aqua

Compliance Guide:

Achieving PCI-DSS
Compliance for Containers

PCI-DSS v3.2.1

Introduction

Many organizations struggle to address quarterly and annual PCI-DSS requirements for containers, in cases where the container life cycle is shorter than the duration of a given PCI-DSS control, you should consider sampling running instances across all in-scope container images. As part of the container instantiation pipeline, an organization may want to perform vulnerability testing to address the challenge of implementing security controls in a highly fluid and elastic environment. Vulnerability testing also evaluates the impact of PCI-DSS requirements.

This guide will support your effort to achieve PCI compliance. However, it will not help you attain full compliance, as the PCI-DSS requirements are not fully applicable to containerized workloads. For example, Requirement 9 states “Restrict physical access to cardholder data.”

PCI-DSS Checklist for Container Security

Containers introduce dramatic changes to application development. They often drive an increase in the use of open-source components, and they also accelerate the pace of software development, challenging established security checkpoints to keep up. This new process may also introduce vulnerabilities and evade vetting processes based on existing version and configuration management.

Key areas where containers may impact PCI compliance include:

Vulnerability management: Containers that use open-source images may contain vulnerabilities. These images should be monitored for security vulnerability information and mitigated before being used in production.

Network security: Containers introduced challenges in tracking where containers are running. The network connections between the different containers should be identified, at any given time, to prevent network traversal and intrusion.

Threat analysis and mitigation: One of the pillars of any given containerized environment is its policy-based security rules that can maintain an automated check for ongoing monitoring and prevention of malicious activity.

User access control, segregation of duties: Containers should be accessible only to specific individuals, with specific job-related needs.

Data Protection: Real-time visibility and event audit trails. Access to PCI-sensitive data and systems must be logged and audited. In addition, access to these files must be restricted and backed-up on a regular basis. When working with containers, existing audit methods may not have sufficient functionality to track this data in this environment.

How Aqua Helps Address PCI-DSS Compliance

Following are detailed explanations and examples of how Aqua CSP addresses many PCI-DSS requirements:

Vulnerability Scanning: Aqua provides in-depth vulnerability assessment for container images, preventing vulnerabilities from getting into applications before they're deployed. By default, Aqua scans images daily to find CVEs. Each image is scanned for vulnerabilities, both in its OS packages and in the development language files. With the image assurance policies, admins can determine which image is allowed or disallowed. This means that only approved images are allowed to run.

Network nano-segmentation: In order to monitor and secure all network connections, Aqua provides a network firewall that prevents unauthorized network connections and nano-segmentation of the network, in order to observe the relationship between groups of containers.

Policy-based security: Aqua admins can also assign labels to images and create a security policy that specifies which of the images are allowed to enter production. Any specific activity that does not comply with the policy will be stopped.

Secrets Management: Aqua provides central management and secure distribution of secrets and cryptographic keys into running containers. When a secret is used, its value will be automatically injected into the container and will disappear once the container stops running. The secret value is never visible outside of the container.

Separation of Duties and Access Control: In a containerized environment, the development team should have limited access to production. Aqua's access control model ensures that only specific users are allowed to view or access specific containers along the pipeline.

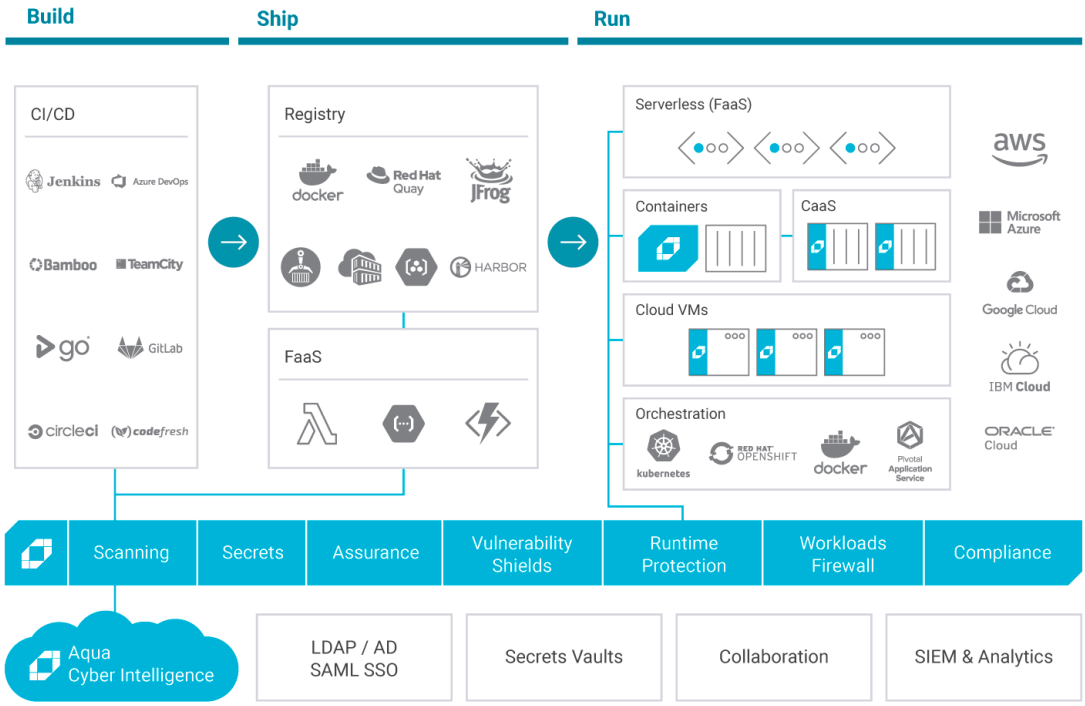
Full Event Logging: Aqua provides a granular audit trail for container events such as start/stop, access, and attempted access events and activities that contravene the security policy. Integration with third-party tools, such as Splunk and a variety of SIEM tools, allows events to be centrally collected, analyzed, and protected from being deleted.

Aqua Container Security Platform: Quick Overview

Aqua's platform is a container-native, full lifecycle solution for securing container-based applications. Aqua CSP is comprised of the following main components:

Aqua Server: This is a central management component and can be deployed on multiple instances for high availability. It provides policy management, image and function scanning, image/function lifecycle controls, monitoring, and reporting. It also integrates with image/functions registries for scanning, with CI/CD tools for security testing as part of the build, and with SIEM/analytics to generate audit and alert data. The Command Center exposes full API access and a management console UI.

Aqua Cyber Intelligence: Aggregates and correlates multiple sources, including NVD, vendor advisories, and proprietary research, providing continuous, up-to-date information to Aqua's vulnerability scanning, malware detection, and threat mitigation features.



Aqua Enforcers: These are designed to enforce security policies that the admin created. They also generate audit events in the system. Each enforcer serves a different purpose and is designed for a specific use case. This allows the admin to implement security policies on the entire cloud-native environment.

Aqua Enforcer Monitors the runtime activity of containers and hosts and provides for their runtime security

Aqua MicroEnforcer Provides runtime protection for containers in PaaS environments in which a host-based solution cannot be deployed

VM Enforcer Provides protection for hosts (virtual machines) without Docker or Kubernetes, monitoring the host image and providing Host Runtime Policies to restrict and monitor specified runtime activities

Aqua Nano Enforcer Provides runtime protection for AWS Lambda functions, provides protection against malicious executables, controls the types of executables that can run, and detects malicious behavior in runtimeactivities

Detailed PCI-DSS Requirements and Aqua CSP Response

The following are detailed examples of how the Aqua Container Security Platform addresses the relevant PCI requirements.

