



# **Znode Storefront Singlefront and Multifront Edition PCI Guide**

Version 5.4.0

# **Znode Storefront Singlefront and Multifront Edition PCI Guide**

**© Copyright 2009, Znode Inc, All Rights Reserved**

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: October 2009 in USA

# Table of Contents

<b>Part 1 Overview</b>	<b>4</b>
<b>Part 2 Revision History</b>	<b>5</b>
<b>Part 3 Install and Maintain a Firewall</b>	<b>6</b>
<b>Part 4 Change Default Passwords</b>	<b>7</b>
<b>Part 5 Protect Stored Cardholder Data</b>	<b>9</b>
<b>Part 6 Encrypt Transmission of Data</b>	<b>10</b>
<b>Part 7 Use Anti-virus Software</b>	<b>11</b>
<b>Part 8 Maintain Secure Systems</b>	<b>12</b>
<b>Part 9 Restrict Access</b>	<b>14</b>
<b>Part 10 Assign a Unique ID to Each Person</b>	<b>15</b>
<b>Part 11 Restrict Physical Access</b>	<b>17</b>
<b>Part 12 Track and Monitor all Access</b>	<b>18</b>
<b>Part 13 Regularly Test Security</b>	<b>19</b>
<b>Part 14 Have a Policy on Security</b>	<b>20</b>

# 1 Overview

PCI-DSS is a standard created by the credit card industry outlining twelve basic requirements on how to securely deploy your application. The goal of this specification is to help prevent the theft of sensitive credit card information.

Znode Storefront is PABP certified which means that it has been verified by a third party to work in a PCI-DSS environment. PABP standards only apply to Znode Storefront as an application and do not apply to the way it is hosted on your server. PCI-DSS requirements provide further guidelines as to how to deploy your PABP certified application securely. This document will help you comply with the PCI-DSS guidelines and host Znode Storefront securely in your environment.

This guide is not a substitute for reading and understanding the Payment Card Industry (PCI) Data Security Standard which goes into much more specific detail on what you need to do to be PCI-DSS compliant. **It is up to you to follow these guidelines in order to be PCI-DSS compliant.**

The Payment Card Industry (PCI) Data Security Standard can be found here:

[https://www.pcisecuritystandards.org/tech/download\\_the\\_pci\\_dss.htm](https://www.pcisecuritystandards.org/tech/download_the_pci_dss.htm)

In each of the following sections we will broadly list out each of the PCI-DSS practices that you need to follow for compliance. Where these practices directly impact Znode Storefront we will make note of it in *Integrator Notes* and *Configuration Notes*. Integrator Notes are things to be aware of that are specific to your environment or practices. Configuration Notes are Znode Storefront related settings that must be made.

## 2 Revision History

Revision	Date	Description
1.0	08/20/2008	Initial Release
1.1	01/06/2008	Updated for Version 5.2 of Znode Storefront Enterprise Edition
1.2	06/21/2009	Updated Version number to 5.3 for Znode Singlefront Edition
1.3	10/28/2009	Update Version number to 5.4 for Znode Singlefront Edition

## 3 Install and Maintain a Firewall

### Integrator Notes

- Servers that you deploy your production application on must have a hardware firewall. The processes for setting the firewall must be documented and the firewall settings must be reviewed and tested on a quarterly basis.
- Per PCI-DSS 1.3.8, install perimeter firewalls between any wireless networks and the cardholder data environment, and configure these firewalls to deny any traffic from the wireless environment or from controlling any traffic (if such traffic is necessary for business purposes)
- Per PCI-DSS 1.3.9, install personal firewall software on any mobile and employee-owned computers with direct connectivity to the Internet (for example, laptops used by employees), which are used to access the organization's network.

## 4 Change Default Passwords

### Integrator Notes

- Assign secure authentication to default accounts and then disable or do not use the accounts. This includes system logins as well as database logins.
- Assign secure authentication for administrative access to payment applications and data. Refer to PCI-DSS Requirements 8.5.8 through 8.5.15 for guidance on how to implement PCI-DSS compliant authentication.
- These secure authentication methods outlined in PCI-DSS 8.5.8-8.5.15 should be implemented on your database server and on any other servers, PC's, etc in your enterprise that are part of the payment application infrastructure.
- Only one primary function is allowed per server. This means that you should have a separate server for your database and one server for your web server. Email and DNS should be hosted on separate servers as well.
- If you utilize wireless technology within your environment, you must change wireless vendor defaults, including but not limited to, wired equivalent privacy (WEP) keys, default service set identifier (SSID), passwords, and SNMP community strings. Disable SSID broadcasts. Enable Wi-Fi protected access (WPA and WPA2) technology for encryption and authentication when WPA-capable. (See PCI-DSS 2.1.1)
- Per PCI-DSS 2.3 you must encrypt all non-console administrative access. Use technologies such as SSH, VPN, or SSL/TLS (transport layer security) for web-based management and other non-console administrative access.

