Chargent
PAYMENTS FOR SALESFORCE®

Salesforce PCI Guide

870 Market Street, Suite 809 | San Francisco | +1-415-275-1115 | AppFrontier.com

Salesforce PCI Guide

with Chargent by AppFrontier

Salesforce and PCI Compliance

Payment Card Industry, Data Security Standard (PCI DSS) are a set of security standards for merchants who accept credit cards, both online and offline. They are not a governmental regulation or law, but rather guidelines for keeping your customer's payment card data secure developed by the PCI Security Standards Council. Enforcement of PCI compliance for merchants, as well as any penalties for non-compliance, is managed by the individual payment companies.

To achieve PCI DSS compliance, follow the 12 requirements in the standard, working with your acquiring bank and the card brands you do business with. PCI compliance is an ongoing process of adhering to security policies, procedures, software design, network architecture and other security measures designed to make sure your customer's data is being kept secure.

Salesforce's PCI certification as a data storage service provider means that much of the PCI scope for Salesforce customers has already been dealt with. Since AppFrontier and the Chargent Payment Processing application never store data outside of Salesforce, Salesforce's PCI certification covers your data while it is at rest in the Salesforce database.

Salesforce customers who must adhere to PCI compliance may store Cardholder Data in Salesforce: Primary Account Number ("PAN" or "credit card number" or "Bank Account Number"), Cardholder Name, and Expiration Date, provided that they are stored using the proper security as detailed below in the Chargent PCI Implementation Guide.

Achieving PCI Compliance — 12 Requirement Areas

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

Your Organization's PCI Compliance

Use of Salesforce or the Chargent Payment Processing for Salesforce add-on application does not transfer PCI compliance to your company. Your organization may need to achieve and maintain its own PCI compliance depending on your use cases.

While the appropriate use of Salesforce and Chargent are a key foundation for achieving PCI compliance, ultimately your organization is responsible for all other security processes and handling of credit card data that you collect.

Chargent PCI Implementation Guide

Salesforce already handles much of your PCI Compliance scope with its advanced security features, but Chargent includes a number of capabilities to further reduce PCI scope. For a demonstration of any of these features, please contact us.

Encrypted Custom Fields

For PCI Compliance, Primary Account Numbers (PANs) may only be stored in Encrypted Custom Fields (ECFs) in Salesforce. PANs should not be stored in clear text fields, attached files, or any other location.

Encrypted custom fields are Salesforce text fields that can contain any letters, numbers or symbols but are encrypted. Examples of encrypted custom fields in the Chargent Payment Processing for Salesforce application include Expiration Month, Expiration Year, Bank Account Number, Credit Card Number, plus Merchant ID and Merchant Security Key in the gateway configuration.

Salesforce users see the data in Encrypted fields as a series of asterisks, but they can still edit or delete the data stored in those fields. For identification purposes, the last 4 digits of the credit card number field are shown in plain text.

Making encrypted fields visible in plain text by granting Salesforce user profiles the "View Encrypted Data" permission is not recommended. If you need to have certain user profiles that can view the encrypted fields in clear text rather than masked with asterisks, either on a permanent or temporarily enabled basis, you should have a clear written policy regarding how and why this is done.

Data that Should NOT Be Stored in Salesforce

If you store any of the following payment data in Salesforce, even in an Encrypted Field, your use of Salesforce will be non-compliant with PCI.

- Full magnetic stripe/chip
- PINs/PIN blocks
- CAV2/CVC2/CVV/CID (3 or 4 digit card security code)

Some customers may wish to use the 3 or 4 digit security code (CVV2, etc.) for initial transactions. In that case, we recommend that you configure your payment gateway to not require the security code in order to approve transactions. Create a process, either manual or using the Chargent gateway field provided for this purpose, to remove the card security code so you do not store this after you run the first charge.

Credit Card Handling Field

Each Chargent Payment Gateway record includes a field called Credit Card Handling. This field allows you to choose if/how credit card data is stored in different scenarios. For PCI Compliance or liability reasons, many customers do not wish to store credit card data. The options are as follows:

- **Never Clear:** Chargent will not remove any card data automatically.
- **Clear After Successful Charge:** Chargent will clear the credit card number, expiration dates and card security code only after a successful charge is run (recommended).
- **Clear After All Transactions:** The credit card number, expiration date and card security code will be erased after any transaction (Charge, Void, Refund)
- **Clear When Token Present:** Only when a token is present in the token field, will the credit card number, expiration date and card security code be cleared.