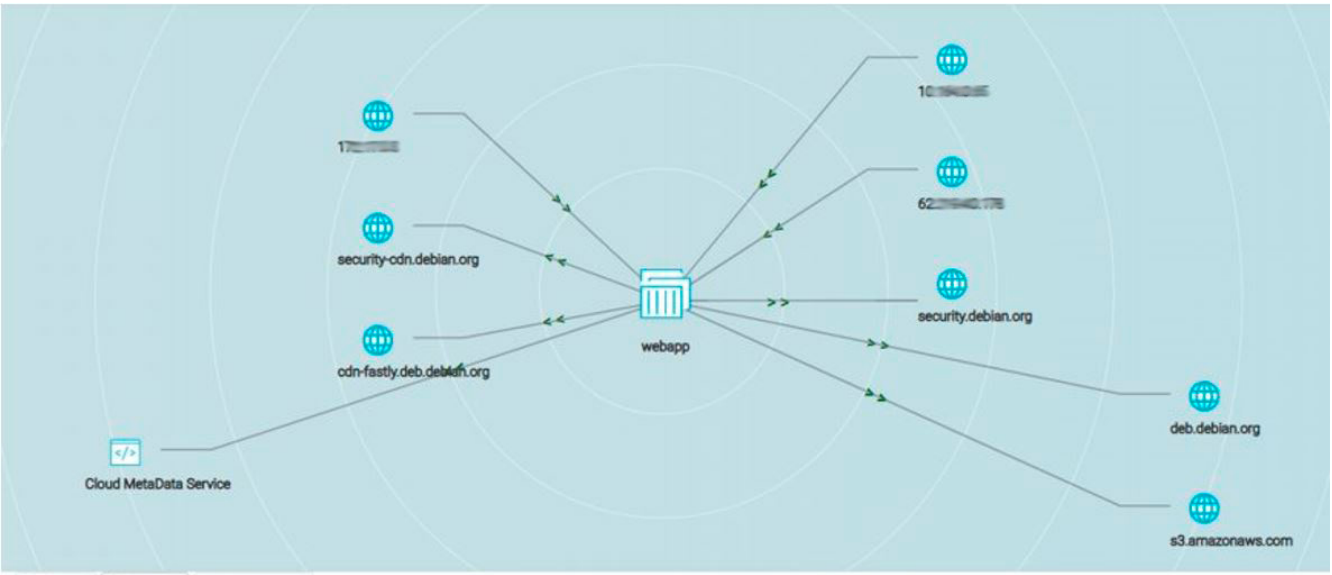
Requirement 1

Install and maintain a firewall configuration to protect cardholder data.
All systems must be protected from unauthorized access from untrusted networks, whether when entering the system via the Internet as e-commerce, employee Internet access through desktop browsers, employee e-mail access, dedicated connections such as business-to-business connections, via wireless networks, or via other sources.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
1.1.2	Current network diagram that identifies all connections between the cardholder data environment and other networks, including any wireless networks	Automatically detects and maps internal and external network connections with Aqua's Network map
1.1.4	Requirements for a firewall at each Internet connection, between any demilitarized zone (DMZ), and the internal network zone	Sets container-level firewall policies and nano-segmentation
1.2	Build firewall and router configurations that restrict connections between untrusted networks and any system components in the cardholder data environment	Initiates container-level firewall policies and nano-segmentation
1.2.1	Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment and specifically deny all other traffic	Sets container profile network rules and container firewall rules
1.3.4	Do not allow unauthorized outbound traffic from the cardholder data environment to the Internet	Sets container profile network rules, based on labeling PCI-related services.

Requirement 1.1.2

Admins can view a visual representation of the network topology and associated connections of a service. This map enables admins to capture monitored connections and use them as a base for the container/pod-level firewall policy.



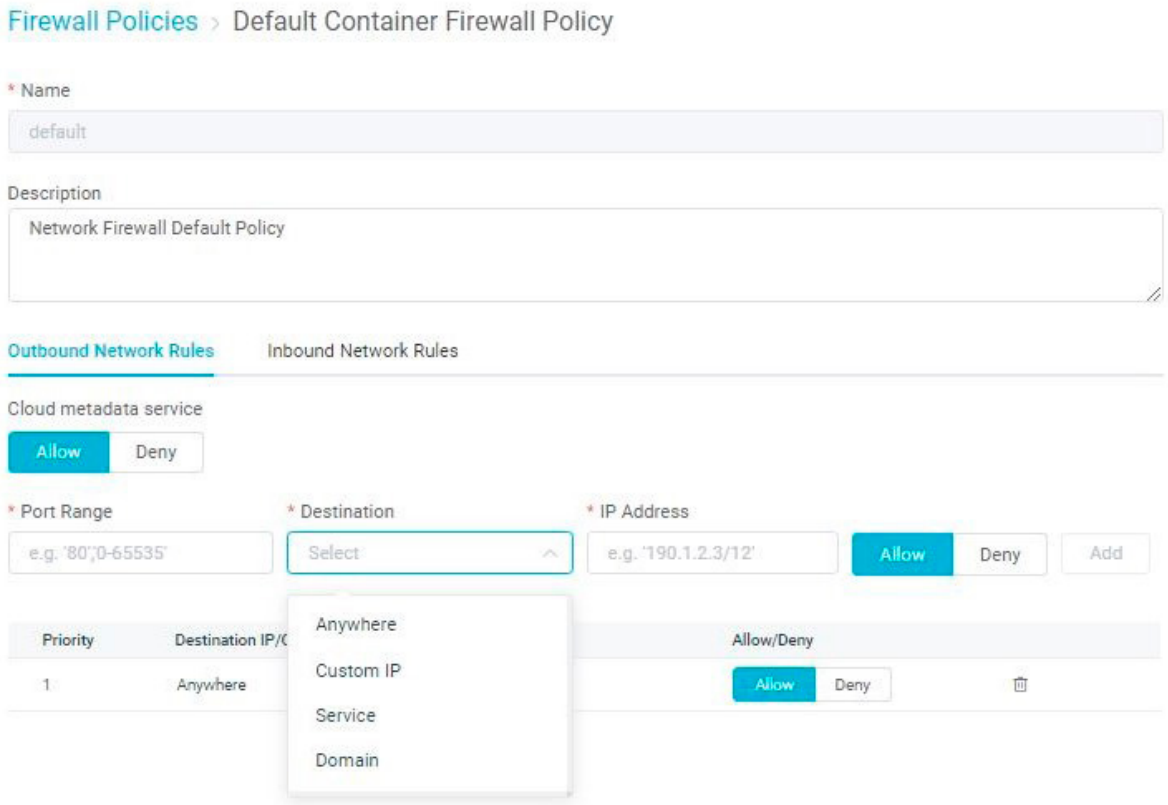
Clear | Pause | Save Rules

IP/CIDR	PORT RANGE	ALLOW/DENY	INBOUND / OUTBOUND
10.	80	Allow	Inbound
17	80	Allow	Inbound
62.	80	Allow	Inbound
security-cdn.debian.org	80	Allow	Outbound

Requirement 1.1.4

Aqua's Container Firewall (Nano- Segmentation) automatically discovers container network topology, both within a host and across hosts, and apply context-based firewall rules that alert or prevent unauthorized network connections. This capability allows the creation of network boundaries across services, where admins can control which networks are accessible for each service.

In addition, admins can manually modify communication rules/policies based on actual activity, without impacting container performance and availability.



