all system components by employing at least one of the following methods to authenticate all users: Something you know, such as a password or passphrase; something you have, such as a token device or smart card; or something you are, such as a biometric.	◐	The type of authenticators to be used (e.g., password or passphrase, token device or smart card, or biometrics) are also managed externally to OpenShift by the identity provider. The protection of the authentication credentials, such as rendering the passwords and passphrases unreadable during transmission and the storage of credentials on system components, is the responsibility of the third-party identity provider. Likewise, modification of authentication credentials is handled by the third-party identity provider. All access to modify parameters for authentication tokens or for generating keys within OpenShift is managed with RBAC and requires prior authentication before the user is authorized to act.

ID	REQUIREMENT	SUPPORT	NARRATIVE
8.2.1	Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.	◐	Along with the third-party identity and authentication provider, OpenShift and RHEL CoreOS provide the means to support encryption of the authentication process to the system component.
8.2.3	Passwords/passphrases must meet the following: Require a minimum length of at least seven characters. Contain both numeric and alphabetic characters. Alternatively, the passwords/ passphrases must have complexity and strength at least equivalent to the parameters specified above.	○	Parameters for authenticators such as password length, maximum password age, minimum password age, password history, and requirements to change the password on first use are also handled by the third-party identity provider.
8.2.4	Change user passwords/passphrases at least once every 90 days.	○	Recommend using a third-party, external identity provider for identity and authentication management.
8.2.5	Do not allow an individual to submit a new password/passphrase that is the same as any of the last four passwords/passphrases he or she has used.	○	Recommend using a third-party, external identity provider for identity and authentication management.
8.2.6	Set passwords/passphrases for first-time use and upon reset to a unique value for each user and change immediately after the first use.	○	Recommend using a third-party, external identity provider for identity and authentication management.
8.3	Secure all individual non-console administrative access and all remote access to the CDE using multi-factor authentication. Note: Multi-factor authentication requires that a minimum of two of the three authentication methods (see requirement 8.2 for descriptions of authentication methods) be used for authentication. Using one factor twice (for example, using two separate passwords) is not considered multi-factor authentication.	○	Recommend using a third-party, external identity provider for identity and authentication management. Select an identity provider that supports multi-factor authentication. See sub-requirements below.
8.3.1	Incorporate multi-factor authentication for all non-console access into the CDE for personnel with administrative access. Note: This requirement is a best practice until January 31, 2018, after which it becomes a requirement.	○	Recommend using a third-party, external identity provider for identity and authentication management. Select an identity provider that supports multi-factor authentication.
8.3.2	Incorporate multi-factor authentication for all remote network access (both user and administrator and including third-party access for support or maintenance) originating from outside the entity's network.	○	Recommend using a third-party, external identity provider for identity and authentication management. Select an identity provider that supports multi-factor authentication.

