COMPLIANCE

Payment Card Industry Data Security Standard (PCI-DSS) Compliance with Rocket® API

The Payment Card Industry requires all organizations that store or process credit card data and transactions to implement technical security requirements over all systems involved in data storage and transmission. The scope of these control requirements ranges from encryption methods to access rights management to vulnerability testing.

Rocket® API is designed with the robust security features you need to implement many PCI-DSS requirements for processing Cardholder Data (CHD) securely, from encryption to authentication to data masking. However, the scope of PCI-DSS extends to a number of organizational and procedural control areas that cannot be satisfied solely through technical means. PCI-DSS compliance will ultimately depend on effective implementation of the technical controls available through Rocket API as well as appropriate technical and procedural controls over your entire CHD environment. Relevant PCI-DSS requirements, and the capabilities Rocket API offers to address them, are listed below.

PCI-DSS Requirements	Rocket API Capabilities
1.3 Prohibit direct public access between the Internet and any system component in the cardholder data environment.	Rocket API provides functionality for gateways between the back-end mainframe and the internet, preventing direct access to the CHD environment.
2.3 Encrypt all non-console administrative access using strong cryptography.	All data transfers using Rocket API are secured through encrypted protocols, including TLS1.2 and SSHv2. Strong encryption provides for data security and confidentiality.
3.1 Keep cardholder data storage to a minimum by implementing data retention and disposal policies, procedures and processes that include at least the following for all cardholder data (CHD) storage: • Limiting data storage amount and retention time to that which is required for legal, regulatory, and business requirements • Processes for secure deletion of data when no longer needed • Specific retention requirements for cardholder data • A quarterly process for identifying and securely deleting stored cardholder data that exceeds defined retention.	Rocket API does not by default store any data involved with API calls, limiting the storage of such data. Customers are able to cache common API calls for performance reasons. This cached data is retained in memory only, not written to any permanent storage mechanism, and is erased when the Rocket API service is stopped, or at preconfigured time intervals. Additionally, Rocket API supports data masking and anonymization capabilities to limit the exposure of data in a personally identifiable form
3.3 Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed).	Rocket API supports data masking and anonymization.
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.	All data transfers using Rocket API are secured through encrypted protocols, including TLS1.2 and SSHv2. Strong encryption provides for data security and confidentiality.

PCI-DSS Requirements	Rocket API Capabilities
6.3 Develop internal and external software applications (including web-based administrative access to applications) securely, as follows: • In accordance with PCI DSS (for example, secure authentication and logging) • Based on industry standards and/or best practices. • Incorporating information security throughout the software-development life cycle • Remove development, test and/or custom application accounts, user IDs, and passwords before applications become active or are released to customers. • Review custom code prior to release to production or customers in order to identify any potential coding vulnerability (using either manual or automated processes) • Code changes are reviewed by individuals other than the originating code author, and by individuals knowledgeable about code-review techniques and secure coding practices. • Code reviews ensure code is developed according to secure coding guidelines • Appropriate corrections are implemented prior to release. • Code-review results are reviewed and approved by management prior to release.	Any changes to the coding of an API must be processed through your source control system before being deployed through Rocket API, and are thereby subject to all the development and change controls that you have implemented in your SDLC environment.
6.4 Follow change control processes and procedures for all changes to system components. The processes must include the following: • Separate development/test environments from production environments, and enforce the separation with access controls • Separation of duties between development/test and production environments • Production data (live PANs) are not used for testing or development • Removal of test data and accounts before production systems become active • Change control procedures must include documentation of impact and change approval by authorized parties • Functionality testing to verify that the change does not adversely impact the security of the system • Back-out procedures • Upon completion of a significant change, all relevant PCI DSS requirements must be implemented on all new or changed systems and networks, and documentation updated as applicable	Any changes to the coding of an API must be processed through your source control system before being deployed through Rocket API, and are thereby subject to all the development and change controls that you have implemented in your SDLC environment.