### Configuration Notes

In setting up Znode Storefront you must supply a database connection string in the web.config file.

- When setting this connection string use a user ID and password that are specific to your web application. If you use SQL Server Authentication NOT share user IDs and passwords between applications. Passwords should be strong with at least 7 characters and a mix of digits and letters.
- Likewise if you use Windows Authentication make your user a member of the IIS\_WPG (IIS Worker Process) group only and not an administrator or user.
- Do NOT use a system administrator account for your database access. Instead you should use the minimum role required for connecting to the database which is db\_owner and public.
- Do use a separate Web server and Database server for your storefront. In other words your connection string should point to a different server to connect to the database. Disable all unnecessary services on each port. For instance, your database server should not have Port 80 open.

The admin section of Znode Storefront can be set up to encrypt your data stream using HTTPS. To enable this you must select "Enable SSL" in the Global Settings. With this setting enabled Znode Storefront will switch to HTTPS at the appropriate times such as when logging in.

"Enable SSL" will not be set by default. On your first login to the Admin click on the "Use Secure Login (SSL)" to switch to https before typing in your user ID and password (SSL must be properly installed on your server).

Znode Storefront has settings to turn on diagnostics. In your production environment you should have these options turned off by setting the following parameters in the web.config:

```
<add key="EnableDiagnostics" value="0"/>  
<add key="EnableIntegrationTest" value="0"/>
```

In addition you should not report errors that occur in the application to outside users. Be sure to have the following setting in your web.config:

```
<customErrors defaultRedirect="~/error.aspx" mode="RemoteOnly"/>
```

Znode Storefront uses an user generated key to encrypt various passwords throughout the system such as gateway login credentials. A default key is provided with the storefront but you should generate your own key before deploying your web site. To regenerate the encryption key do the following:

1. Log into the Admin.
2. Click on the Maintenance link on the left navigation.
3. On the Security tab you will see a "Rotate key" link. Click it.
4. Click Generate on the Key Rotation page.
5. **After rotating your key you will need to reset your payment settings for credit card processing and other settings that require a password.** Login passwords are encrypted through the Microsoft Membership Provider and will be unaffected.

## 5 Protect Stored Cardholder Data

### Integrator Notes

To alleviate much of the risk and complexity of storing card holder data, Znode Storefront will not store the PAN (Primary Account Number, or Credit Card Number) or log any sensitive customer data. Not storing the PAN alleviates the need to encrypt the Cardholder Name, and Expiration Date.

While you can modify the code to store the PAN or log sensitive data, you should NOT unless you intend to have your new code PABP re-certified by an accredited third party.

Per PCI-DSS 1.3.4 your database should reside on an internal network segment that is segmented from the DMZ.

When testing a storefront you should not use live credit card data but instead use test credit cards provided by your gateway provider.

Should you need to store sensitive data for debugging purpose you must ensure the following:

- 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

### Configuration Notes

Versions of Znode Storefront before 5.0 optionally allowed you to store sensitive Credit Card data. Under PCI-DSS requirement 3.2, storage of sensitive authentication data is prohibited. We provide a script called "Remove Credit Card Information from Versions 4.5 and Below.sql" that will erase sensitive Credit Card data in these fields. Erasing this data will not disable the application but render the saved credit card data unreadable. This script will not work in Version 5.0 of the database and above as these fields have been removed from the schema. Because of this, removal of sensitive credit card data from version 5.0 databases and above should not be an issue.

To run the "Remove Credit Card Information from Versions 4.5 and Below.sql" script, open it in SQL Server Management Studio and run it against your pre-5.0 Znode Storefront database. After the script has been run, verify that the ZnodeOrder table does not have any sensitive credit card data in it.

When upgrading to version 5.0 you must also be sure to securely remove the Web/bin/ZNode.Libraries.Framework.Business.dll. This will remove the previous encryption key used for the above credit card data and is required for PCI-DSS compliance. Use a utility such as sdelete to securely delete the DLL and its associated encryption key. Sdelete can be found at <http://technet.microsoft.com/en-us/sysinternals/bb897443.aspx>

## 6 Encrypt Transmission of Data

### Integrator Notes

To prevent sensitive data from being exposed on an open or public network it must be encrypted using strong cryptography and security protocols. For the purposes of your storefront this means using an SSL certificate on your web server.