ID	REQUIREMENT	SUPPORT	NARRATIVE
8.5	Do not use group, shared, or generic IDs, passwords, or other authentication methods as follows: Generic user IDs are disabled or removed. Shared user IDs do not exist for system administration and other critical functions. Shared and generic user IDs are not used to administer any system components.	●	Any user with cluster admin privileges has privileged access to the nodes and should be protected in the same manner as root. Customers can configure multiple administrative users but must be cautious of how much privilege is provided and when. It is important to consider how quickly authorization changes are to take effect when group membership changes via LDAP synchronization. OpenShift will not know about changes to LDAP group membership until synchronization is run. Customers must also ensure to synchronization from LDAP is performed regularly to ensure limited access.
8.6	Where other authentication mechanisms are used (for example, physical or logical security tokens, smart cards, certificates, etc.), use of these mechanisms must be assigned as follows: • Authentication mechanisms must be assigned to an individual account and not shared among multiple accounts. • Physical and/or logical controls must be in place to ensure only the intended account can use that mechanism to gain access.	●	OpenShift provides integrated management of X.509 certificates for internal cluster components. Containerized applications are responsible for managing their own certificates signed by organizational CAs or may make use of the OpenShift CA. The OpenShift CAs are managed by the cluster and are only used within the cluster. Additionally, application development teams are responsible for managing authentication for workloads. Applications deployed into OpenShift do not have their certificates managed by default. Application developers can provide a certificate for their application manually by creating a secret object that contains the key and certificate files and then mounting that secret in their deployment.
10.1	Implement audit trails to link all access to system components to each individual user.	●	In OCP, auditing occurs for both the RHEL CoreOS layer and the OpenShift API events. All-access to the API or web console for OpenShift is logged, including the HTTP method being invoked.
10.2	Implement automated audit trails for all system components to reconstruct the following events:	◐	See sub-requirements below.
10.2.1	All individual user accesses to cardholder data.	●	In OCP, auditing occurs for both the RHEL CoreOS layer and the OpenShift API events. All-access to the API or web console for OpenShift is logged, including the HTTP method being invoked. Typically, CHD would be contained and isolated within the operations of the pod and would not likely be accessible through the OpenShift API or web interface. Where persistent storage is used to support extended storage of CHD, access to the storage should be logged as well. OpenShift does not directly provide access to persistent storage.

ID	REQUIREMENT	SUPPORT	NARRATIVE
10.2.2	All actions taken by any individual with root or administrative privileges.	●	In OCP, auditing occurs for both the RHEL CoreOS layer and the OpenShift API events. All-access to the API or web console for OpenShift is logged, including the HTTP method being invoked.
10.2.3	Access to all audit trails.	◐	Access to audit logs is RBAC controlled. For more information, see https://docs.openshift.com/container-platform/4.7/security/audit-log-view.html and For better protection of audit trails, it is recommended to configure the system to forward logs to a qualified central syslog server or SIEM solution. For more information, see https://docs.openshift.com/container-platform/4.7/logging/cluster-logging-external.html
10.2.4	Invalid logical access attempts.	◐	Invalid logical access attempts, as they pertain to incorrect input of credentials for access to OpenShift, would most likely be logged by the identity and authentication provider external to OpenShift. Unauthorized attempts to access system components, run unauthorized commands, or execute unauthorized privileged commands would be logged by RHEL CoreOS.
10.2.5	Use of and changes to identification and authentication mechanisms—including but not limited to the creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges.	◐	For OpenShift, this will be primarily handled by the third-party external identity and authentication system that is integrated OpenShift.
10.2.6	Initialization, stopping, or pausing of the audit logs.	●	Initializing, stopping, or pausing of audit logs generated by OpenShift are handled by the Control Plane. This requires modifying an object (config) which will also be logged. For RHEL CoreOS, the auditd configuration file (/etc/audit/auditd.conf) can be modified as necessary to meet requirements such as retention and fault tolerance. https://docs.openshift.com/container-platform/4.7/security/audit-log-policy-config.html
10.2.7	Creation and deletion of system-level objects.	●	Creation and deletion of system-level objects is logged by OpenShift (for OpenShift objects) and by RHEL or RHEL CoreOS for OS objects.
10.3	Record at least the following audit trail entries for all system components for each event:	●	See sub-requirements below.

