Pushpay Security Overview

Version 1.2

Last updated 24-09-2013

Version Control:

Date	Version	Description	Author
11/04/13	1.1		Phil Howie
24/09/13	1.2	Updated to include explicit PCI responsibilities	Paul Shingles

Pushpay Security Overview

Overview

This document gives a high level overview of the security measures Pushpay has in place to ensure all data is kept safe and private from unwanted access, and to also keep our systems robust and reliable.

Application

The Pushpay application consists of three major components – an iPhone and Android application, a public website with secure access for customers and a supporting application API to process all of the functionality from the mobile application.

The mobile application for iOS is built using Apple's Cocoa framework on top of iOS. This is the official recommended path for building solid iPhone applications and takes advantage of all the security measures present in iOS such as encryption, sandboxing and passcode access.

The Android application is also built using technologies such as Java, which are native to the operating system. The code for both applications has been independently reviewed and signed off by our security specialist consulting company.

Primarily we are concerned with safeguarding access to our customer's credit card data while maintaining a useable and seamless experience. To achieve this we:

- Require a passcode be set by the user as a second factor (in conjunction to their API user session credentials) which is then used to protect sensitive parts of the application such as performing a payment
- Allow the user to avoid having to specify their username and password by exchanging this for a server generated token. This token is held in local storage on the device and will expire after a period of time (e.g. two weeks) after which the customer will have to re-authenticate. This token is also removed from the device when the application is uninstalled or reinstalled.

The public website and API application are built on the Microsoft .NET platform (including SQL Server) leveraging the ASP.NET MVC 4 Framework. For the public website we are primarily concerned with providing access to perform administrative operations and view transactional information while maintaining a robust level of security. For the API we are concerned with safeguarding access against unauthorised use of the API while allowing for relatively simple integration.

We have specifically applied approaches which mitigate known web vulnerabilities by both leveraging the standard framework features of ASP.NET MVC and leveraging appropriate 3rd party components where applicable.

Development

Pushpay follows a set development process to ensure code in written and reviewed appropriately. Code is peer reviewed to ensure a high quality, as well as externally reviewed to ensure there are no security vulnerabilities present.

Access to source code is highly restricted and is only available to specific people who are working on the application.

Data

Pushpay handles credit cards and facilitates transactions on behalf of merchants through our payment gateway. Credit card numbers are **not** stored by Pushpay, but instead each credit card is tokenised by the gateway and the token is stored. This ensures credit card numbers cannot be exposed via security intrusions or accidental data exposure and it is generally desirable not to store them if possible.

As we are still handling card details briefly during the customer registration process, we are still taking full measures (such as PCI-DSS compliance) to ensure data is kept safe whilst it passes through our system.

Hosting

Pushpay is hosted in a first class data centre with a completely dedicated network and firewall. Access to the hosting environment is highly restricted and is secured over VPN access. The hosting infrastructure and server environment are fully PCI-DSS compliant (see more below) and are setup with industrial grade security measures. Banks and government organisations are housed in the same hosting facility.

All communication in and out of the Pushpay ecosystem is secured and encrypted with SSL. This includes any visitors to the website, and any traffic between the mobile application and supporting API. Communication with external providers such as Paystation (our payment gateway) is also encrypted.

An Extended Validation SSL certificate has also been put in place to add further validity to our public website and to help ensure consumer confidence.

Compliance

Pushpay is fully PCI DSS compliant – Level 1. We work closely with Confide to ensure all requirements are met and maintain our level of compliance.

As an organisation we adhere to the following principals. Anyone involved in these areas will read and understand the applicable sections of PCI DSS v2. For convenience, a high level overview of our responsibilities is listed below (Figure 1)

PCI Data Security Standard – High Level Overview

Build and Maintain a Secure Network	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. Use 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. 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 for all personnel.

Figure 1 - PCI DSS High Level Responsibilities

Professional Advice

Pushpay recognises we need the best security people we can find and therefore are working closely with both Confide and Insomnia Security to ensure our application meets and surpasses security requirements.

Application code is independently reviewed and is also penetration tested on a regular basis to expose any weaknesses the application might have.