Requirement 1.2

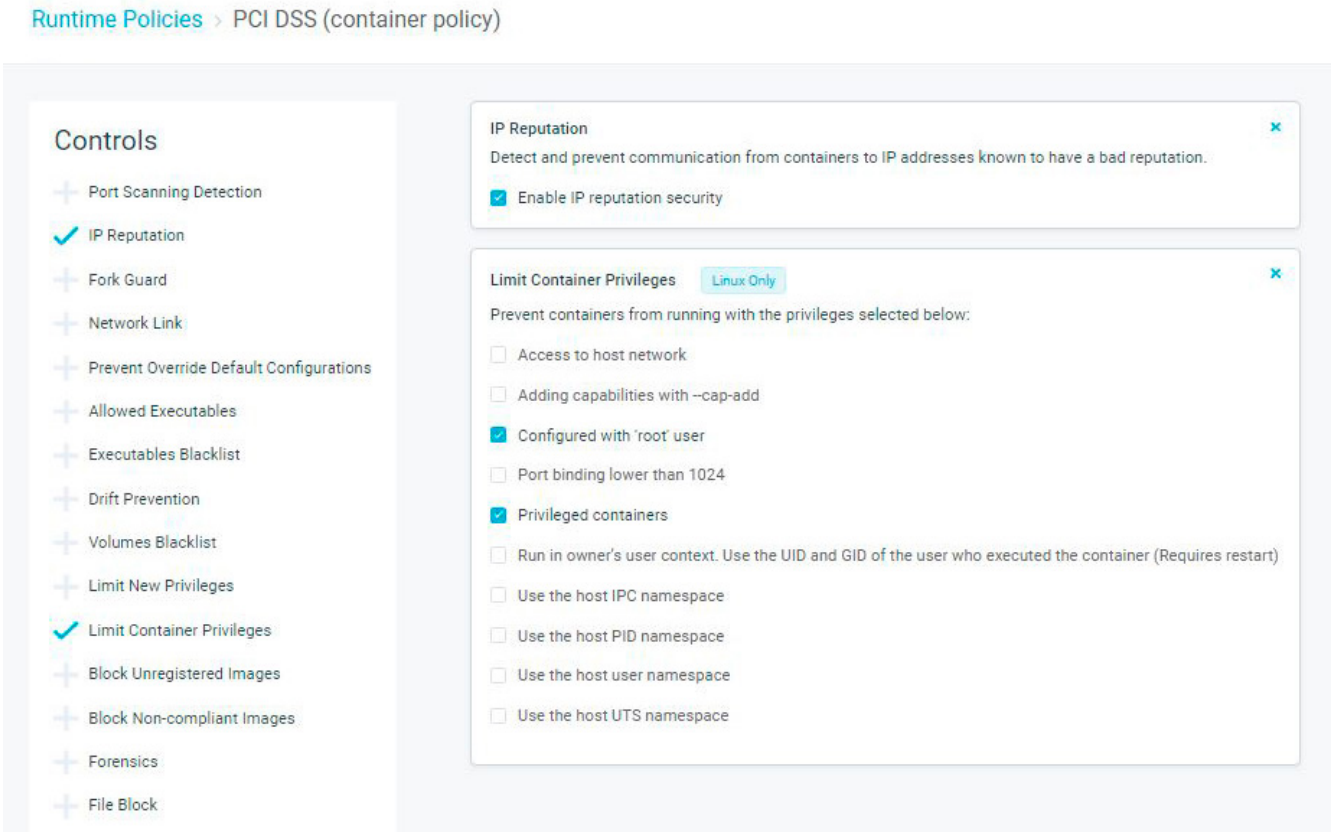
With Aqua's Container Firewall (Nano- Segmentation), admins can limit the network connectivity between services by applying a firewall-like concept for the container environment. This capability allows the creation of network boundaries across services, where admins can control which networks are accessible for each service. Aqua's label-based management allows you to label groups of containers as PCI-sensitive. Aqua admins can use labels and services to automatically group containers that can be deployed on separate nodes and network segments.

Requirement 1.2.1

Aqua provides several ways to limit container network traffic. In addition to nano-segmentation (explained in 1.1.2, 1.1.4, and 1.2), the security profiles of containers can categorically deny outbound or inbound connections. For example, a database container will typically not require outbound connectivity, so it will be denied automatically.

Requirement 1.3.4

Aqua's container firewall functionality can also be used to create global rules that prevent containers tagged with a certain label (for example – PCI-DSS compliance) from having outbound or inbound connections. The functionality might only permit some containers to access specific IP addresses/ URLs.



For more information on how to comply with PCI-DSS requirement 1, see Aqua's documentation: Firewall Policies

Requirement 2

Do not use vendor-supplied defaults for system passwords and other security parameters. Malicious individuals (external and internal to an entity) often use vendor default passwords and other vendor default settings to compromise systems. These passwords and settings are well known by hacker communities and are easily found via public channels.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
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.	Aqua CSP allows the user to change the default passwords via environment variables.
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.	Enforces CIS Benchmarks, best practices, and image vulnerability scanning.
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.	Installs the Aqua Console on a dedicated host, As documented in the PCI-DSS requirements. For more information, review Aqua's documentation on security best practices.
2.2.2	Enable only necessary services, protocols, daemons, etc., as required for the function of the system.	Follows behavioral container security profiles that whitelist legitimate activities and enforces the least functionality.
2.2.3	Implement additional security features for any required services, protocols, or daemons that are considered to be insecure.	Employs global security controls that enforce configuration and provides container-specific threat mitigation capabilities.
2.3	Encrypt all non-console administrative access using strong cryptography.	Communication between Aqua's entities are made via secured channels.
2.4	Maintain an inventory of system components that are in scope for PCI-DSS.	Performs automated registry scanning, provides views of all running containers pods, namespaces, deployments, and issues compliance reports.

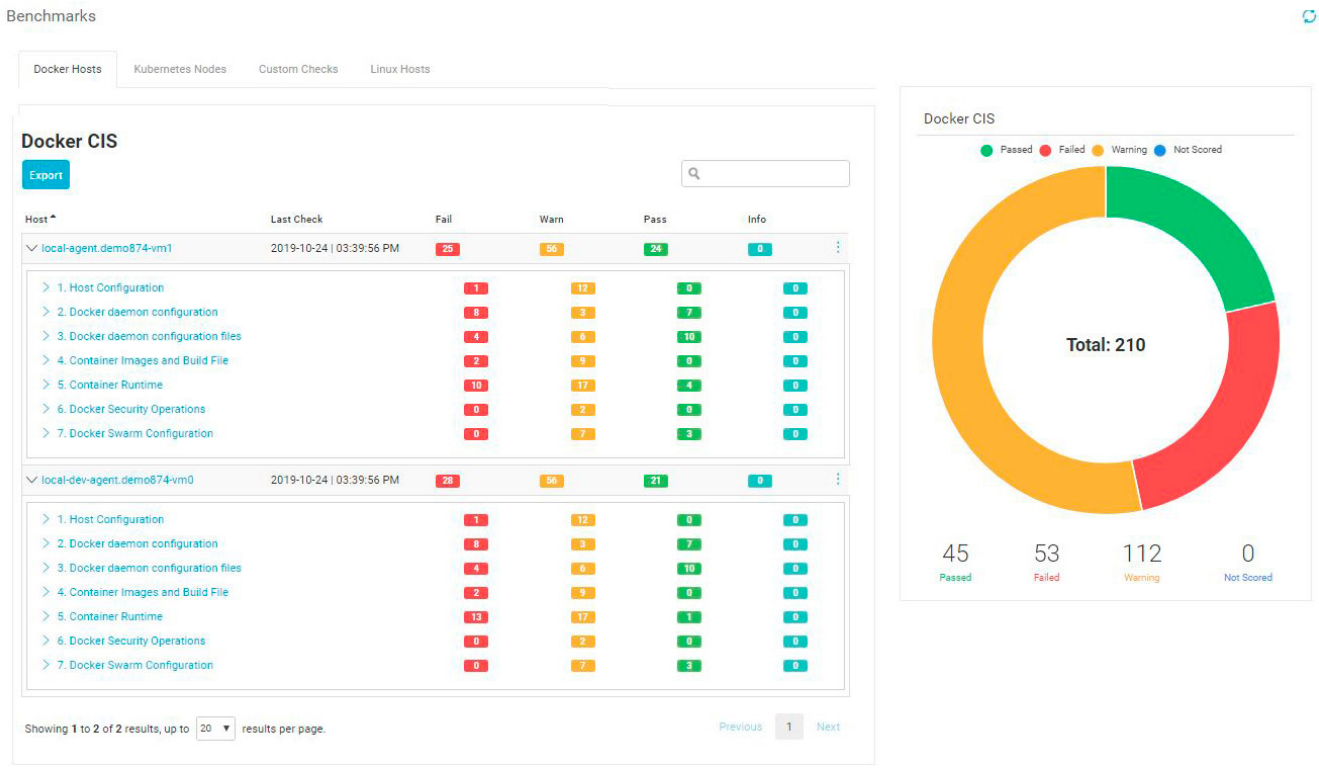
Requirement 2.1

Review detailed information on how to change the default passwords in Aqua's documentation in the environment variables section.