ID	REQUIREMENT	SUPPORT	NARRATIVE
10.3.1	User identification	●	User identification is present as a logged artifact.
10.3.2	Type of event	●	The type of event is present as a logged artifact.
10.3.3	Date and time	●	Date and time are present as a logged artifact.
10.3.4	Success or failure indication	●	Success or failure of the action taken is a logged artifact.
10.3.5	Origination of event	●	The origination or source of the event is a logged artifact.
10.3.6	Identity or name of affected data, system component, or resource.	●	The name of the affected data, system component, or resource is a logged artifact.
10.4	Using time-synchronization technology, synchronize all critical system clocks and times and ensure that the following is implemented for acquiring, distributing, and storing time. Note: One example of time synchronization technology is Network Time Protocol (NTP).	●	OpenShift is dependent on time synchronization to support sensitive operations, such as log keeping and time stamps. OpenShift uses the time of the underlying host. By default, OpenShift uses NTP to synchronize all Control Plane and Worker Nodes. This is performed via the chrony package. The Machine Config operator manages configuration. Additional information is available in the documentation. OpenShift operations including etcd, leader election and health checks for pods, and can help prevent time skew problems. https://docs.openshift.com/container-platform/4.7/installing/install_config/installing-customizing.html#installation-special-config-chrony_installing-customizing
10.4.1	Critical systems have the correct and consistent time.	●	The use of time synchronization with the industry accepted NTP server as a centralized and trusted time source helps to ensure that the systems have the correct and consistent time. The payment entity may want to use the optional File Integrity operator to monitor for changes to chrony.conf. https://docs.openshift.com/container-platform/4.7/security/file_integrity_operator/file-integrity-operator-understanding.html

ID	REQUIREMENT	SUPPORT	NARRATIVE
10.4.2	Time data is protected.	●	Timekeeping is configured at the OS layer by the Machine Config Operator. Cluster admin access, as well as access to the OS, can be limited, controlled, and logged to minimize risk and the possibility of compromise to time data. The payment entity may want to use the optional File Integrity Operator to monitor for changes to chrony.conf. https://docs.openshift.com/container-platform/4.7/security/file_integrity_operator/file-integrity-operator-understanding.html
10.4.3	Time settings are received from industry-accepted time sources.	●	NTP can be configured to use an industry-accepted time source. Where the system is unable to directly access a time source from the internet, the payment entity can implement an internal NTP server that can communicate with the external source. Internal protected devices can be configured to use the internal NTP server as a time source.
10.5	Secure audit trails so they cannot be altered.	●	Securing of audit trails occurs at the OS layer, where minimal access is granted to log or journal files. All-access to these files is also logged. Coalfire additionally recommends the use of a centralized syslog or SIEM solution to provide greater protection of the critical audit trails.
10.5.1	Limit viewing of audit trails to those with a job-related need.	●	Using RBAC, view access to audit trails can be limited to those with a job-related need. OpenShift's audit trails are stored in the OS filesystem. Cluster admin access as well as access to the OS can be limited, controlled, and logged to minimize risk and the possibility of compromise to time data. Coalfire also recommends the use of a centralized syslog server or SIEM solution dedicated to log management, security, analysis, and reporting.
10.5.2	Protect audit trail files from unauthorized modifications.	●	Modification of the audit trail would require filesystem-level access at the OS layer. Cluster admins have the ability to access audit logs. Cluster admin access as well as access to the OS can be limited, controlled, and logged to minimize risk and the possibility of compromise to time data. Again, a central syslog or SIEM solution is recommended.
10.5.3	Promptly back up audit trail files to a centralized log server or media that is difficult to alter.	●	The system provides the means to integrate with a centralized log server for log shipping.

ID	REQUIREMENT	SUPPORT	NARRATIVE
10.5.4	Write logs for external-facing technologies onto a secure, centralized, internal log server or media device.	●	For all system components, logs can be directed to a centralized, internal log server or media device.
10.5.5	Use file-integrity monitoring or change-detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert).	○	Coalfire recommends using an external syslog server or SIEM solution with built-in or integrated file-integrity monitoring and change detection.
10.7	Retain audit trail history for at least one year, with a minimum of three months immediately available for analysis (for example, online, archived, or restorable from backup).	○	Coalfire recommends using an external syslog server or SIEM solution for log management. These solutions are more apt to provide the level of retention necessary to support the payment entity's analysis and reporting requirements as well as its retention requirements.
11.5	Deploy a change-detection mechanism (for example, file-integrity monitoring tools) to alert personnel to unauthorized modification (including changes, additions, and deletions) of critical system files, configuration files, or content files, and configure the software to perform critical file comparisons at least weekly. Note: For change-detection purposes, critical files are usually those that do not constantly change, but the modification of which could indicate a system compromise or risk of compromise. Change-detection mechanisms such as file-integrity monitoring products usually come pre-configured with critical files for the related operating system. Other critical files, such as those for custom applications, must be evaluated and defined by the entity (that is, the merchant or service provider).	●	With OCP 4, the addition of the File Integrity Operator allows for close monitoring of any file changes on the nodes. https://access.redhat.com/documentation/en-us/openshift_container_platform/4.6/html/security_and_compliance/file-integrity-operator