If you use wireless networks for communication with the storefront, web server, or database server you must encrypt the Wi-Fi transmissions (per PCI-DSS 4.1.1) with the following:

- Use with a minimum 104-bit encryption key and 24 bit-initialization value
- Use ONLY in conjunction with Wi-Fi protected access (WPA or WPA2) technology, VPN or SSL/TLS
- Rotate shared WEP keys quarterly (or automatically if the technology permits)
- Rotate shared WEP keys whenever there are changes in personnel with access to keys
- Restrict access based on media access code (MAC) address.
- Per PCI-DSS 4.2, never send unencrypted Credit Card numbers by email or other public messaging technology such as instant messenger.

### Configuration Notes

Always install an SSL certificate on your production server. The “Enable SSL” option in Znode Storefront is not set by default. On your first login to the Admin click on the “Use Secure Login (SSL)” to switch to https before typing in your user ID and password (SSL must be properly installed on your server).

Once you are securely logged into the Znode Storefront Admin, go to the Global Settings and check “Enable SSL”. From then on, your site will switch to HTTPS at the appropriate times.

This step is required to maintain compliance with PCI-DSS.

## 7 Use Anti-virus Software

### Integrator Notes

You should ensure that anti-virus software is installed on your servers and is kept up to date. Personal computers within your enterprise should also have anti-virus software installed.

## 8 Maintain Secure Systems

### Integrator Notes

On your servers you should ensure that the latest security patches have been applied.

When making changes to Znode Storefront use the following industry best practices:

- Test of all security patches and system and software configuration changes before deployment
- Separate development, test, and production environments
- Separation of duties between development, test, and production environments
- Production data (live PANs or Credit Card numbers) are not used for testing or development
- Remove test data and accounts before production systems become active
- Remove custom application accounts, usernames, 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.

Follow change control procedures for all system and software configuration changes. The procedures must include the following:

- Documentation of impact
- Management sign-off by appropriate parties
- Testing of operational functionality
- Back-out procedures

Develop all web applications based on secure coding guidelines such as the Open Web Application Security Project guidelines. Review custom application code to identify coding vulnerabilities. Cover prevention of common coding vulnerabilities in software development processes, to include the following:

- Un-validated input
- Broken access control (for example, malicious use of user IDs)
- Broken authentication and session management (use of account credentials and session cookies)
- Cross-site scripting (XSS) attacks
- Buffer overflows
- Injection flaws (for example, structured query language (SQL) injection)
- Improper error handling
- Insecure storage
- Denial of service
- Insecure configuration management

Ensure that all web-facing applications are protected against known attacks by applying either of the following methods:

- Have all custom application code reviewed for common vulnerabilities by an organization that specializes in application security
- Install an application layer firewall in front of web-facing applications. Note that installing an application layer firewall is required per PCI-DSS Requirement 6.6.

Further information on secure web development can be found at:

[http://www.owasp.org/index.php/OWASP\\_Top\\_Ten\\_Project](http://www.owasp.org/index.php/OWASP_Top_Ten_Project)

**When deploying applications to production be sure to make backups of both your database and application code first.** PCI-DSS requires that you have well defined procedures for testing your code prior to deploying to production and that you have production back-out procedures should you have issues with your deployment.

### Configuration Notes

By default the web services feature of Znode Storefront is secured and not accessible. You should properly configure your webservices directory to ensure that web services are not publicly accessible.

## 9 Restrict Access

### Integrator Notes

You can use the Role Based security in Znode Storefront to limit access depending on user role. If you modify the behavior of the user roles then keep the amount of information accessible by users of the system to a minimum. Generally you will want to restrict customer billing name and address. Role based security does not apply to Credit Card numbers as they are not saved in the system.

User Role support is only included in some editions of Znode Storefront. For those editions where it is not included all your admin users will have full access to the system. Be sure to only give this access to users that need this capability.

### Configuration Notes

The role based security of the admin can be modified using the web.config file in Web\Admin\Secure\web.config.

As shipped Znode Storefront does not have the user roles feature enabled. To install the user roles you must run the Web\Plugins\UserRoles\Create\_UserRoles.sql script in the database.

## 10 Assign a Unique ID to Each Person

### Integrator Notes

By default Znode Storefront will require strong passwords and applies the proper policies when it comes to locking out invalid logins, password rotation, and password retrieval. In addition you must create unique user names and complex passwords in your server and database environment.