Requirement 2.2

Aqua supports the CIS Docker, Kubernetes and Linux benchmarks for host hardening, compliance reports per host, and a comprehensive image vulnerability report. The Host CIS screen provides information regarding compliance with the CIS Benchmarks, which has best practices for an engine configuration, container runtimes, and host configurations.

Aqua admins can build policies based on CIS benchmarks and review the degree of compliance with Aqua Host CIS reports. Additionally, Aqua's image scanning provides a deep analysis of known vulnerabilities, configuration errors, malware, embedded secrets, and sensitive data in images, including binaries and packages, supporting numerous programming languages.



Requirement 2.2.2

Aqua allows admins to enforce the least functionality with the automatic creation of a runtime profile for containers. Aqua Enforcer profiles the behavior of a given container and analyzes and reports on the security profile by noting OS resource use, file access, volume mounts, executables, system calls, and inbound/outbound network connections. This provides a baseline for all the legitimate container activity and uses machine learning to create a whitelisting policy. Admins can also manually tweak the profile parameters for specific parameters.

Furthermore, with the Aqua image assurance policy, images that have not been vetted will be blocked by Aqua (e.g., images that were not scanned, or images with over-provisioned access permissions). This means that only approved images will be allowed to run. The Aqua Image Assurance Policy works across multiple orchestration tools (e.g., K8s, OpenShift, DC/OS, Docker Swarm).

Requirement 2.2.3

Aqua provides numerous settings that can enforce best practices, such as the use of sanctioned base images.

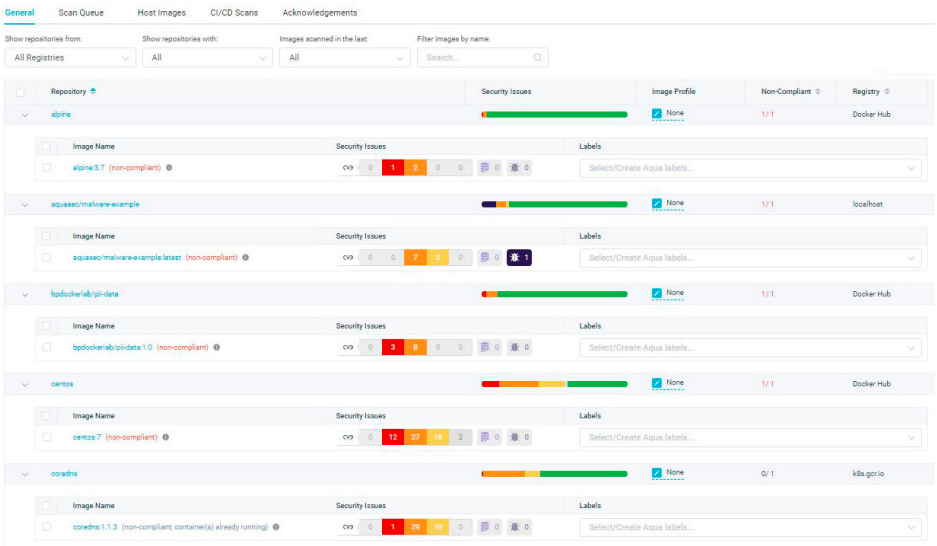
This prevents:

- Containers from running as root
- Containers from running executables that were not in the original image
- Image drift, i.e., the use of unsanctioned image versions

Additionally, any behavior that is not allowed by the container behavioral profiles described in 2.2.2 will result in generating and logging alerts.
Furthermore, Aqua admins can apply and enforce custom runtime policies to specific runtime environments (e.g., apply blacklisted executables per namespace or not allow unregistered images in a PCI-related cluster). Aqua's security research team provides threat mitigation protection ("IPS for containers") that blocks specific behaviors that are indicative of attacks, such as fork bombs and attempts to access malicious IP addresses.

Requirement 2.4

Aqua provides an inventory of containerized applications, covering the different repositories, images, functions, containers, and hosts in the organization. Aqua connects to image and function registries and enumerates all images/functions stored in them. Aqua also processes all images stored on the hosts that were not pulled from an image registry and creates a package inventory for every image.
In addition, Aqua provides a view of all running containers and their originating images, including package inventory for every running container. By using Aqua labels, users can attach a simple name to any resource to create a label-based rule for operations or inventory purposes. For example, admins can label container groups as sensitive or inherit security group definitions from the orchestrator (e.g., PCI sensitive).



For more information on how to comply with Requirement 2, see aqua's documentation:
Environment Variables, Runtime Policies, Benchmarks, Image Assurance

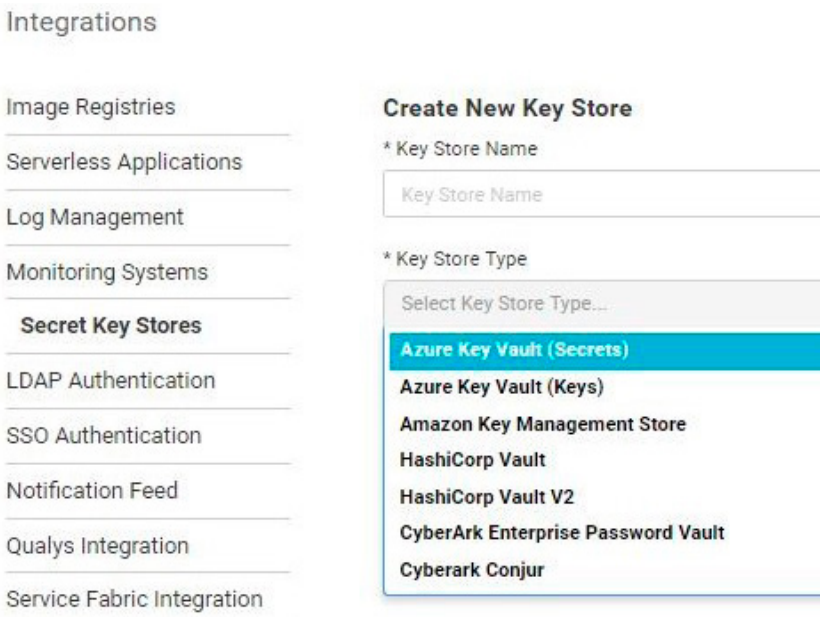
Requirement 3

Protect stored cardholder data.
Protection methods such as encryption, truncation, masking, and hashing are critical components of cardholder data protection. If an intruder circumvents other security controls and gains access to encrypted data without the proper cryptographic keys, the data is unreadable and unusable to that person.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
3.6.2	Secure cryptographic key distribution	Uses secrets management that securely injects secrets into running containers without container downtime/restart.

Requirement 3.6.2

Aqua provides central management and secure distribution of secrets and cryptographic keys into running containers with no container downtime/restart. Admins can define a secret in the Aqua Management console and assign access control policies that authorize users or groups to run containers that make use of the secret. Aqua integrates with several secret stores, including HashiCorp Vault, Amazon KMS, Azure Vault, and CyberArk, thereby allowing organizations to leverage these central stores and extend them for use with containers.



