Magento Payment Bridge ver. 1.0.0.0

PA-DSS Implementation Guide

Version 1.0.0.0

Approval Date: 07/02/2010

Magento™ PaymentBridge

Table of Contents

Notice

THE INFORMATION IN THIS DOCUMENT IS FOR INFORMATIONAL PURPOSES ONLY. MAGENTO INC. MAKES NO REPRESENTATION OR WARRANTY AS TO THE ACCURACY OR THE COMPLETENESS OF THE INFORMATION CONTAINED HEREIN. YOU ACKNOWLEDGE AND AGREE THAT THIS INFORMATION IS PROVIDED TO YOU ON THE CONDITION THAT NEITHER MAGENTO INC. NOR ANY OF ITS AFFILIATES OR REPRESENTATIVES WILL HAVE ANY LIABILITY IN RESPECT OF, OR AS A RESULT OF, THE USE OF THIS INFORMATION. IN ADDITION, YOU ACKNOWLEDGE AND AGREE THAT YOU ARE SOLELY RESPONSIBLE FOR MAKING YOUR OWN DECISIONS BASED ON THE INFORMATION HEREIN.

Nothing herein shall be construed as limiting or reducing your obligations to comply with any applicable laws, regulations or industry standards relating to security or otherwise including, but not limited to, PA-DSS and PCI-DSS.

The retailer may undertake activities that may affect compliance. For this reason, Magento Inc. is required to be specific to only the standard software provided by it.

About this Document

This document describes the steps that must be followed in order for your Magento Payment Bridge installations to comply with Payment Application – Data Security Standards (PA-DSS). The information in this document is based on PCI Security Standards Council Payment Application Data Security Standards program (version 1.2 dated October, 2008).

Magento Inc. instructs and advises its customers, resellers, and system integrators to deploy Magento Inc. applications in a manner that adheres to the PCI Data Security Standard (v1.2). Subsequent to this, best practices and hardening methods, such as those referenced by the Center for Internet Security (CIS) and their various "Benchmarks", should be followed in order to enhance system logging, reduce the chance of intrusion and increase the ability to detect intrusion, as well as other general recommendations to secure networking environments. Such methods include, but are not limited to, enabling operating system auditing subsystems, system logging of individual servers to a centralized logging server, the disabling of infrequently-used or frequently vulnerable networking protocols and the implementation of certificate-based protocols for access to servers by users and vendors.

If you do not follow the steps outlined here, your Magento Payment Bridge installations will not be PA-DSS compliant.

Revision History

Revision	Date	Name	Title	Summary of Changes
Revision 1.0	03/20/2010	Michael Bessolov	Director of Technology	Initial Release
Revision 1.1	05/28/2010	Michael Bessolov	Director of Technology	Minor changes
Revision 1.2	06/01/2010	Michael Bessolov	Director of Technology	Minor changes
Revision 1.3	06/09/2010	Michael Bessolov	Director of Technology	Minor changes
Revision 1.4	06/25/2010	Michael Bessolov	Director of Technology	Minor changes
Revision 1.5	07/02/2010	Michael Bessolov	Director of Technology	Minor changes

Note: This PA-DSS Implementation Guide must be reviewed on a yearly basis, whenever the underlying application changes or whenever the PA-DSS requirements change. Updates should be tracked and reasonable accommodations should be made to distribute or make the updated guide available to users. Magento Inc. will distribute the Implementation Guide to customers, resellers, and integrators via the Magento Inc. support website (https://support.varien.com/).

Executive Summary

Magento Payment Bridge version 1.0.0.0 has been PA-DSS (Payment Application Data Security Standard) certified, with PA-DSS Version 1.2. For the PA-DSS assessment, we worked with the following PCI SSC approved Payment Application Qualified Security Assessor (PAQSA):

Coalfire Systems, Inc. 361 Centennial Parkway Suite 150 Louisville, CO 80027	Coalfire Systems, Inc. 150 Nickerson Street Suite 106 Seattle, WA 98109

This document also explains the Payment Card Industry (PCI) initiative and the Payment Application Data Security Standard (PA-DSS) guidelines. The document then provides specific installation, configuration, and ongoing management best practices for using the Payment Application as a PA-DSS validated Application operating in a PCI Compliant environment.