Table 2: Compliance Applicability of Red Hat OpenShift on RHEL CoreOS

CONCLUSION AND COALFIRE OPINION

It is Coalfire's opinion that the Red Hat OpenShift Container Platform hosted on Red Hat Enterprise Linux CoreOS (RHEL CoreOS) as reviewed can be effective in providing support for the outlined objectives and requirements of PCI DSS v3.2.1. Through proper implementation and integration into the payment entity's infrastructure, OpenShift and the underlying Red Hat OS may be usable in support of CDE. The OpenShift control plane, including the Control Plane, can be viewed as a Tier 2 system for scoping, while the OpenShift nodes that host CDE, including payment applications or other applications that process, transmit, or store CHD, would be considered as a Tier 1 system for scoping. The inclusion for assessment scoping for PCI DSS is conditional upon clearly defining network segmentation to separate CDE from non-CDE. OpenShift on RHEL CoreOS provides process, network, and storage isolation for containers in the environment.

Coalfire's opinion is based on observations and analysis of the provided documentation, interviews with Red Hat personnel, and hands-on engagement with a lab environment. The provided opinion is dependent upon several underlying presumptions or caveats. These caveats include adherence to vendor best practices and hardening of configuration as supported by the system components. This solution should be implemented in alignment with the payment entity's mission, values, business objectives, general approach to security, and the overall compliance program. Inclusion into the entity's overall compliance program includes considerations for supporting network infrastructure (build and maintain a secure network), segmentation efforts, physical security, personnel security, vulnerability testing, and penetration testing, and ongoing risk and compliance evaluation.

A COMMENT REGARDING REGULATORY COMPLIANCE

Coalfire disclaims any product's generic suitability to cause a payment card entity to use that product and achieve regulatory compliance solely. Payment card entities attain compliance through a Governance, Risk Management, and Compliance (GRC) program, not via the use of a specific product. This is true for payment card entities subject to PCI DSS 3.2.1 and customers targeting compliance with other regulations.

LEGAL DISCLAIMER

Coalfire expressly disclaims all liability concerning actions taken or not taken based on the contents of this white paper and the supporting controls workbook, and the opinions contained therein. The opinions and findings within this evaluation are solely those of Coalfire and do not represent any other parties' assessment findings or opinions. This document's contents are subject to change at any time based on revisions to the applicable regulations and standards (e.g., Health Information Portability and Accountability Act [HIPAA], PCI DSS, et al.) Consequently, any forward-looking statements are not predictions and are subject to change without notice. While Coalfire has endeavored to ensure that the information contained in this document has been obtained from reliable sources, there may be regulatory, compliance, or other reasons that prevent Coalfire from doing so.

Consequently, Coalfire is not responsible for any errors or omissions or the results obtained from the use of this information. Coalfire reserves the right to revise any or all this document to reflect an accurate representation of the content relative to the current technology landscape. To maintain this document's contextual accuracy, all references to this document must explicitly reference the document's entirety, including the title and publication date. Neither party will publish a press release referring to the other party or excerpting highlights from the document without the other party's prior written approval. For questions regarding any legal or compliance matters referenced herein, you should consult legal counsel, your security advisor, or the relevant standard authority.