For more information on how to comply with Requirement 3, review the following topics in Aqua's documentation: Integration of Secret Key Stores, Manage Secrets

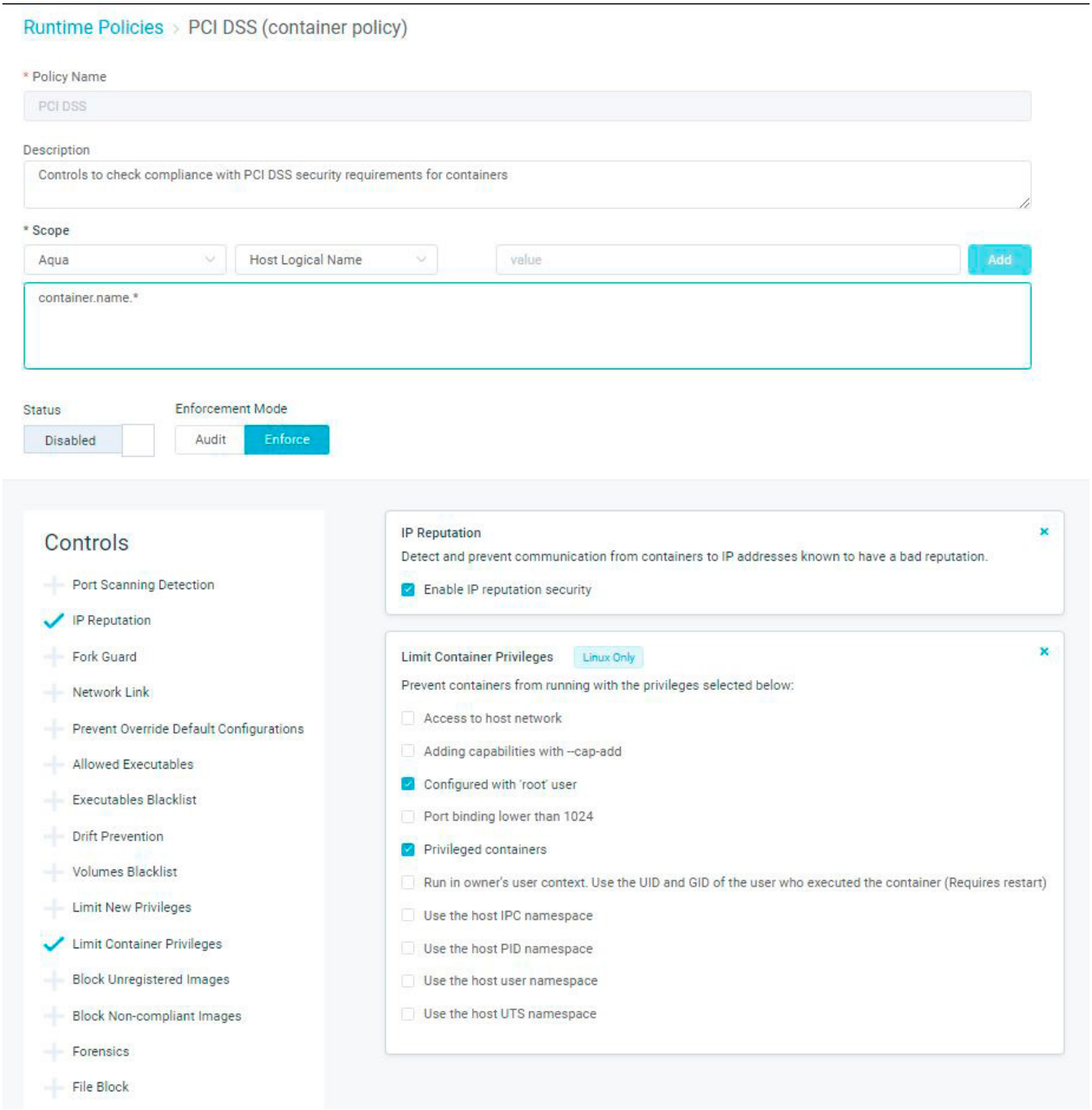
Requirement 5

Protect all systems against malware and regularly update anti-virus software or programs.
Malicious software, commonly referred to as “malware” - including viruses, worms, and Trojans - enters the network during many business-approved activities including employee e-mail and use of the Internet, mobile computers, and storage devices, resulting in the exploitation of system vulnerabilities

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
5.1	Deploy anti-virus software on all systems commonly affected by malicious software (particularly personal computers and servers).	Aqua’s Runtime Policies protection monitors running containers for zero-day attacks by monitoring suspicious behavior in the container file system, managing the communication to banned IP addresses.
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.	RBAC Runtime Policies allow the admin to restrict access or modify permission to unauthorized personnel.

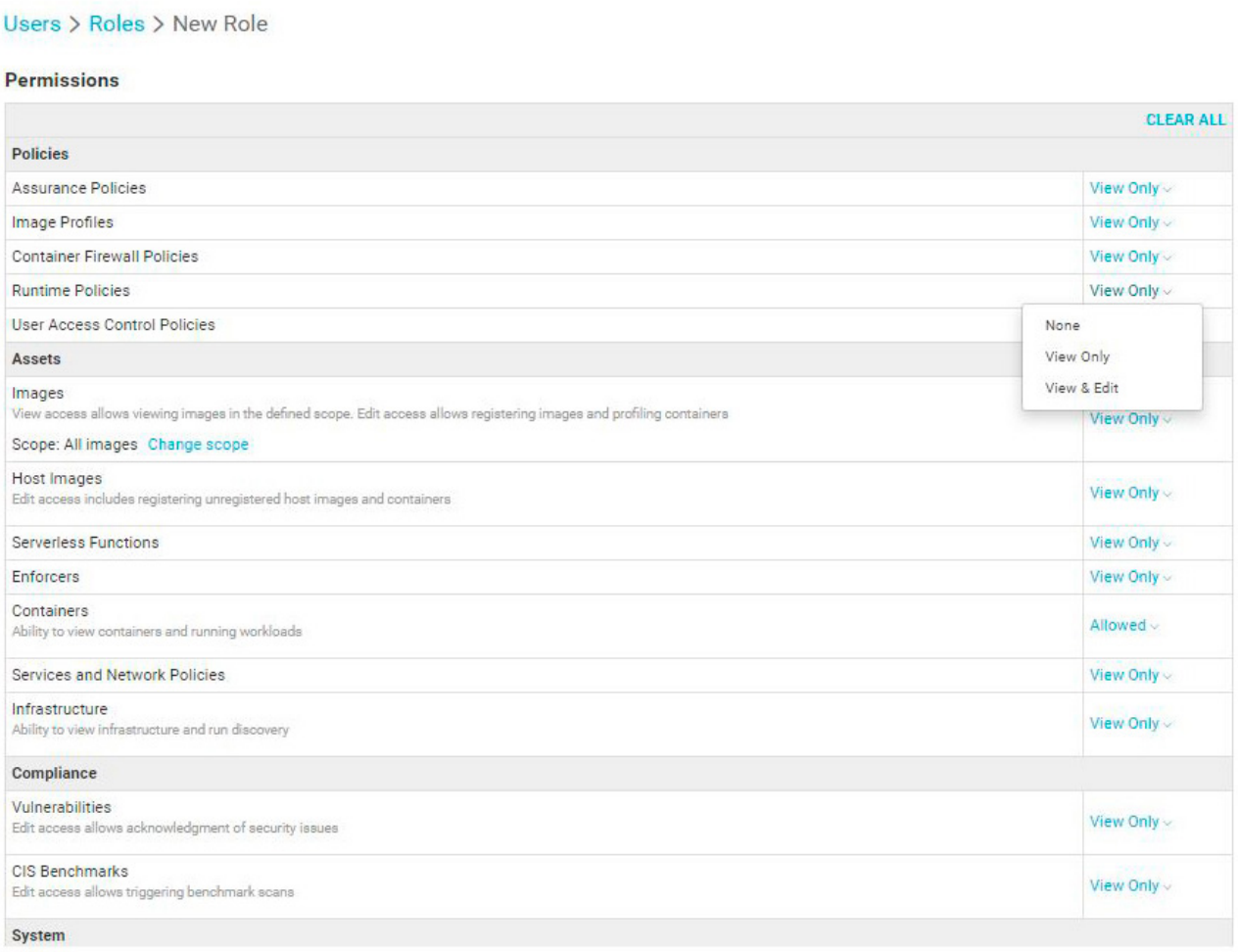
Requirement 5.1

Aqua's Build-in Runtime policy check compliance with PCI-DSS security requirements for containers. Other security measurements can be applied based on the company's security requirements.



Requirement 5.3

With Aqua's RBAC Roles Runtime policies, the admin can restrict access or modify permission to designated Runtime polices.



