Guide to PCI Compliance for VMs Containers K8s - Microservices environment

Araali Networks

Feb 2020

Introduction

Modern cloud-native application architectures give developers the ability to test, iterate, and release applications and microservices at the speed of light. Also, companies are adopting hybrid and multi-cloud strategies to lower cost and keep vendors honest. This strategy helps companies build performance and resiliency as it allows for scaling and redundancy on one hand while mitigating cost and downtime risks on the other hand.

However, cloud-native and multi-cloud present its own set of challenges. The challenges, especially security and compliance, are more profound for companies in the regulated industry. Enterprises have to reevaluate and rethink their security processes and security solutions to remain compliant with regulatory controls.

What is PCI DSS

The Payment Card Industry Data Security Standard (PCI DSS) was developed to enhance cardholder data security and facilitate the broad adoption of consistent data security measures globally. This will minimize theft, fraud, and misuse. Anyone that accepts transmits or stores cardholder data must comply with the PCI DSS. This includes merchants, processors, acquirers, issuers, and service providers. PCI DSS also applies to all other entities that store, process, or transmit cardholder data (CHD) and/or sensitive authentication data (SAD).

PCI DSS provides a baseline of technical and operational requirements designed to protect account data. Below is a high-level overview of the 12 PCI DSS requirements

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

What is PCI?

PCI or Payment Card Information is the sensitive data processes by different entities mentioned above. Is has two components - cardholder data and sensitive authentication data which are defined as follows:

Card Holder Data (CHD):

- Primary Account Number (PAN)
- Cardholder Name
- Expiration Date
- Service Code

Sensitive Authentication Data (SAD):

- Full track data (magnetic-stripe data or equivalent on a chip)
- CAV2/CVC2/CVV2/CID
- PINs/PIN blocks

Sensitive Authentication Data must not be stored after authorization, even encrypted. The cardholder data can be stored only the PAN must be rendered unreadable according to PCI DSS Requirement 3.4

Who is in scope?

Any company/entity collecting or processing the PCI data (merchants, processors, acquirers, issuers, and service providers) is in the scope of PCI DSS. Some PCI DSS requirements may also apply to organizations that outsource their payment operations or manage their cardholder data environment (CDE). Additionally, organizations that outsource their CDE or payment operations to third parties are responsible for ensuring that the account data is protected by the third party per the applicable PCI DSS requirements.

What systems are in scope?

The PCI DSS security requirements apply to all system components included in or connected to the cardholder data environment. The cardholder data environment (CDE) is comprised of people, processes, and technologies that store, process, or transmit cardholder data or sensitive authentication data. “System components” include network devices, servers, computing devices, and applications.

How microservices architecture and mix of VMs, Containers, and K8s affect compliance

Microservice architecture leads to decoupled modular apps that are great for application development as every team can own and work on its piece but increase the attack surface and, thereby, security risks. Similarly, the adoption of containers allows organizations to consume open-source software and libraries at a higher rate. This may introduce vulnerabilities and evade the vetting process based on existing version and configuration management. Containers also run on a shared kernel that limits isolation and requires dynamic networking, limiting observability with existing security tools like IDS/IPS and firewalls. Finally, Kubernetes brings dynamism in the environment where containers can be orchestrated on the go, and IPs become ephemeral. This limits the usage of traditional security based on IPs and was suitable for a static environment.

Most of the companies using containers and k8s still have a sizable fleet of VMs and Bare Metal (BMs). Most of the new container focussed tools don’t cover VMs and BMs well. This leaves enterprises with a zoo of tools to manage these hybrid setups making them tool rich yet capability poor.

Segmentation

Network segmentation of the cardholder data environment from the remainder of an entity’s network is not a PCI DSS requirement. However, it is strongly recommended as a method that reduces:

- 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 cardholder data into fewer, more controlled locations)

Without adequate network segmentation, the entire network is in the scope of the PCI DSS assessment. Network segmentation can be achieved through several physical or **logical** means, such as properly configured internal network firewalls, routers with a strong access

control lists, or other technologies that restrict access to a particular network segment. To be considered out of scope for PCI DSS, a system component must be properly isolated (segmented) from the CDE. Even if the out-of-scope system component was compromised it could not impact the security of the CDE.