Chargent Gateway Object

The Chargent Gateway object should generally only be available to the administrator of your Salesforce account or a finance person. During normal operation of Chargent Payment Processing for Salesforce, you should give read-only permissions to the Gateway object and associated tab, fields, etc. to any Salesforce users who would not normally have access to your payment accounts.

Create, Edit and Delete permissions on the Gateway should be given only to System Administrators and/or finance profiles. You can remove the Gateway field from page layouts if you only have one active gateway, since in that case the single active gateway will automatically be chosen for any transactions.

Chargent Payment Console

Chargent Payment Processing for Salesforce's Payment Console feature allows you to submit payments directly from a Salesforce modal (popup) window to your payment gateway. In addition to being a customizable, convenient interface for call center agents, customer service, billing or sales teams, it is able to initiate payments, receive tokens back from the payment gateway, and create transaction records in Salesforce.

Most importantly, with the Payment Console feature, transactions and tokens are created without ever saving or storing cardholder account number information in Salesforce. (Please note that the Chargent Payment Console feature requires a Chargent Platform edition license. Please contact us for details.)

Tokenization

Another security feature to consider using is tokenization. Tokenization lowers your liability and the scope of a PCI audit, as you won't be storing account numbers in Salesforce. The downside of tokenization is that you lose some control over your data, as tokens are unique to the payment gateway provider, and cannot be moved between payment providers.

Tokenization is not a Salesforce security feature, but is a payment gateway technology supported by Chargent in many of our payment gateway integrations. Your payment gateway / processor will store the credit card data for you, and give you back a unique token which you then store in Salesforce to use for any future transactions against that credit card.

Tokenization provides an additional level of security, since the token or surrogate value represents an account number that is now stored encrypted in an external system, rather than stored encrypted in your Salesforce org. In addition to reducing PCI compliance scope, tokenization helps satisfy a number of PCI requirements -- the mandate to store payment data in the minimum number of locations, as well as the requirement that access keys be stored securely in as few places as possible.

In credit card tokenization, the last 4 digits of the credit card are preserved for identification purposes. So your staff will still be able to identify the correct card to a customer with the token stored in Salesforce, no differently than with the encrypted credit card number field, which shows asterisks except for the last 4 digits.

Maintaining PCI scope reduction with and without tokenization

The primary difference between these two approaches is that without tokenization a user may need to input Credit Card information every time a transaction is to be processed. This is due to Chargent not storing the credit card number and card security code in the Salesforce database. (Note that this does require proper configuration of Chargent.)

Salesforce T&Cs do allow for customers to store card numbers and bank account numbers in encrypted fields. When using this model, there are some options to maintain PCI compliance while enabling users to use the payment data on multiple occasions. We recommend working with a Qualified Security Advisor to best understand your options here.

When Chargent is configured to use tokenization - the PAN data is stored at the gateway, so charges can be done without storing cardholder data in the Salesforce database.

Other Salesforce Security Features

Salesforce is designed to be highly secure, but also to deliver flexibility for you to implement your own security design to meet the needs of your organization. Salesforce therefore has a large number of security features that Salesforce System Administrators need to configure to support their organization's PCI controls. Here are several of the more common ones, but for complete details we recommend reviewing Salesforce's security documentation directly, especially the Security Implementation Guide.

Password Rules

Salesforce gives you a number of capabilities to control the security of your users passwords, such as the level of complexity required for your passwords, and specifying an amount of time before users' passwords expire. Salesforce's default security already addresses a portion PCI compliance requirement 2, since there are no default passwords. Initial passwords are always uniquely generated and there are default minimums for password length and character requirements.

Sharing Model

Choosing which data in Salesforce your users or groups / types of users have access to is one of the most important aspects of data security. Salesforce delivers more in this area than perhaps any other system of its kind, with a layered approach to security where you have a large amount of flexibility and the ability to control data access on a very granular level. There are 3 basic areas that control data access in Salesforce:

- Permission Sets and Profiles control the objects and features that users can access
- Field-Level Security and profile page layouts control the fields that users can access
- Organization-Wide sharing settings, role hierarchy, and sharing rules all control the individual records that users can access
- Object Level Security (User Licenses and Profiles)

For the Chargent Payment Processing for Salesforce application, you must first assign a Chargent license to each user, in order for them to be able to see the Chargent fields and transaction records. System Administrators can control this by finding the Chargent Transaction package under Setup > Installed Packages and clicking the "Modify" link.

After that, Profiles are probably the most important aspect of controlling the security around Chargent's fields and any access to the Encrypted Fields. Each user is assigned a profile, which is typically related to their job function (eg. System Administrator, Sales, Billing, etc.). The profile can prevent a user from seeing, modifying, creating or deleting any record of a particular type of object (such as Opportunities, Transactions, or Chargent Orders).