For more information on how to comply with Requirement 5, review the following topics in Aqua's documentation: Runtime Policies Overview, Users and Roles

Requirement 6

Develop and maintain secure systems and applications.
All systems must have all appropriate software patches to protect them against the exploitation and compromise of cardholder data by malicious individuals and malicious software.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
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.	Provides image and function scanning for vulnerabilities, malware, configuration errors, embedded secrets, vulnerability severity ranking, and CVSS scores.
6.4.1	Separate development/test environments from production environments and enforce the separation with access controls.	Provides label-based management combined with RBAC.
6.4.2	Separation of duties between development/test and production environments	
6.5.6	All “high risk” vulnerabilities identified in the vulnerability identification process (as defined in PCI-DSS Requirement 6.1).	Performs image/function scanning and severity ranking, and continuous monitoring of running containers for newly discovered vulnerabilities.

Requirement 6.1

Aqua scans images, functions, and repositories for known vulnerabilities, configuration errors, embedded secrets, and malware that could impact the system information assurance and security. Each vulnerability is ranked from low to high based on its Common Vulnerability Scoring System (CVSS). The scanning process is performed by integrating the Aqua Command Center with an image/function registry. In addition, Aqua can scan images within CI tools (such as Jenkins, Microsoft VSTS, and Bamboo) to mitigate risks during the build.
Each image is scanned for vulnerabilities in both its OS packages and in the development language files. All new and old identified vulnerabilities are audited and can be automatically mitigated by creating image assurance policies. Aqua also providers actionable mitigation information for detected vulnerabilities for fast remediation.
Aqua is constantly updating new vulnerabilities, culled from several sources (commercial, public, and proprietary) with the Aqua CyberCenter feed for continuous improvement. It checks and rechecks registries and images for ongoing monitoring.

Requirement 6.4

By using Aqua, organizations can separate their container development from their production deployments in an easy way. Organizations can integrate Aqua into the development pipeline by adding Aqua as a build step in CI/CD tools such as Jenkins and TeamCity. This also ensures that developers are not exposed to Aqua's console while they continue working with their existing tools. Aqua admins can also assign labels to images and create a security policy that allows only images with specific labels to enter production. For example, if a user tries to run a container with a "development" label on a production host, the container will not be allowed to run.

Requirement 6.4.2

Aqua's labels-based management can be used to label hosts as dev/test/production. Different RBAC and container policies can be applied by these labels so that specific users can only access resources with specific labels.
Furthermore, Aqua provides access control and real-time privileged activity monitoring and enforcement for Docker and Kubernetes administrative operations by determining which user can access and perform specific activities.

Requirement 6.5.6

As stated in section 6.1 - Admins can scan images, functions, registries, and images on hosts for known vulnerabilities, malware, secrets, and configuration errors that could impact the system information assurance and security. Each vulnerability is ranked from low to high, based on its Common Vulnerability Scoring System (CVSS). In addition, all running containers are continuously monitored and matched against the vulnerability database feed in order to flag any container with a newly identified vulnerability.
The Aqua admins can create custom security policies for image assurance. For example, an admin can block any image with a high severity vulnerability, or prevent images, with an average score higher than X from running.

For more information on how to comply with Requirement 6, review the following topics in Aqua's documentation:
Aqua Labels, Role-Based User Access Control, Aqua CyberCenter, Image Assurance

Requirement 7

Restrict access to cardholder data by business need to know.
To ensure critical data can only be accessed by authorized personnel, systems and processes must be in place to limit access based on the need to know and according to job responsibilities.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
7.1	Limit access to system components and cardholder data to only those individuals whose job requires such access:	Container-level RBAC, Aqua system roles and integration with Active Directory / LDAP; Aqua can assign and enforce what the user can do.
7.1.2	Restrict access to privileged user IDs to least privileges necessary to perform job responsibilities.	
7.1.3	Assign access based on individual personnel's job classification and function	
7.2.2	Assignment of privileges to individuals based on job classification and function.	

Requirements 7.1, 7.1.2, 7.1.3 and 7.2.2

Aqua provides RBAC for administrative container operations by determining which user can access specific container resources.

Aqua can also automatically create an image security profile based on container activity, which Aqua tracks and analyzes. The profiler analyzes and reports on the security profile by noting vital component usage, resources, and network settings. This will create the least privileges security runtime profile and the Aqua admin can then edit and save it.

To further extend RBAC, admins can use the user access control policies to prevent containers from running as root and/or allow the container to run with specific users.

Aqua integrates with the organization's identity management systems such as Active Directory / LDAP, to map container users to the organizational user groups, and SAML for single sign-on. This allows even more granular access control and separation of duties.

For more information on how to comply with Requirement 7, review the following topics in Aqua's documentation: Role-Based User Access Control, User Authentication

Requirement 8

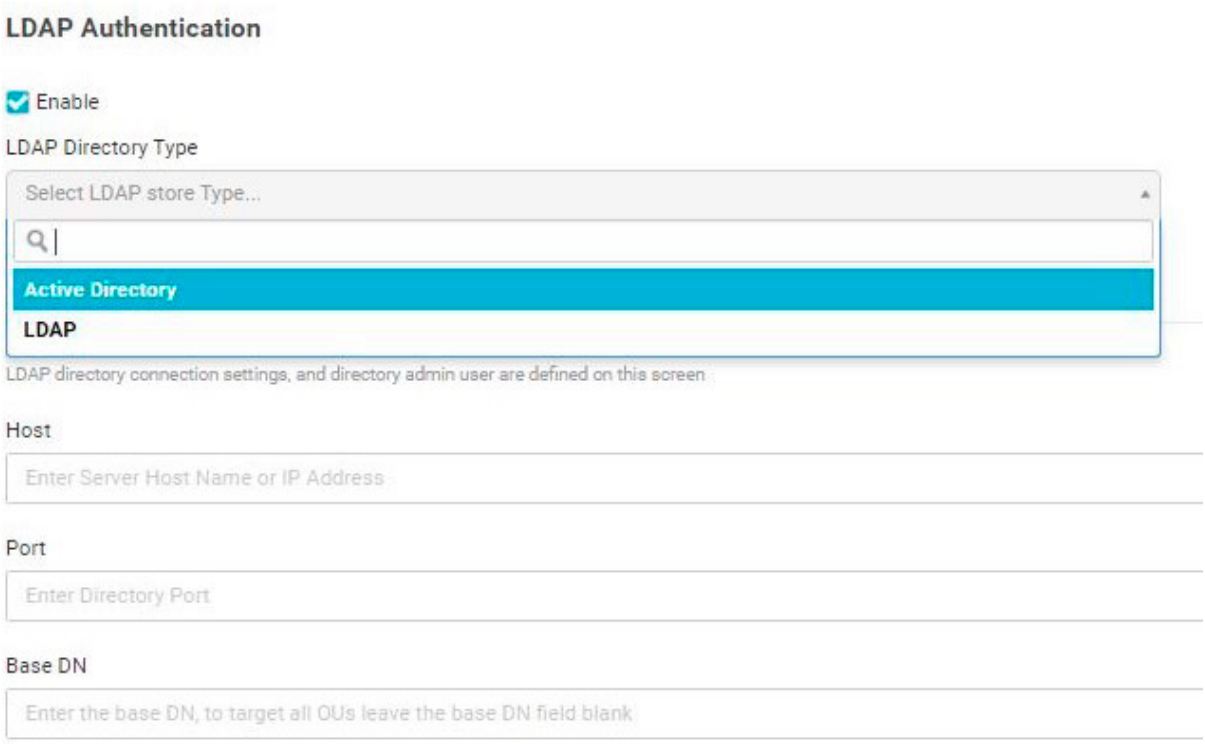
Identify and authenticate access to system components.
Assigning a unique identification (ID) to each person with access ensures that each individual is uniquely accountable for their actions. When such accountability is in place, actions taken on critical data and systems can be traced to known and authorized users and processes.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
8.1.1	Assign all users a unique ID before allowing them to access system components or cardholder data	Container-level RBAC, Aqua system roles and integration with Active Directory / LDAP; Aqua can assign and enforce what users can do.
8.2.1	Using strong cryptography, render all authentication credentials (such as passwords/ phrases) unreadable during transmission and storage on all system components	Secrets management that securely injects secrets into a running container with no container downtime/restart.