How Araali addresses PCI DSS

Araali leverages the PCI DSS framework v3.2 published by the PCI security standards council in Apr 2016 (link).

PCI DSS compliance is reached by satisfying sub-categories specified in the cybersecurity framework. An organization looking for PCI compliance should work with its auditors to fully understand and appreciate the subtle nuances around all control categories. The following table summarizes the specific sub-categories of controls that Araali can help achieve through its security solution.

PCI DSS Requirements	Araali
1: Install and maintain a firewall configuration to protect cardholder data	
1.1 Establish and implement firewall and router configuration standards that include the following: 1.1.1 Establish and implement firewall and router configuration standards that include the following:	Araali is a network overlay that provides distributed firewall functionality. Each app/asset gets its personalized firewall to make localized access control decisions.
1.1.1 A formal process for approving and testing all network connections and changes to the firewall and router configurations 1.1.1.a Examine documented procedures to verify there is a formal process for testing and approval of all: (a) Network connections and (b) Changes to firewall and router configurations	Araali has inbuilt workflows that allow a single team or multiple teams to verify and accept auto-created policies or create access policies (Configurations). The security team can finally review the policy before it goes live. The configurations can be seen in both tabular as well as network diagram format.
1.1.2 Current network diagram that identifies all connections between the cardholder data environment and other networks, including any wireless networks	Araali out of the box provides network topology of the environment (Zones), application, containers/VM, and process. This gives a bird's eye view of all connections between CDE and other environments and whether access control rules are defined.
1.1.3 Current diagram that shows all cardholder data flows across systems and networks	Araali provides an out-of-the-box view of all the apps and processes communicating with the database (holding customer data). This can be used to map and track data access

	and flow in the environment.
1.1.4 Requirements for a firewall at each Internet connection and between any demilitarized zone (DMZ) and the internal network zone	Araali enables micro-segmentation using distributed app firewalls across VMs, containers, and Kubernetes leveraging application ids and whitelisted application policies.
1.1.5 Description of groups, roles, and responsibilities for management of network components	Araali gives RBAC capabilities to allow the right personnel to configure and manage application rules. The roles could even be assigned based on app ownership (i.e., teams responsible for a particular app can review and manage their own rules)
1.1.6 Documentation of business justification and approval for use of all services, protocols, and ports allowed, including documentation of security features implemented for those protocols considered to be insecure.	Araali automatically discovers all the ports (used and unused), which makes it easy for security teams to show documentation of all services, and ports used. Also, only allowed/used ports are whitelisted by personnel, and rest are automatically dropped/closed. Araali gives out of box visibility into vulnerability (known CVEs) and privileges (root or other privileges) to allow personnel to prioritize securing apps that are running high privileges with vulnerability.
1.1.7 Requirement to review firewall and router rule sets at least every six months	Araali creates a list of whitelisted policies by app and zone which are distributed by app owners and are easy to review on an ongoing basis.
1.2 Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment	Araali enables micro-segmentation using distributed app firewall leveraging application ids and whitelisted application policies.
1.2.1 Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment and specifically deny all other traffic	Araali enables every application to enforce process-level inbound and outbound whitelisted policies to allow particular traffic and deny rest.
1.2.3 Install perimeter firewalls between all wireless networks and the cardholder data environment and configure these firewalls to deny or, if traffic is necessary for business purposes, permit only authorized traffic between the wireless environment and the cardholder data environment.	Araali enables micro-segmentation using a distributed app firewall. This can be used to permit data only from a certain environment into the cardholder data environment.