PCI Security Standards Council Reference Documents

The following documents provide additional detail surrounding the PCI SSC and related security programs (PA-DSS, PCI DSS, etc):

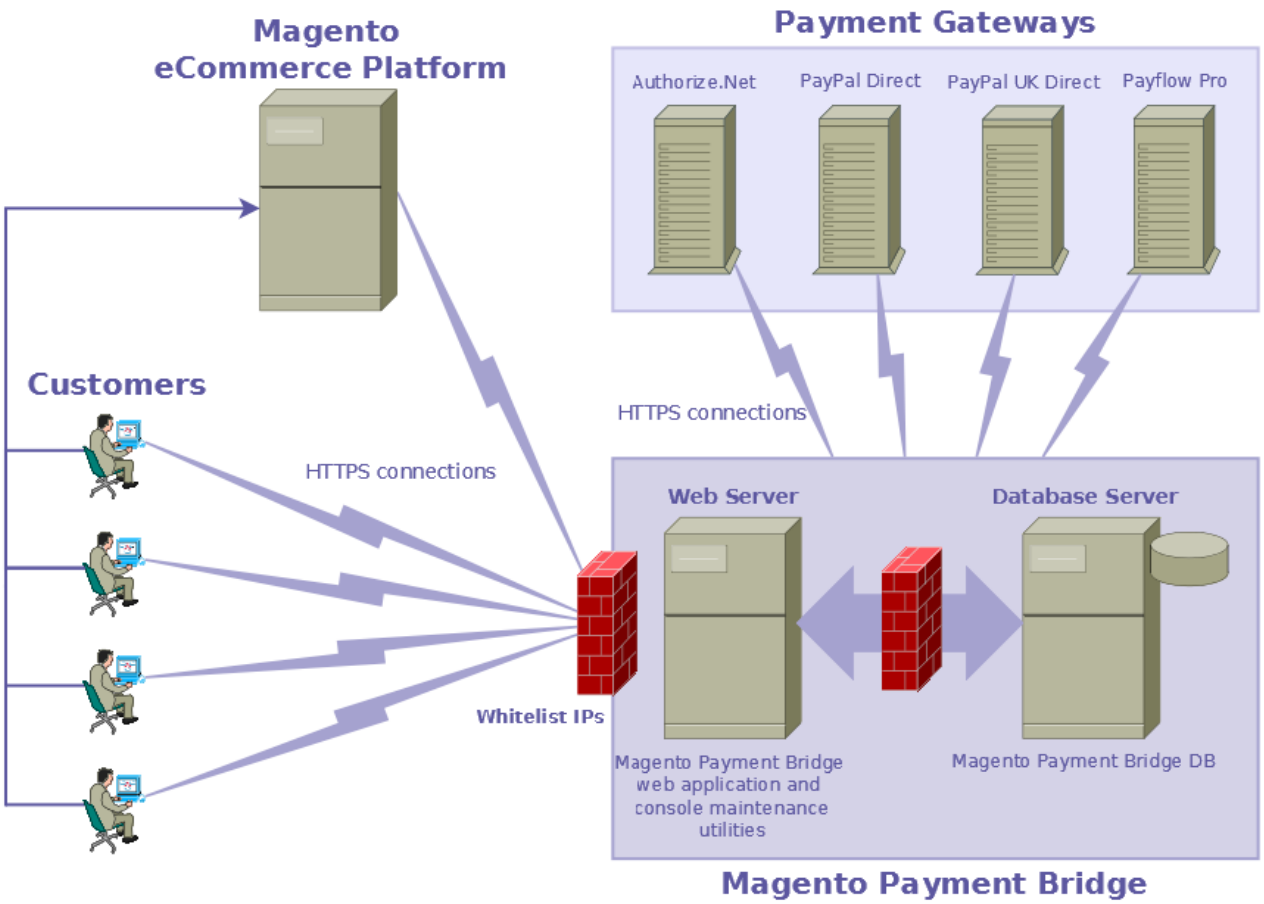
- Payment Applications Data Security Standard (PA-DSS)
  https://www.pcisecuritystandards.org/security_standards/pa_dss.shtml
- Payment Card Industry Data Security Standard (PCI DSS)
  https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml
- Open Web Application Security Project (OWASP)
  http://www.owasp.org