When creating user names and passwords on your database or server use the following best practices:

- Do not use group, shared, or generic accounts and passwords
- Change user passwords at least every 90 days
- Require a minimum password length of at least seven characters
- Use passwords containing both numeric and alphabetic characters
- Do not allow an individual to submit a new password that is the same as any of the last four passwords he or she has used
- Limit repeated access attempts by locking out the user ID after not more than six attempts
- Set the lockout duration to thirty minutes or until administrator enables the user ID
- If a session has been idle for more than 15 minutes, require the user to re-enter the password to re-activate the terminal
- Assign secure authentication to default accounts even if they won't be used. Disable or do not use these accounts.

If you allow remote access to your servers or web application, per PCI-DSS 8.3 you must implement a two factor authentication method. Use technologies such as remote authentication and dial-in service (RADIUS) or terminal access controller access control system (TACACS) with tokens; or VPN (based on SSL/TLS or IPSEC) with individual certificates.

Additionally the following best practices should be observed when permitting access to the payment application environment:

- Change default settings in the remote access software (for example, change default passwords and use unique passwords for each customer).
- Allow connections only from specific (known) IP/MAC addresses
- Use strong authentication and complex passwords for logins, according to PCI-DSS Requirements 8.1, 8.3 and 8.5.8-8.5.15.
- Enable encrypted data transmission according to PCI-DSS Requirement 4.1.
- Enable account lockout after a certain number of failed login attempts according to PCI-DSS 8.5.13
- Configure the system so a remote user must establish a VPN connection via a firewall before access is allowed.
- Enable the logging function
- Restrict access to customer passwords to authorized personnel
- Establish passwords according to PCI-DSS Requirements 8.1, 8.2, 8.4 and 8.5

In Znode Storefront you should also follow these practices:

- Each user of the Znode Storefront Admin should have their own login ID. Do not use group or shared accounts and passwords.
- Do not use system administrative logins and passwords for Znode Storefront.
- The ZnodeActivityLog table in the database will log user login attempts (as well as other activities).

### Configuration Notes

- There is a setting in the “membership” section of the web.config that controls password strength and number of attempts before locking out the user. Modifying the behavior of the login mechanism of Znode Storefront may invalidate the PABP certification.
- Use secure passwords with at least seven characters and both numeric and alphabetic characters in your database connection string in the web.config. Do not use default logins in your connection string.
- Changing the default user access constraints for secure access will result in non-compliance with PCI-DSS.

# 11 Restrict Physical Access

## Integrator Notes

Znode Storefront does not save sensitive cardholder data but these general practices should be followed when data is stored in other systems.

- Physically secure all servers that may hold cardholder data.
- Physically secure and monitor any hard copy or backups of cardholder data.
- Destroy any hard copies or backup copies of cardholder data after it is no longer needed.

## 12 Track and Monitor all Access

### Integrator Notes

The ZnodeActivityLog table in the database will log user login attempts (as well as other activities). Access to this table is available directly through SQL Server. Logging to the database is done using the LogActivity method.

Znode also provides the facility to log events to the Data\Default\logs\ZnodeLog.txt file through the LogMessage and LogObject functions. This log file can be used for debugging purposes.

If you use either of these logging facilities for your own purposes you should not log sensitive credit card information.

Logging must be enabled in SQL Server to maintain PCI-DSS compliance. For more information on this see: <http://technet2.microsoft.com/windowsserver/en/library/bac482ae-39c4-44b7-bd9f-291ab354ef2b1033.msp?mfr=true>

### Configuration Notes

In a production environment application logging and diagnostics should be turned off. To turn off these features set the following in your web.config:

```
<add key="EnableDebugging" value="0"/>
<add key="EnableDiagnosticsPage" value="0"/>
<add key="EnableIntegrationTest" value="0"/>
<add key="EnableActivationPage" value="0"/>
```

Activity logging to the ZnodeActivityLog table in the database is controlled by another setting in the web.config. **Disabling or modifying the logs is prohibited and will result in non-compliance with PCI-DSS.** Be sure that the following setting is in your web.config.

```
<add key="EnableLogging" value="1"/>
```

## 13 Regularly Test Security

### Integrator Notes

Quarterly external vulnerability scans must be performed by a scan vendor qualified by the payment card industry. Scans conducted after network changes may be performed by the company's internal staff.

# 14 Have a Policy on Security

## Integrator Notes

Per PCI-DSS 12.3, develop usage policies for critical employee-facing technologies (such as modems and wireless) to define proper use of these technologies for all employees and contractors. Ensure these usage policies require the following:

- Explicit management approval
- Authentication for use of the technology
- List of all such devices and personnel with access
- Labeling of devices with owner, contact information, and purpose
- Acceptable uses of the technologies
- Acceptable network locations for the technologies

**Znode Storefront Singlefront and Multifront Edition PCI Guide**

© Copyright 2009, Znode Inc, All Rights Reserved

Printed: October 2009 in USA.