1.3 Prohibit direct public access between the Internet and any system component in the cardholder data environment.	Araali policies can enforce that no traffic comes from the internet (unknown IP) into the cardholder data environment. Traffic can only come from whitelisted internal approved zones.
1.3.1 Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.	Araali can be deployed on DMZ VMs or Containers to whitelist the environment and only allow legitimate access from DMZ to the cardholder data zone.
1.3.2 Limit inbound Internet traffic to IP addresses within the DMZ.	Araali policies can be used to enforce that external traffic can only come from DMZ, which can be enforced at a process grain.
1.3.3 Implement anti-spoofing measures to detect and block forged source IP addresses from entering the network. (For example, block traffic originating from the Internet with an internal source address.)	Araali uses application identity to verify and give privileges. The internal traffic is allowed to talk based on the identity and not IP address. This problem is automatically solved based on how Araali is implemented.
1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet	Araali out of the box enables applications to enforce process-level outbound policies to traffic going to other Araali protected assets or the Internet.
1.3.5 Permit only “established” connections into the network. (A firewall that maintains the "state" (or the status) for each connection through the firewall knows whether an apparent response to a previous connection is actually a valid, authorized response (since it retains each connection’s status) or is malicious traffic trying to trick the firewall into allowing the connection.)	Araali is a distributed app-based firewall and maintains the state of all connections out of the box. It uses technologies to prevent getting tricked from malicious traffic.
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.	Araali enables micro-segmentation using a distributed app firewall. This can be used to ensure internal trusted apps can only access the database.

1.4 Install personal firewall software or equivalent functionality on any portable computing devices (including company and/or employee-owned) that connect to the Internet when outside the network (for example, laptops used by employees), and which are also used to access the CDE. Firewall (or equivalent) configurations include: Specific configuration settings are defined. Personal firewall (or equivalent functionality) is actively running. Personal firewall (or equivalent functionality) is not alterable by users of the portable computing devices.	Araali can be deployed on Linux and Mac endpoints to verify and only allow trusted connections.
1.5 Ensure that security policies and operational procedures for managing firewalls are documented, in use, and known to all affected parties.	Araali gives out of box documentation of rules split by applications. These can be reviewed and maintained by individual app teams allowing easier management.
2: Do not use vendor-supplied defaults for system passwords and other security parameters	
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.).	Araali default is to "deny all" in Enforcement Mode. Its policies have to be configured to be used in enforcement mode.
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.	Araali provides an out of box list of vulnerabilities (CVEs) for apps that might be running on VM or containerized environments.
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.)	Araali shows out of the box show which app is running on which server (asset inventory). This can be used to verify that only one app is running per server.
2.2.2 Enable only necessary services, protocols, daemons, etc., as required for the function of the system.	Araali is based on a whitelisted model, which only allows the approved process to run and denies the rest.

2.2.3 Implement additional security features for any required services, protocols, or daemons that are considered to be insecure	Araali can be whitelisted to talk to only Araali and not any IP in the cardholder data zone. This will ensure that even if servers are deployed without Araali security, they will not be able to communicate with the rest of the applications.
2.4 Maintain an inventory of system components that are in scope for PCI DSS.	Araali gives out of box inventory of all servers (IPs), applications, and processes communicating in the system.
3. Protect stored cardholder data	
3.6.2 Secure cryptographic key distribution	Araali does not burn any secrets into the containers and distributes it during runtime and manages its lifecycle including rotating and destroying keys
5: Protect all systems against malware and regularly update anti-virus software or programs	
5.1 Deploy anti-virus software on all systems commonly affected by malicious software (particularly personal computers and servers).	Araali monitors every asset and application for unknown behavior patterns in a deterministic manner. It can detect virus or intrusion attempts by monitoring deviations and raising alerts intelligently routed to the app owner.
5.2 Ensure that all anti-virus mechanisms are maintained as follows: Are kept current, Perform periodic scans Generate audit logs which are retained per PCI DSS Requirement 10.7.	Araali continuously monitors and creates a log for every communication on the network. These logs are stored with Araali to prevent tampering or deletion of records
5.3 Ensure that anti-virus 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.	Araali agents cannot be disabled. It is part of Araali default policies.