Application Summary

Name:	Magento Payment Bridge
Application Version Number(s):	1.0
Operating System(s):	Latest supported Linux distributive based on RedHat Enterprise Linux (RHEL), x86/x86_64
Database Software:	MySQL 5.0 or higher
Components of Application Suite(i.e. POS, Back Office, etc):	• Magento Payment Bridge web application v. 1.0 – payment application which works as a secure payment gateway between a merchant's store and payment providers, such as Authorize.Net, PayPal, etc. • Magento Payment Bridge DB v. 1.0 – a storage for all Magento Payment Bridge web application data, such as payment sessions and configuration parameters for merchants and their payment provider accounts, etc. • Magento Payment Bridge console maintenance utilities v. 1.0 – utilities used for administering Magento Payment Bridge web application, for example for adding/removing/changing merchant data.
Other Required Payment Application Software:	PayPal API version 60.0, Authorize.Net API 3.1
Other Required Third Party Software:	PHP 5.2.0 or higher; required PHP extensions: mcrypt (2.5.6 or higher), curl (7.10.5 or higher, in order to meet PHP 5.x.x requirements), base64_, json_, iconv, hash, PDO_MySQL, ionCube.
Setup:	Manual installation
Application Description:	Payment Bridge (PB) is a standalone application that can process payments and can serve as a wrapper between an Online Store (OS) and a payment gateway. The main point is to avoid saving customer payment information in the online store, instead it uses Payment Bridge tokens only and no cardholder data is stored. Payment Bridge is configured according to payment methods it supports. It stores access credentials to the gateway and customers payment information including credit card numbers at the time when a payment is not processed.

Application Environment:	• Payment Bridge is a web-based application running on a separate web server (under Apache HTTP Server) from the Magento eCommerce platform. • Database storage is located on a separate server. • All high security data is stored with encryption.
Application Target Clientele:	Retailers and Manufacturers looking for a flexible eCommerce platform to support the growth of their online channel.
Description of Versioning Methodology:	The version schema is as follows: X.Y.Z.P ▪ X - Major version number - Major version changes, which include adding and removing features and functionality. These releases will provide an upgrade path to allow users to upgrade, and will have minimum backwards compatibility to previous major versions. Changes at this level will have an impact on PA-DSS requirements. ▪ Y - Minor version number - Minor version changes, which include the adding of new features and bug fixes. Upgrades should be straight forward. The release should have maximum backwards compatibility to previous minor versions and changes at this level may or may not have an impact on PA-DSS requirements. ▪ Z - Revision version number - Bug fixes and minimal new features that do not affect PA-DSS requirements. Upgrades should be straight forward. The release should be fully backwards compatible to the current minor version. ▪ P - Patch version number - Urgent bug and/or security fixes that do not affect PA-DSS requirements. Upgrades should be straight forward. The release should be fully backwards compatible to the current minor version.