Requirement 8.1.1

Aqua provides RBAC for administrative container operations by determining which user can access specific container resources and what the user can do.

Aqua integrates with the organization's identity management systems such as Active Directory / LDAP, to map container users to the organizational user groups and SAML for single sign-on. This allows even more granular access control and separation of duties.



Requirement 8.2.1

Aqua securely delivers secrets to runtime containers in memory with no persistence on disk. Secrets can be rotated, updated, and revoked with no container downtime or restart, managed by existing third-party enterprise secrets vaults, such as HashiCorp Vault, CyberArk Password Vault, CyberArk Conjur, AWS KMS, and Azure Vaults.

For more information on how to comply with Requirement 8, review the following topics in Aqua's documentation: Role-Based User Access Control, User Authentication, Integration of Secret Key Stores, Manage Secrets

Requirement 10

Track and monitor all access to network resources and cardholder data.
Logging mechanisms and the ability to track user activities are critical in preventing, detecting, or minimizing the impact of a data compromise. The presence of logs in all environments allows thorough tracking, alerting, and analysis when something does go wrong.

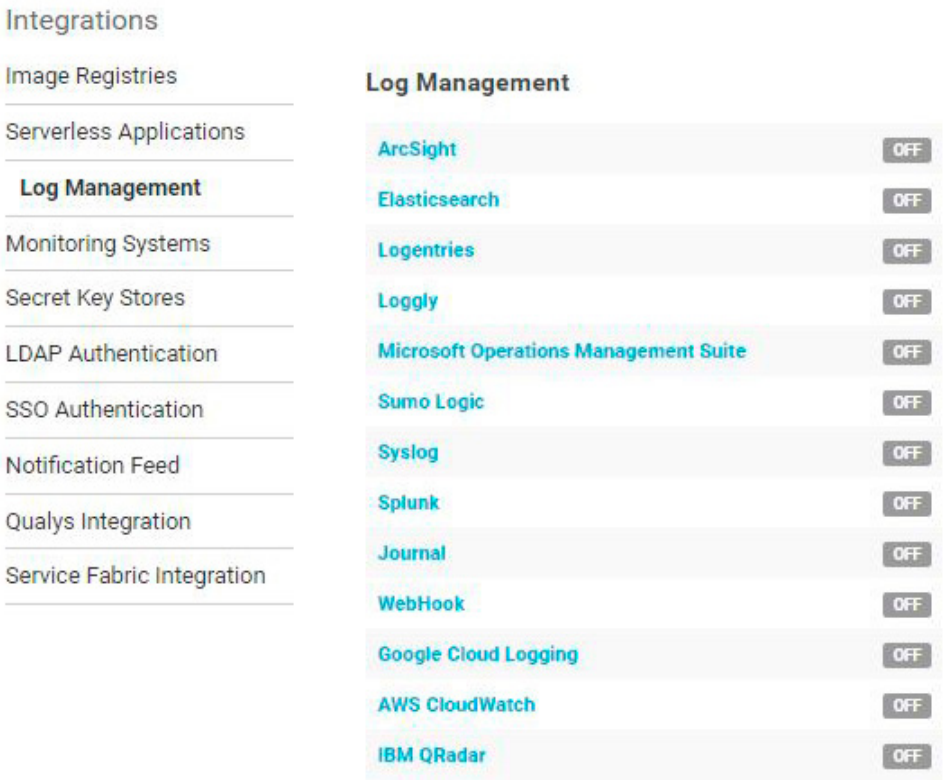
PCI-DSS Requirement		Aqua Feature Addressing the Requirement
10.1	Implement audit trails to link all access to system components to each individual user	Container-level events are logged to establish an audit trail, including named user traceability., integration with 3rd party SIEM / analytics
10.2	Implement automated audit trails for all system components to reconstruct the following events:	Container-level events are logged to establish an audit trail, including named user traceability. All events can be exported to 3rd party SIEM/analytics tools such as Splunk, ArcSight, LogRhythm, etc.
10.2.2	All actions were taken by any individual with root or administrative privileges	
10.3	Record at least the following audit trail entries for all system components for each event:	Container-level events are logged to establish an audit trail, including named user traceability. All events can be exported to 3rd party SIEM/analytics tools such as Splunk, ArcSight, LogRhythm, etc.
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	
10.5.2	Protect audit trail files from unauthorized modifications	Use Aqua's access control and FIM protection in order to ensure that the current audit trail files are protected from unauthorized modifications via access control mechanisms, physical segregation, and/or network segregation.

Requirement 10.1, 10.2 and 10.2.2

Aqua generates granular audit trails of all access activity, scan events and coverage, Linux, Docker and Kubernetes commands, container activity, secrets activity, and system events. In addition, it provides full user accountability and controlled super-user permissions.
Aqua admin can use pre-built alerts and reports for key compliance mandates (PCI, GDPR, HIPAA, NIST SP 800-190). Aqua also provides automated CIS Linux, CIS Docker, and CIS Kubernetes benchmark reports. Furthermore, Aqua tracks changes in vulnerability status and provides timeliness of scan and remediation trends.

Requirement 10.3

Aqua's Audit feature enables admins to view audit events from the Aqua management console. All audit messages generated by Aqua Enforcer are sent to the Aqua Command Center and are available from the console's audit screen. Admins can also configure forwarding of events to external SIEM and analytics tools, such as ArcSight, AWS CloudWatch, Datadog, ElasticSearch, Google CSCC, Logentries, Loggly, Microsoft OMS, Splunk, Sumologic, and Syslog. The audit display results appear in table format per time, audit type, and provides a brief description.



For more information on to how to comply with Requirement 10, review the following topics in Aqua's documentation: Role-Based User Access Control, User Authentication, External Log Collectors, Forward audit events to external log collectors