6: Develop and maintain secure systems and applications	
6.1 Establish a process to identify security vulnerabilities, using reputable outside sources for security vulnerability information, and assign a risk ranking (for example, as “high,” “medium,” or “low”) to newly discovered security vulnerabilities.	Araali gives out of box view of vulnerability for every app ranked as critical, high, medium, and low with links to National CVE database
6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release	Araali can automatically patch its software component on the fly, in a way that does not disrupt the business
6.3.1 Remove development, test and/or custom application accounts, user IDs, and passwords before applications become active or are released to customers.	Araali gives identity to every app, and they mutually authenticate based on identity and not default passwords.
6.4.1 Separate development/test environments from production environments, and enforce the separation with access controls.	Araali uses zone/environment setting to ensure that integrity is maintained, and apps from the test/dev zone cannot communicate to other production zones.
6.4.2 Separation of duties between development/test and production environments	Araali uses zone/environment setting to ensure integrity is maintained. Also, Araali uses RBAC that only lets the right app owners see their apps and policy.
6.5.1 Injection flaws, particularly SQL injection. Also consider OS Command Injection, LDAP and XPath injection flaws as well as other injection flaws.	Araali can prevent compromised apps from laterally moving, or talking to command and control (i.e., preventing new behaviors that are not whitelisted/seen before)
6.5.2 Buffer overflows	Araali can prevent compromised apps from laterally moving, or talking to command and control (i.e., preventing new behaviors that are not whitelisted/seen before)
6.5.4 Insecure communications. 6.5.4 Examine software-development policies and procedures and interview responsible personnel to verify that insecure communications are addressed by coding techniques that properly authenticate and encrypt all sensitive communications.	Araali can provide inflight encryption MTLS between the apps.
6.5.6 All “high risk” vulnerabilities identified in the vulnerability identification process (as defined in PCI DSS Requirement 6.1).	Araali gives out of box visibility into vulnerability (known CVEs) and privileges (root or other privileges) to allow personnel to prioritize securing apps that are running high privileges with vulnerability.
6.5.7 Cross-site scripting (XSS)	Araali can prevent compromised apps from laterally moving, or talking to command and control (i.e., preventing new behaviors that

	are not whitelisted/seen before)
6.5.9 Cross-site request forgery (CSRF)	Araali can prevent compromised apps from laterally moving, or talking to command and control (i.e., preventing new behaviors that are not whitelisted/seen before)
7: Restrict access to cardholder data by business need to know	
7.1 Limit access to system components and cardholder data to only those individuals whose job requires such access:	Araali uses RBAC that only allows the right app owners to see their apps and policy.
7.1.2 Restrict access to privileged user IDs to least privileges necessary to perform job responsibilities.	Araali annotates out of the box, which apps might be running with root privilege. Whitelisted policies ensure only the least privileges are used to perform the job.
7.1.3 Assign access based on individual personnel's job classification and function	Araali uses RBAC that only allows the right app owners to see their apps and policy.
7.2.3 Default "deny-all" setting.	Araali default is to deny all. Its policies have to be configured to be used in enforcement mode.
8: Identify and authenticate access to system components	
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: 8.1.1 Assign all users a unique ID before allowing them to access system components or cardholder data	Araali uses RBAC that only allows the right app owners to see their apps and policy.
8.2.1 Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components	Araali adds a second-factor auth on top of existing secrets and passwords. The application can only communicate once they are authenticated and authorized by Araali.