Typical Network Implementation

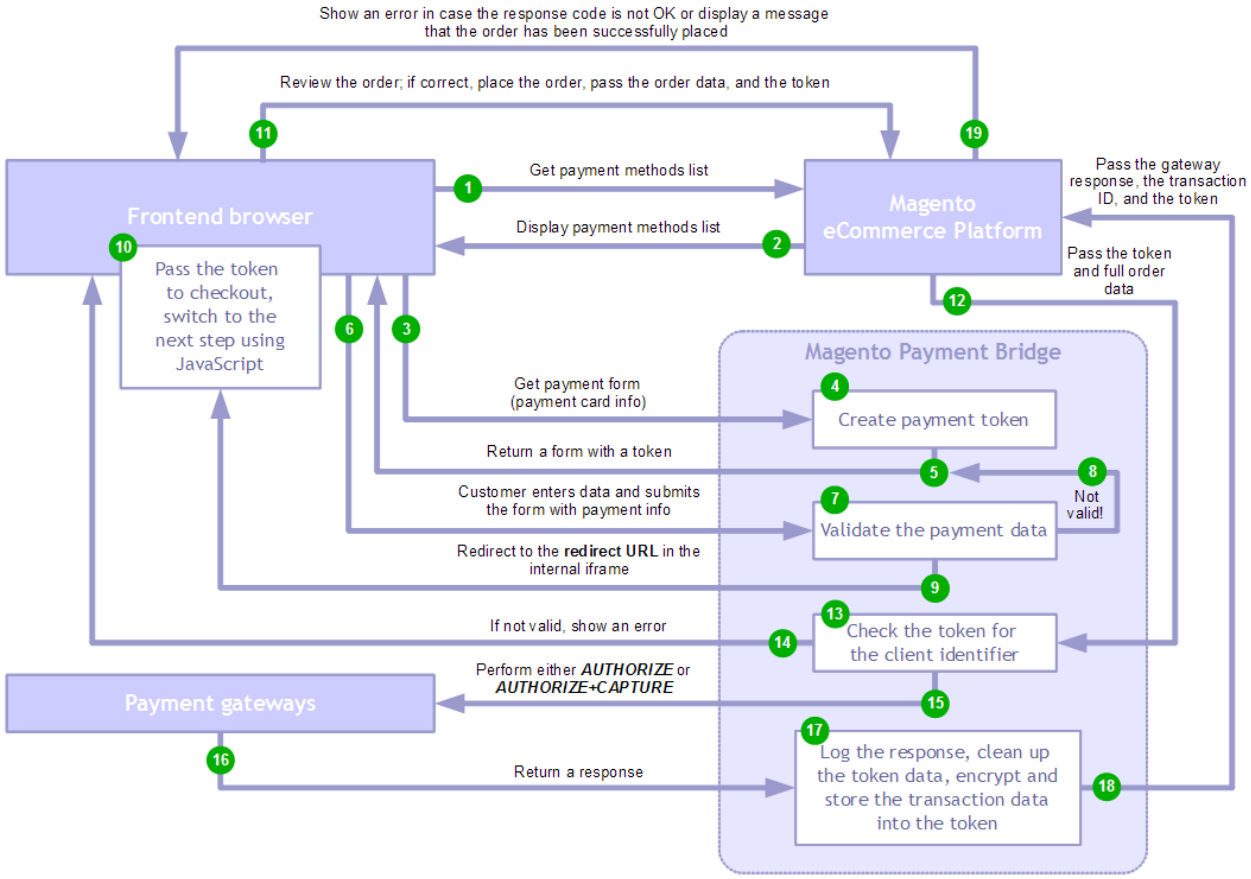


Notes:

- Customers interact with the Magento eCommerce Platform by using their web browser to access Magento-driven eCommerce websites.
- The checkout process initiated in the Magento eCommerce Platform communicates with Magento Payment Bridge via HTTPS protocol to access a payment form for the specified payment method along with the generated token.
- Only the Magento eCommerce Platform instances registered to eligible merchants can work with Magento Payment Bridge.
- Magento Payment Bridge uses two separate servers. One of them processes web requests and runs internal logic, the other one processes the Payment Bridge database.
- All Cardholder data and passwords are stored encrypted in the database.
- Magento Payment Bridge serves as a proxy between Magento eCommerce Platform instances and corresponding payment gateways supported by Magento Payment Bridge.
- Magento Payment Bridge interacts with the payment gateways via HTTPS protocol using the corresponding gateways' APIs.
- Magento Payment Bridge interacts with customers via HTTPS protocol.

- It is highly recommended to use three different servers for installing Magento eCommerce Platform, Magento Payment Bridge, and Magento Payment Bridge DB instances.

Dataflow Diagram



Notes:

- All steps are performed sequentially in the numerical order shown in the diagram.
- All connections between Frontend browser, Magento eCommerce Platform, Magento Payment Bridge, and Payment gateways are secured (https).

Difference between PCI Compliance and PA-DSS Validation

As a software vendor, it is our responsibility to be "PA-DSS Validated."

We have performed an assessment and certification compliance review with an independent assessment firm to ensure that our platform conforms to industry best practices when handling, managing and storing payment related information.

PA-DSS is the standard against which the Payment Application has been tested, assessed, and validated.

PCI Compliance is later obtained by the merchant, and is an assessment of the merchant's actual server (or hosting) environment.

Obtaining "PCI Compliance" is the responsibility of the merchant and your hosting provider, working together, using PCI compliant server architecture, with proper hardware & software configurations and access control procedures. The PA-DSS Validation is intended to ensure that the Payment Application will help you achieve and maintain PCI Compliance with respect to how the Payment Application handles user accounts, passwords, encryption, and other payment data related information.