In addition to profiles, Permission Sets are useful in granting additional access settings to particular users beyond the profile, because multiple permission sets can be granted to a single user, but each user can have only one profile.

Field-Level Security

Field-Level Security lets you protect certain fields on an object, without hiding the whole object from users. It is an important security feature of Salesforce, because even though you can use Page Layouts to hide certain fields from users based on their Profile, the fields will still be accessible in list views, reports, search and elsewhere if the user has permissions on the object that the field is part of. Field-Level Security can ensure that certain sensitive fields are completely protected from being viewed, edited, modified or deleted by users who you do not wish to have access to them.

Record-Level Security

Salesforce offers a variety of ways you can control who has what level of access to which records, including Organization-Wide sharing settings, role hierarchy, and Sharing Rules. Record-level Security in Salesforce tends to be more about separating data access for different business units or levels in an organization, and less about securely configuring Chargent Payment Processing for Salesforce and ensuring PCI compliance, so please refer to Salesforce's security documentation to learn more.

Field History Tracking

AppFrontier recommends enabling field history on a number of Chargent fields, to provide an audit trail of what values have been changed in those key fields, and by whom. You may also wish to enable field history on certain fields in other objects such as Accounts and Contacts.

When Field History is enabled on a particular field, modifying that field adds an entry to the History related list on that object. All history entries include the time, date, what was changed, and which Salesforce user made the change. Note that up to 20 fields per object can be tracked, and some types of fields such as formulas and roll-up summaries cannot have history tracking enabled.

Find recommended Chargent fields for turning on history tracking in our documentation.

Frequently Asked Questions about Chargent

What are Chargent's data retention policies?

- It's up to each Chargent customer to decide and configure when data should be removed from their Salesforce organization. The Chargent application is not responsible for Salesforce data or retention policies.
- All the operations with sensitive data (PAN (CC#), card security code) work in real time (3-30 seconds "in memory", this is dependent upon Salesforce's internet connection speed at the time of transmission and gateway response time at the time of transmission).
- PCI protected data is not being stored in the Salesforce database, provided that Chargent's PCI configuration guide was followed correctly.

What are AppFrontier's access control mechanisms?

- The AppFrontier (the makers of Chargent) team will not require any access to your company's Salesforce organization.
- Your company may decide that for support reasons, granting access to the AppFrontier support team is necessary. In this case, access to the Salesforce Organization with Chargent packages installed can be granted from Salesforce organization, by your company's Salesforce users. Chargent support personnel will access the system with the same level of access as the user who granted them access.
- Access to the Chargent application for your company's Salesforce users is configured inside Salesforce like access to any other element of Salesforce is defined. We recommend consulting with your Salesforce administrator for more information and to collaborate on ensuring your company's access policies are maintained in Salesforce for the Chargent application.

How is Chargent managed once it is deployed?

- The Chargent app will be managed as part of your Salesforce system. It will be managed just like any other Salesforce app that your company uses today.
- All configurations and maintenance will be completed by your company's administrator, your company's developers and any consultants you might hire.
- The AppFrontier support team can be called upon to answer questions.

Why is Chargent not listed on PCI-DSS or PA-DSS site?

- The PCI-DSS & PA-DSS require that the application provider, the service provider or the merchant hosting credit card data has both a network and servers, which can be quarantined into a lab for full review and scanning. As Chargent is intellectual property, namely code that is installed into Salesforce servers and transmitted over the Salesforce network, there is no way for AppFrontier to have these scans completed by an ASV. QSAs have reviewed the AppFrontier architecture and decided that current PCI standards do not include advanced models like the Salesforce AppExchange deployed applications.
- Due to this, Chargent customers treat Chargent as a bespoke application when including Chargent in their PCI scope.

We're Always Here to Help!

Our team at Chargent can provide you with more information, a demo, or documentation. We also offer a 30-day free trial of Chargent. Contact us to learn more.

Additional Resources

PCI Security Standards Council

PCI DSS Self-Assessment Questionnaire (SAQ)
(Many merchants and service providers can self-evaluate compliance with PCI using the SAQ. Please consult your acquiring bank for details regarding your particular PCI DSS validation requirements.)

Payment Card Industry (PCI) Data Security Standard
Requirements and Security Assessment Procedures, Version 3.0, November 2013

Salesforce Security Implementation Guide

Salesforce Security, Privacy and Architecture Documentation

Six Ways to Reduce PCI DSS Audit Scope by Tokenizing Cardholder Data