PCI-DSS Requirements	Rocket API Capabilities
7.1 Limit access to system components and cardholder data to only those individuals whose job requires such access. Define access needs for each role, including system components and data resources that each role needs to access for their job function and level of privilege required (for example, user, administrator, etc.) for accessing resources. Restrict access to privileged user IDs to least privileges necessary to perform job responsibilities. Assign access based on individual personnel's job classification and function. Require documented approval by authorized parties specifying required privileges.	Rocket API leverages access credentials from the back-end mainframe operating system, and thereby inherits all access rights and restrictions associated with those credentials, including read and write capabilities. In addition the access rights controlled through the back-end mainframe, Rocket API provides application-layer security that can further restrict API calls by user, by function, and by data being accessed. The Rocket Access and Connectivity Hub (RACH) management interface, which manages the inventory of APIs and deployment to API gateways, enforces granular user access controls that are configurable by each customer. RACH uses LDAP authentication to leverage the password controls and other mechanisms that authenticate your users. RACH audit logging records all user activity within the application—including uploading and deployment of compiled APIs as well as administration of the application itself—providing individual accountability for all access and activity.
7.2 Establish an access control system for systems components with multiple users that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed.	RACH allows for granular assignment of permissions to support the rule of least privilege, and does not allow access to functions unless explicitly granted. Access for execution of an API leverages access credentials from the back-end mainframe operating system, and thereby inherits all access rights and restrictions associated with those credentials, including read and write capabilities for specific data elements.

PCI-DSS Requirements	Rocket API Capabilities
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: • Assign all users a unique ID before allowing them to access system components or cardholder data. • Control addition, deletion, and modification of user IDs, credentials, and other identifier objects. • Immediately revoke access for any terminated users. • Remove/disable inactive user accounts at least every 90 days. • Manage IDs used by vendors to access, support, or maintain system components via remote access • Limit repeated access attempts by locking out the user ID after not more than six attempts. • Set the lockout duration to a minimum of 30 minutes or until administrator enables the user ID. • If a session has been idle for more than 15 minutes, require the user to re-authenticate to re-activate the terminal or session.	RACH uses LDAP authentication to leverage the password controls and other mechanisms that authenticate your users. Account management and authentication controls applied to your Identity and Access Management (IAM) system will apply to Rocket API.
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 • Something you are, such as a biometric.	RACH uses LDAP authentication to leverage the password controls and other mechanisms that authenticate your users. This can include a password as well as other authentication factors enforced by your IAM.
8.7 All access to any database containing cardholder data (including access by applications, administrators, and all other users) is restricted as follows: • All user access to, user queries of, and user actions on databases are through programmatic methods. • Only database administrators have the ability to directly access or query databases. • Application IDs for database applications can only be used by the applications (and not by individual users or other non-application processes).	Rocket API provides the programmatic method for reading and writing to databases with CHD. All such access is fully logged.
10.1 Implement audit trails to link all access to system components to each individual user.	All logs for Rocket API, including the RACH management platform, are tied to the individual user executing the command.

PCI-DSS Requirements	Rocket API Capabilities
10.2 Implement automated audit trails for all system components to reconstruct the following events: • All individual accesses to cardholder data • All actions taken by any individual with root or administrative privileges • Access to all audit trails • Invalid logical access attempts • Use of identification and authentication mechanisms — including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges • Initialization, stopping, or pausing of the audit logs • Creation and deletion of system-level objects	Audit logging functionality can record all API calls, showing details of the user accessing the function, data being accessed, and data values being read and/or written. RACH audit logging records all user activity within the application—including uploading and deployment of compiled APIs as well as administration of the application itself—providing individual accountability for all access and activity.
10.3 Record at least the following audit trail entries for all system components for each event: • User identification • Type of event • Date and time • Success or failure indication • Origination of event • Identity or name of affected data, system component, or resource.	API logs include all relevant details of each recorded event.

Rocket

rocketsoftware.com
info@rocketsoftware.com
US: 1 877 577 4323
EMEA: 0800 520 0439
APAC: 1800 823 405
twitter.com/rocket
www.linkedin.com/company/rocket-software
www.facebook.com/RocketSoftwareInc
blog.rocketsoftware.com