The Payment Card Industry (PCI) has developed security standards for handling cardholder information in a published standard called the PCI Data Security Standard (DSS). The security requirements defined in the DSS apply to all members, merchants, and service providers that store, process or transmit cardholder data.

The PCI DSS requirements apply to all system components within the payment application environment, which is defined as any network device, host, or application included in, or connected to, a network segment where cardholder data is stored, processed or transmitted.

The 12 Requirements of the PCI DSS:

Build and Maintain a Secure Network
1. Install and maintain a firewall configuration to protect data
2. Do not use vendor-supplied defaults for system passwords and other security parameters

Protect Cardholder Data
3. Protect Stored Data
4. Encrypt transmission of cardholder data and sensitive information across public networks

Maintain a Vulnerability Management Program
5. Use and regularly update anti-virus software
6. Develop and maintain secure systems and applications

Implement Strong Access Control Measures
7. Restrict access to data by business need-to-know
8. Assign a unique ID to each person with computer access
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

Considerations for the Implementation of Payment Application in a PCI-Compliant Environment

The following areas must be considered for proper implementation in a PCI-Compliant environment.

- Sensitive Credit Card Data requires special handling
- Remove Historical Credit Card Data
- Set up Good Access Controls
- Properly Train and Monitor Admin Personnel
- Key Management Roles & Responsibilities
- PCI-Compliant Remote Access
- Use SSH, VPN, or SSL/TLS for encryption of administrative access
- Log settings must be compliant
- PCI-Compliant Wireless settings
- Data Transport Encryption
- PCI-Compliant Use of Email
- Network Segmentation
- Never store cardholder data on internet-accessible systems
- Use SSL for Secure Data Transmission
- Delivery of Updates in a PCI Compliant Fashion

Remove Historical Credit Card Data (PA-DSS 1.1.4.a)

This is the first version of Magento Payment Bridge, and therefore, previous versions of Magento Payment Bridge did not store sensitive authentication data. Therefore, there is no need for secure removal of this historical data by the application as required by PA-DSS v1.2.

Sensitive Authentication Data requires special handling (PA-DSS 1.1.5.c)

The following guidelines must be followed when dealing with sensitive authentication data (swipe data, validation values or codes, PIN or PIN block data):

- Collect sensitive authentication data only when needed to solve a specific problem
- Store such data only in specific, known locations with limited access
- Collect only the limited amount of data needed to solve a specific problem
- Encrypt sensitive authentication data while stored
- Securely delete such data immediately after use

Purging of Card Data (PA-DSS 2.1.a)

The following guidelines must be taken into account when dealing with card data (PAN alone or with any of the following: expiry date, cardholder name, or service code):

- A customer defined retention period must be defined with a business justification.

- Card data exceeding the customer-defined retention period must be purged.
- The card data exceeding the customer-defined retention period is automatically purged by the system from the ***token*** database table. After this operation, card data cannot be restored anymore.
- Card data is stored in the ***token*** table only temporarily and is cleaned up immediately after the first successful payment gateway transaction.

Removal of Cryptographic material (PA-DSS 2.7.a)

Previous versions of Magento Payment Bridge never used encryption and, therefore, there is no cryptographic data to be securely removed as required by PA-DSS v1.2.

Set up Good Access Controls (3.1.c and 3.2)

The PCI DSS requires that access to all systems in the payment processing environment be protected through use of unique users and complex passwords. Unique user accounts indicate that every account used is associated with an individual user and/or process with no use of generic group accounts used by more than one user or process. The following instructions should be taken into account:

- Do not use administrative accounts for application logins (e.g., don't use the superuser account for application access to the database).
- Assign strong passwords to these default accounts (even if they won't be used), and then disable or do not use the accounts.
- Assign strong application and system passwords whenever possible.
- Create PCI DSS-compliant complex passwords to access the payment application, per PCI Data Security Standard 8.5.8 through 8.5.15

The PCI standard requires the following password complexity for compliance (often referred to as using "strong passwords"):

- Do not use group, shared, or generic user accounts (8.5.8)
- Passwords must be changed at least every 90 days (8.5.9)
- Passwords must be at least 7 characters (8.5.10)
- Passwords must include both numeric and alphabetic characters (8.5.11)
- New passwords cannot be the same as the last 4 passwords (8.5.12)

PCI user account requirements beyond uniqueness and password complexity are listed below:

- If an incorrect password is provided 6 times the account should be locked out (8.5.13)
- Account lock out duration should be at least 30 min. (or until an administrator resets it) (8.5.14)
- Sessions idle for more than 15 minutes should require re-entry of username and password to reactivate the session (8.5.15)

These same account and password criteria must also be applied to any applications or databases included in payment processing to be PCI compliant. Magento Payment Bridge, as tested to in our PA-DSS audit, meets, or exceeds these requirements.