ADDITIONAL INFORMATION, RESOURCES, AND REFERENCES

This section contains a description of the links, standards, guidelines, and reports used for the materials used to identify and discuss the features, enhancements, and security capabilities of the OpenShift Container Platform 4.

RED HAT RESOURCES

More information about Red Hat OpenShift Container Platform, including architecture, implementation, and administration, can be found on the OpenShift Container Platform documentation pages found at the following link: https://docs.openshift.com/container-platform/4.7/welcome/index.html

More information about Red Hat Enterprise Linux, including Release Notes, Installation, Migration, Systems Administration, Networking, and more may be found at the following link: https://access.redhat.com/documentation/en/red-hat-enterprise-linux/

A hardening guide for RHEL CoreOS on OCP 4.7 may be found at the following link: https://docs.openshift.com/container-platform/4.7/security/container_security/security-hardening.html

The OpenShift Security Guidebook may be found at the following link: https://access.redhat.com/articles/5059881

CENTER FOR INTERNET SECURITY RESOURCES

A benchmark for OpenShift 4.6 may be found at the following link: https://workbench.cisecurity.org/benchmarks/5248

PCI SECURITY STANDARDS COUNCIL DATA SECURITY STANDARD REFERENCES

The current version of the PCI DSS is 3.2.1 as of May 2018. This white paper references the PCI DSS Standard revisions from 2014-2018. The current revision may be accessed via the following link: https://www.pcisecuritystandards.org/document_library

An information supplement from the PCI SSC is the guidance for PCI DSS Scoping and Network Segmentation. The guidance document may be found at the following link: https://www.pcisecuritystandards.org/documents/Guidance-PCI-DSS-Scoping-and-Segmentation_v1_1.pdf

Details of the PCI SSC Cloud Special Interest Group updates to virtualization and cloud Information Supplement developed as PCI SSC Cloud Computing Guidelines, April 2018, are available at this location: https://www.pcisecuritystandards.org/pdfs/PCI_SSC_Cloud_Guidelines_v3.pdf

COALFIRE REFERENCES

The Coalfire corporate payment card references and the Solutions Engineering offerings may be found here: https://www.coalfire.com/industries/payments, https://www.coalfire.com/solutions/cyber-engineering

Coalfire corporate information is available at the following link: https://www.coalfire.com/about

ABOUT THE AUTHORS

Byron Estrada | Sr. Consultant, Product Guidance, Coalfire Systems, Inc.

Byron holds both a Bachelor of Science and Master of Science in Cybersecurity and is an author and thought leader on information security topics for Coalfire's clientele with a focus in enterprise architecture, security, and regulatory frameworks.

Jason Macallister | Senior Consultant, Coalfire Systems, Inc.

Jason consults on Information Security and regulatory compliance topics as they relate to advanced infrastructure, emerging technology, and cloud solutions.

Fred King | Principal I, Payments, Coalfire Systems, Inc.

Fred is responsible for client service delivery with an emphasis on Security Architecture. Fred brings three decades of technical and management experience. Fred has designed and delivered IT security, networking, and virtualization architectures in both internal and professional services roles. Fred is a leader, a trained and experienced technologist, a security architect, and a QSA in the payment card industry.

Chris Krueger, CISSP | Principal II, Product Guidance, Coalfire

As Principal, Chris contributes as an engineer, author, and thought leader on regulatory compliance, information security, and product design topics for Coalfire's customers in "new and emerging" cybersecurity arenas.

Published originally May 2018. Revised May 2021.

ABOUT COALFIRE

Coalfire is the trusted cybersecurity advisor that helps private and public sector organizations avert threats, close gaps, and effectively manage risk. By providing independent and tailored advice, assessments, technical testing, and cyber engineering services, we help clients develop scalable programs that improve their security posture, achieve their business objectives, and fuel their continued success. Coalfire has been a cybersecurity thought leader for over 20 years and has offices throughout the United States and Europe. For more information, visit Coalfire.com.