10: Track and monitor all access to network resources and cardholder data	
10.1 Implement audit trails to link all access to system components to each individual user	Araali keeps an audit of all application access by other applications
10.2 Implement automated audit trails for all system components to reconstruct the following events: 10.2.2 All actions taken by any individual with root or administrative privileges	Araali collects detailed process level logs for app communications. These are analyzed for policy creations, alerts, and zero-day vulnerability detection and can be shared with SIEM
10.3 Record at least the following audit trail entries for all system components for each event: 10.3.1 User identification 10.3.2 Type of event 10.3.3 Date and time 10.3.4 Success or failure indication 10.3.5 Origination of event 10.3.6 Identity or name of affected data, system component, or resource	Araali generated a detailed contextual audit of every event in an app. These audits cannot be tampered and can only be accessed based on RBAC from Araali UI.
10.5 Secure audit trails so they cannot be altered	Araali generated a detailed contextual audit of every event in an app. These audits cannot be tampered and can only be accessed based on RBAC from Araali UI.
10.5.1 Limit viewing of audit trails to those with a job-related need.	Araali generated a detailed contextual audit of every event in an app. These audits cannot be tampered and can only be accessed based on RBAC from Araali UI.
10.5.2 Protect audit trail files from unauthorized modifications	Araali generated a detailed contextual audit of every event in an app. These audits cannot be tampered and can only be accessed based on RBAC from Araali UI.
10.5.3 Promptly back up audit trail files to a centralized log server or media that is difficult to alter.	Araali backs up all the audit trails and can keep the data for any specified period.

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).	Araali generated a detailed contextual audit of every event in an app. These audits cannot be tampered and can only be accessed based on RBAC from Araali UI.
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).	Araali backs up all the audit trails and can keep the data for any specified period.
10.9 Ensure that security policies and operational procedures for monitoring all access to network resources and cardholder data are documented, in use, and known to all affected parties.	Araali gives out of box view of all the policies, by applications which could be distributed to application owners for review and lifecycle management
11: Regularly test security systems and processes.	
11.2 Run internal and external network vulnerability scans at least quarterly and after any significant change in the network (such as new system component installations, changes in network topology, firewall rule modifications, product upgrades).	Araali gives out of box scans of zones, apps, and assets in the cardholder zone.
11.2.1 Perform quarterly internal vulnerability scans. Address vulnerabilities and perform rescans to verify all “high risk” vulnerabilities are resolved in accordance with the entity’s vulnerability ranking (per Requirement 6.1). Scans must be performed by qualified personnel.	Araali runs a realtime scan to show the allowed process, vulnerable apps (CVEs), and privileges. All this information is rank to help with prioritization.
11.4 Use intrusion-detection and/or intrusion-prevention techniques to detect and/or prevent intrusions into the network. Monitor all traffic at the perimeter of the cardholder data environment as well as at critical points in the cardholder data environment, and alert personnel to suspected compromises. Keep all intrusion-detection and prevention engines, baselines, and signatures up to date.	Araali monitors every asset and application for unknown behavior patterns in a deterministic manner. It can detect virus or intrusion attempts by monitoring deviations and raising distributed alerts to the app team.
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,	Araali fingerprints apps during the fortification process. This prevents any unknown or modified workload from running and communicating

configuration files, or content files; and configure the software to perform critical file comparisons at least weekly.	
12: Maintain a policy that addresses information security for all personnel	
12.1.1 Review the security policy at least annually and update the policy when the environment changes.	Araali provides policies in a way that is easy to comprehend and review (tabular and network diagram). Any changes in the environment automatically trigger policy change that has to be reviewed and accepted.
12.10.3 Designate specific personnel to be available on a 24/7 basis to respond to alerts.	Araali can distribute alerts based on apps to the right personnel.
12.10.5 Include alerts from security monitoring systems, including but not limited to intrusion-detection, intrusion-prevention, firewalls, and file integrity monitoring systems.	Araali agent runs on the host and does intrusion detection, intrusion prevention (Enforcement Mode), access control, firewall, and file integrity monitoring. All these alerts are automatically stitched together, and Araali also offers full Alert Lifecycle Management to track them for security and compliance.

About Araali

For CISOs and App teams, who care about security and compliance without loss in productivity, Araali Networks in a distributed firewall leveraging application identity and context. It covers any workload - BM, VM, container and serverless, and any app - customer or open-source without re-compile. Unlike traditional firewall and contemporary container security companies, Araali protects any workload and leverages its Infrastructure-Agnostic and Location-Transparent policies, which are not IP but app-id based. This makes cloud adoption simpler while lowering TCO and improving security posture.