You must control access via unique username and PCI DSS-compliant complex passwords to any PCs or servers with payment applications and to databases storing cardholder data.

Properly Train and Monitor Admin Personnel

It is your responsibility to institute proper personnel management techniques for allowing admin user access to credit cards, site data, etc. You can control whether each individual admin user can see credit card PAN (or only last 4).

In most systems, a security breach is the result of unethical personnel. Therefore, pay special attention to whom you trust into your admin site and who you allow to view fully decrypted and unmasked payment information.

Key Management Roles & Responsibilities (PA-DSS 2.5)

Magento Payment Bridge uses 256-bit AES encryption for important data transfer and storage. CBC mode is used when an initialization vector is prepent to the encryption value.

32-byte encryption keys must be generated and applied to the stored data by the Admin User who uses console scripts in the following way:

- To regenerate data stored in the database with a new encryption key, run php <system_dir>/tools/crypt.php -r and enter a new encryption key. The encryption key consists of two parts which must be entered by two different Magento Payment Bridge administrators.
- To regenerate a transfer key for the merchant, run php <system_dir>/tools/merchant.php --ek <merchant_code>, where <merchant_code> is the merchant code of a particular merchant.

PCI compliance guidelines for security procedures related to key management procedures should be taken into account by your Magento Payment Bridge administrator(s).

Magento Payment Bridge administrator must control the merchant access information and transfer key by updating it at least once per year.

If a merchant finds out that his or her access information or the transfer key has been stolen, the Magento Payment Bridge administrator must be informed about this issue immediately. In this case, the Magento Payment Bridge administrator must provide new access information or a transfer key to the merchant.

For detailed instructions on how to manage the merchant access information and transfer key, please refer to the Magento Payment Bridge Administration Guide.

Log settings must be compliant (PA-DSS 4.2.b)

Magento Payment Bridge customer must have logging turned on and configured according to the Magento Payment Bridge Administration Guide instructions which comply with PCI DSS.

Note: To be compliant with PCI-DSS, logging should be configured per PCI DSS 10.2 and 10.3 as follows:

Implement automated assessment trails for all system components to reconstruct the following events:

10.2.1. All individual user accesses to cardholder data
10.2.2. All actions taken by any individual with root or administrative privileges
10.2.3. Access to all assessment trails
10.2.4. Invalid logical access attempts
10.2.5. Use of identification and authentication mechanisms
10.2.6. Initialization of the assessment logs
10.2.7. Creation and deletion of system-level objects.

While events 10.2.1 through 10.2.3 and 10.2.7 must be set up according to the Magento Payment Bridge Administration Guide, events 10.2.4 through 10.2.7 are enabled by default.

Record at least the following assessment trail entries for all system components for each event from 10.2.x:

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.

Disabling or subverting the logging function set up according to the Magento Payment Bridge Administration Guide instructions in any way will result in non-compliance with PCI DSS.

Also, Magento Payment Bridge has its own internal logging. Every log record is put into a corresponding file under the directory <app_base_dir>/var/log/<current_date>, where <app_base_dir> is an absolute path to the Magento Payment Bridge application base directory. This is done in the following way:

- Every action performed by Magento Payment Bridge is stored in the access.log.
- All application errors, warnings, and notices are stored in the error.log.
- Operations performed by PayPal Instant Payment Notification requests are stored in the ipn.log.

PCI-Compliant Wireless settings (PA-DSS 6.1.b and 6.2.b)

Magento Payment Bridge does not require or support wireless technologies. However, should the merchant implement wireless access within the cardholder data environment, the following guidelines for secure wireless settings must be followed per PCI Data Security Standard 1.2.3, 2.1.1 and 4.1.1:

1.2.3: Perimeter firewalls must be installed between any wireless networks and systems that store cardholder data, and these firewalls must deny or control (if such traffic is necessary for business purposes) any traffic from the wireless environment into the cardholder data environment.