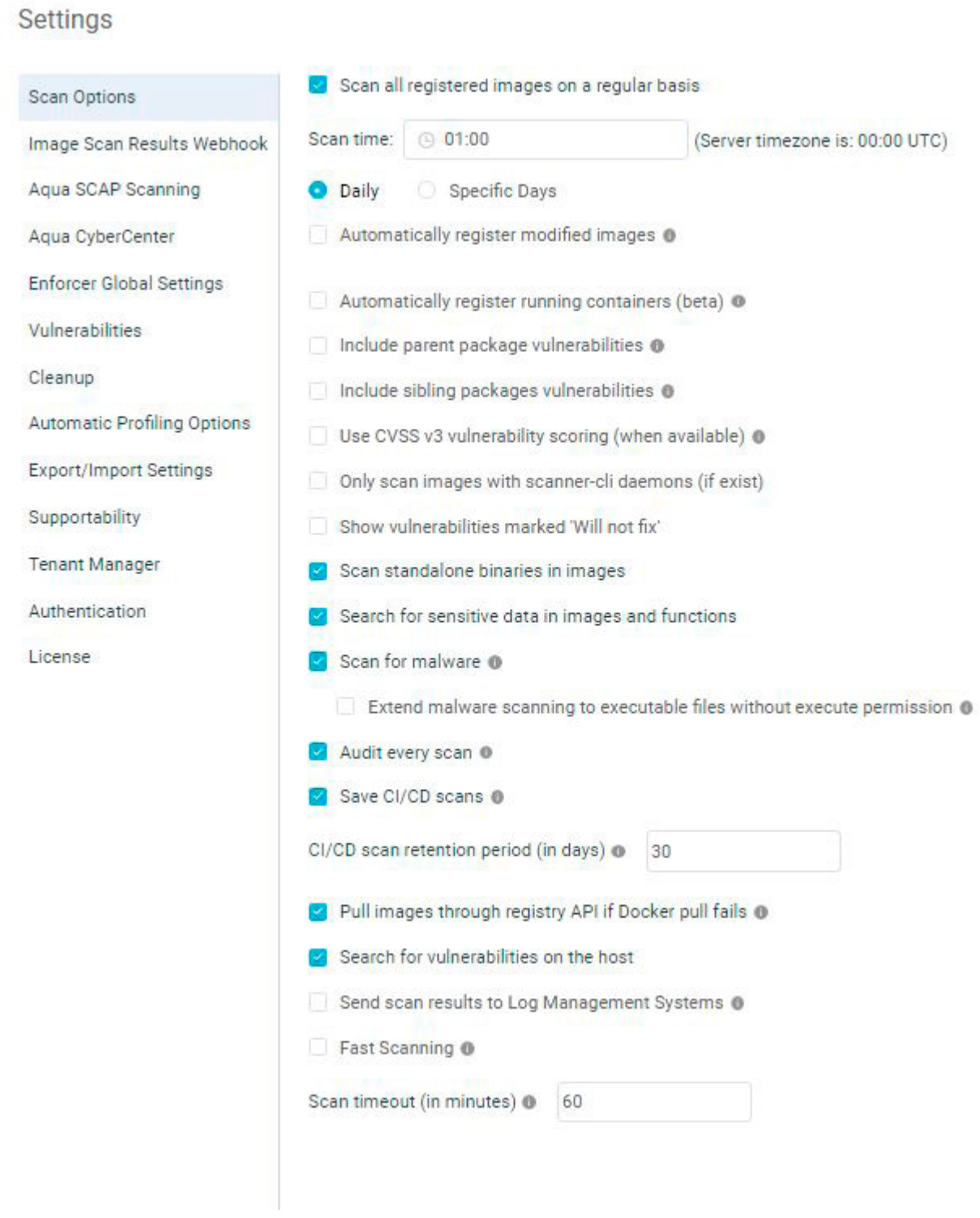
Requirement 11

Regularly test security systems and processes.
System components, processes, and custom software should be tested frequently to ensure security controls continue to reflect a changing environment.

PCI-DSS Requirement		Aqua Feature Addressing the Requirement
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.	Aqua allows the user to schedule a periodic scan or an ad hoc scan of all/specific/ configured images in his environment for CVEs matched with Aqua's Cyber Intelligence which is continuously updating itself with newly discovered CVEs.
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.	Aqua's runtime policies protection monitors running containers for Zero-Day attacks by monitoring suspicious behavior in the container file system, managing the communication to banned IP addresses.
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.	Tracking and monitoring of container images, as well as allowed executables within images. Encrypting images in the build to protect sensitive data and IP.

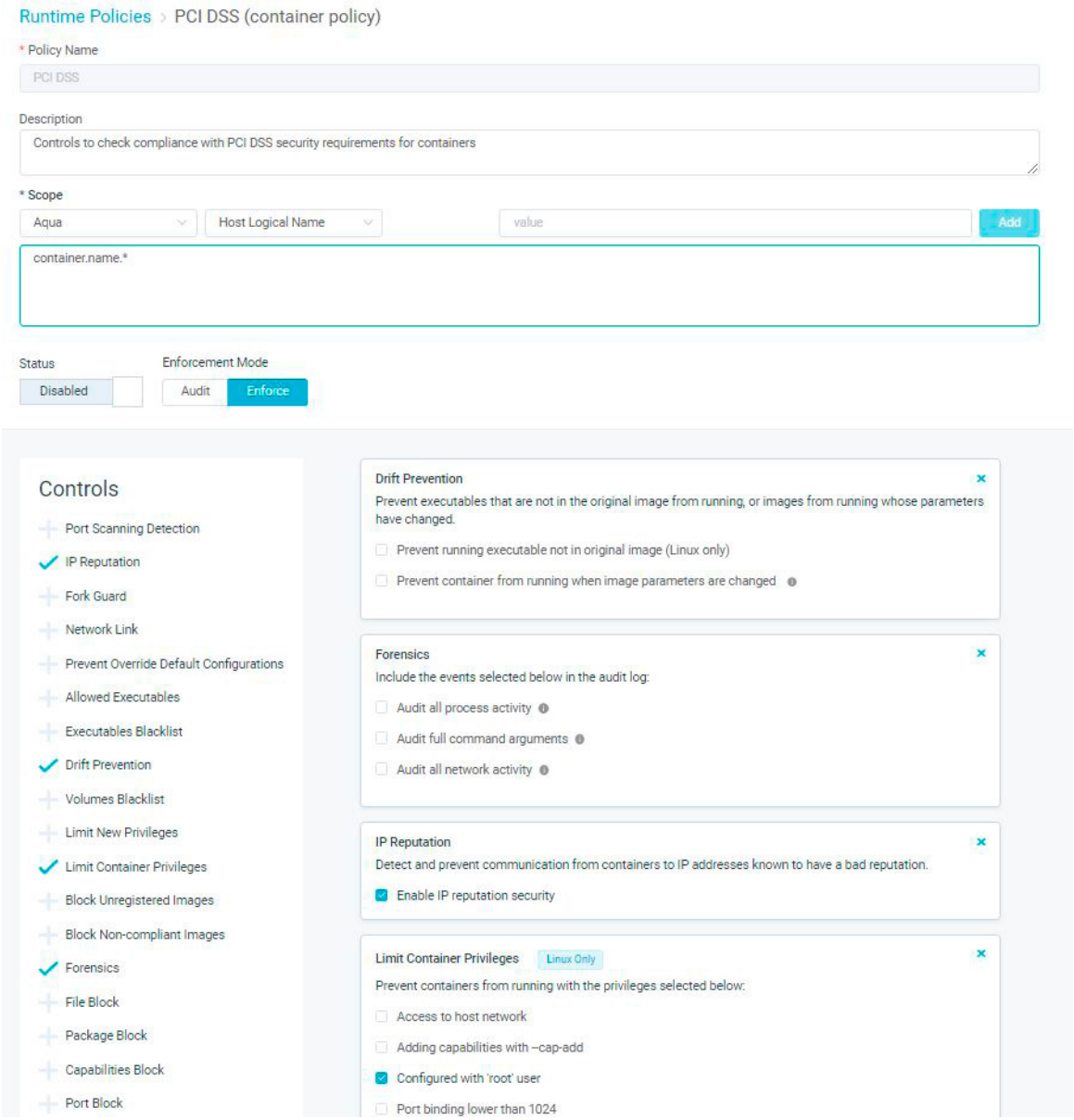
Requirement 11.2.1

Configures Aqua to scan all registered images on a regular basis (for example, nightly). This will ensure that your images are scanned at least daily for the latest security issues.



Requirement 11.4

Audits all container activity and blocks any prohibited activity in runtime. It configures one or more container runtime policies to restrict the runtime activities of containers, according to the security requirements of your organization. It monitors or blocks any suspicious behavior in the container file system and manages communication to and from the container.



Requirement 11.5

Aqua's image Runtime policy enforces container immutability and detects any unapproved changes to running containers by continuously comparing them to their originating images, including executables, privilege elevation, and image parameters. Protects high-value information, such as sensitive data and intellectual property. The Aqua MicroEnforcer encrypts images during the build process. Once a container is instantiated, the Aqua MicroEnforcer decrypts the image in runtime**.**

For more information on how to comply with Requirement 11, review the following topics in Aqua's documentation: Configure Scan Options, Drift Prevention, Runtime Policies Overview

contact@aquasec.com
aquasec.com
@AquaSecTeam
AquaSecTeam

Aqua Security helps enterprises secure their cloud native, container-based and serverless applications from development to production. Aqua bridges the gap between DevOps and security, promoting business agility and accelerating digital transformation. Aqua's Cloud Native Security Platform provides full visibility and security automation across the entire application lifecycle, using a modern zero-touch approach to detect and prevent threats while simplifying regulatory compliance. Aqua customers include some of the world's largest financial services, software development, internet, media, hospitality, and retail companies, with implementations across the globe spanning a broad range of cloud providers and on-premise technologies.

version 11-2019