2.1.1:

- All wireless networks implement strong encryption (e.g. AES)
- Encryption keys were changed from default at installation, and are changed anytime anyone with knowledge of the keys leaves the company or changes positions
- Default SNMP community strings on wireless devices were changed
- Default passwords/passphrases on access points were changed
- Firmware on wireless devices is updated to support strong encryption for authentication and transmission over wireless networks (for example, WPA/WPA2)
- Other security-related wireless vendor defaults, if applicable

4.1.1:

- Industry best practices are used to implement strong encryption for the following over the wireless network in the cardholder data environment (4.1.1):
    - Transmission of cardholder data
    - Transmission of authentication data
- Payment applications using wireless technology must facilitate the following regarding use of WEP:
- For new wireless implementations, it is prohibited to implement WEP as of March 31, 2009.
- For current wireless implementations, it is prohibited to use WEP after June 30, 2010.

Never store cardholder data on internet-accessible systems (PA-DSS 9.1.b)

Never store cardholder data on Internet-accessible systems (e.g., web server and database server must not be on same server.)

PCI-Compliant Delivery of Updates (PA-DSS 10.1)

As a development company, we keep abreast of the relevant security concerns and vulnerabilities in our area of development and expertise. We do this by subscribing to relevant data feeds and news services which inform us of potential security issues

Once we identify a relevant vulnerability, we work to develop and test a patch that helps protect Magento Payment Bridge against the specific, new vulnerability. We attempt to publish a patch within 10 working days of the identification of the vulnerability. We will then contact vendors and dealers and inform them about the patch availability and that they must install it in order to stay PCI compliant. Typically, merchants are expected to respond quickly to and install available patches within 30 days.

We do not deliver software and/or updates via remote access to customer networks. Instead, software and updates are available in the “Downloads” section under your customer account on https://support.varien.com after successful authorization.

PCI-Compliant Remote Access (11.2 and 11.3.b)

Magento Payment Bridge web application does not provide any remote access for the end user.

Data Transport Encryption (PA-DSS 12.1.b)

The PCI DSS requires the use of strong cryptography and encryption techniques with at least a 128 bit encryption strength (either at the transport layer with SSL or IPSEC; or at the data layer with algorithms such as RSA or Triple-DES) to safeguard cardholder data during transmission over public networks (this includes the Internet and Internet accessible DMZ network segments).

PCI DSS requirement 4.1: Use strong cryptography and security protocols such as secure sockets layer (SSL) / transport layer security (TLS) and Internet protocol security (IPSEC) to safeguard sensitive cardholder data during transmission over open, public networks.

- Refer to the Dataflow diagram for an understanding of the flow of encrypted data associated with Magento Payment Bridge.

PCI-Compliant Use of End User Messaging Technologies (PA-DSS 12.2.b)

Magento Payment Bridge does not allow or facilitate the sending of PANs via any end user messaging technology (for example, e-mail, instant messaging, and chat).

Non-console administration (PA-DSS 13.1)

Although Magento Payment Bridge does not support non-console administration and we do not recommend using non-console administration, should you ever choose to do this, must use SSH, VPN, or SSL/TLS for encryption of this non-console administrative access.

Network Segmentation

The PCI DSS requires that firewall services be used (with NAT or PAT) to segment network segments into logical security domains based on the environmental needs for internet access. Traditionally, this corresponds to the creation of at least a DMZ and a trusted network segment where only authorized, business-justified traffic from the DMZ is allowed to connect to the trusted segment. No direct incoming internet traffic to the trusted application environment can be allowed. Additionally, outbound internet access from the trusted segment must be limited to required and justified ports and services.

- Refer to the standardized Network diagram for an understanding of the flow of encrypted data associated with Magento Payment Bridge.

Maintain an Information Security Program

In addition to the preceding security recommendations, a comprehensive approach to assessing and maintaining the security compliance of the payment application environment is necessary to protect the organization and sensitive cardholder data.

The following is a very basic plan every merchant/service provider should adopt in developing and implementing a security policy and program:

- Read the PCI DSS in full and perform a security gap analysis. Identify any gaps between existing practices in your organization and those outlined by the PCI requirements.
- Once the gaps are identified, determine the steps to close the gaps and protect cardholder data. Changes could mean adding new technologies to shore up firewall and perimeter controls, or increasing the logging and archiving procedures associated with transaction data.
- Create an action plan for on-going compliance and assessment.
- Implement, monitor and maintain the plan. Compliance is not a one-time event. Regardless of merchant or service provider level, all entities should complete annual self-assessments using the PCI Self Assessment Questionnaire.
- Call in outside experts as needed.

Application System Configuration

Below are the operating systems and dependent application patch levels and configurations supported and tested for continued PCI DSS compliance.

- Linux x86, x86-64 Operation System. All latest updates and hot-fixes should be tested and applied.
- 512 MB of RAM minimum, 1GB or higher recommended for Payment Application.
- 2 GB of available hard-disk space.
- TCP/IP network connectivity.
- Apache HTTP Server version 2.0 or higher. All latest updates and hot-fixes should be tested and applied.
- PHP version 5.2 or higher. All latest updates and hot-fixes should be tested and applied.
- MySQL Server version 5.0 and higher. All latest updates and hot-fixes should be tested and applied.

Payment Application Initial Setup & Configuration

Installing Magento Payment Bridge, setting up permissions for directories

1. Outside the DOCUMENT_ROOT web server directory, create an application system directory (<system_dir>).
2. Deploy the distributive archive into <system_dir>.
3. Move all files from <system_dir>/pub to the DOCUMENT_ROOT (<web_dir>).
4. Set corresponding permissions for the <system_dir>/var directory so that the web server can create directories and write log files into that directory.
5. Edit the <web_dir>/bridge.php file by modifying the $appLocation variable value so that it points to <system_dir> relatively from <web_dir>.

Database

1. Create a database.
2. Create a database structure using the file <system_dir>/sql/structure.sql.

Set up the configuration file <system_dir>/cfg/config.php

1. logging (true/false) – turning the logging on or off.
2. development (true/false) – turning the development/testing mode on or off (when turned on, all the errors, warnings, notices, exceptions are visible to the user).
3. base_url (string value) – should be either empty or pointed to <web_dir_> (recommended).
4. crypt_key_db (string value) – an encryption key used for the database data encryption/decryption. To set this value, you must follow the instructions in the Application Setup Tool section of the Magento Payment Bridge Administration Guide.
5. resource – database access info, contains the following:
    - dsn – PDO Data Source Name, where dbname is the name of the database; host is the database host name.
    - username – database access user name.
    - password – database access password. To set this value, you must follow the instructions in the Application Setup Tool section of the Magento Payment Bridge Administration Guide.
6. payment_gateways – a list of payment gateways supported by Magento Payment Bridge,
    - debug (true/false) – turn the gateway-specific logging on or off, for PayPal/PayPalUK, controls turning on or off an IPN log.
7. token (numeric value in seconds) – token lifetime (token is a unique application session identifier).Token will be deleted when its lifetime expires. Applicable for tokens which haven't yet received a successful response from any payment gateway.
8. cron – allows configuring scheduled processes,
    - period (numeric value in seconds) – processes launch recurrence.
    - the rest parameters are not recommended to be changed.

Create a merchant (console script <system_dir>/tools/merchant.php is used)

1. Run php <system_dir>/tools/merchant.php -a (create a merchant), then take all steps for filling up data to create a merchant.

Configure Magento

1. Run php <system_dir>/tools/merchant.php -i <merchant_code>, where <merchant_code> is a specified merchant code. As a result, all merchant information will be printed out.
2. Fill in corresponding configuration options in Magento (System > Configuration > Payment Methods > Payment Bridge) in accordance with the previously printed information.

Set up IP filtering

1. Run php <system_dir>/tools/ipfilter.php -a <magento_IP>, where <magento_IP> is an IP address of the host with Magento installed.

Configure payment gateways

1. Run php <system_dir>/tools/merchant.php --pgc <merchant_code>, where <merchant_code> is the previously specified merchant code.
2. Payment gateway code will be requested.

3. Enter the payment gateway code, then specify all necessary values, such as API URL, API credential, etc., interactively. Some parameters with predefined values can be skipped by pressing the ENTER key.