PASSCODE VULNERABILITY ON POCKET EXPENSE FOR iOS

POCKET EXPENSE
SIMPLE, ATTRACTIVE INTERFACE
POWERFUL STATISTICS AND REPORTS OF ALL YOUR TRANSACTIONS
TRACK YOUR ACCOUNTS

July 22th, 2014

@YoKoAcc (yk@firstsight.me)

[English Version]

TABLE OF CONTENTS

TABLE OF CONTENTS.. 1

TABLE OF FIGURES... 2

I. ABSTRACT 3

II. INTRODUCTION 3

III. AFFECTED VERSION AND CONDITION 4

IV. PROOF OF CONCEPT 4

V. RESPONSE FROM DEVELOPER 6

VI. SUMMARY AND RECOMMENDATION 7

VII. ADDITIONAL INFORMATION 8

TABLE OF FIGURES

Figure 1 Pocket Expense Version for iOS 4

Figure 2 Pocket Expense Application Directory 4

Figure 3 SQLite Files on Directory Folder 5

Figure 4 PocketExpense Database Structure 5

Figure 5 Passcode on iOS 7 - iPad Mini Retina Display 5

Figure 6 Passcode on iOS 6 - iPhone 4s 6

Figure 7 Pocket Expense v.4.5.1 6

Figure 8 Bug Report to Developer 7

Figure 9 Purchase Date and Purchase History 8

Figure 10 Show in iTunes Store 8

I. ABSTRACT

In our daily life, many people are often finding some difficulties to manage their financial record. This record generally is about data of income and outcome activities which were in the past or even in the future. In an era which is all about digital like now, manual record is being problem which is hard to be fulfilled. Beside ineffective in recording and searching process, this manual record also has problem to protect itself from being used without permission.

With seeing the need of solving those problems, some developers and even companies are sprung to build and offer many software that help its user to manage their financial record easily. Spending Tracker, Money Monitor, or Pocket Expense are some of the well-known software and placing top for "money" or "account" keyword in Indonesia's iTunes. Some developers or companies realize the importance of the data and make them offer protection which is considered to protect the user's confidentiality. Generally, the protection is in the form of PIN or being well-known as Passcode

Talking about this topic, The Writer will discuss about the safety weakness of Passcode which is offered by "Pocket Expense" (http://www.appxy.com/pocket-expense/) that has reported by The Writer to the developer on July 7th, 2014 GMT+7.

II. INTRODUCTION

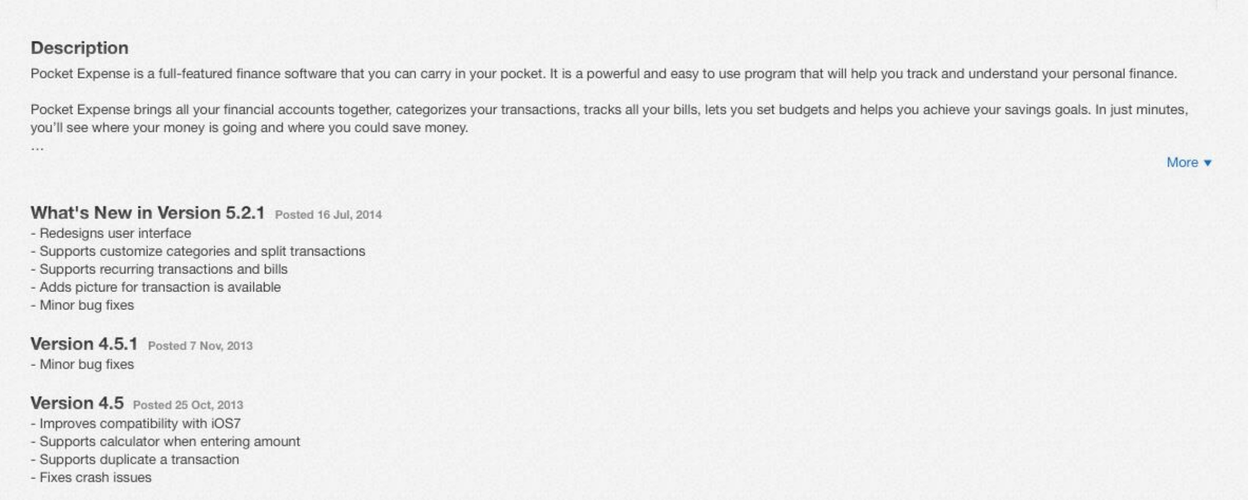
Passcode is a need for everyone who wants to protect their confidentiality about something which is considered to be a secret. According to the Oxford Dictionary, Passcode is "A string of characters used as a password, especially to gain access to a computer or smartphone". In another reference, passcode is a kind of password that is generally used as a mechanism to show identity. Passcode and Password are assessed as something which are made by the user.

In general, the difference between Password and Passcode can be seen from the type of characters that are inputted. If Passcode makes the user must input (only) number, therefore in Password, the user can input a combination of alphabet, number, and symbol.

Then, let's get to the main topic. Based on the statements above, Pocket Expense is one of the application that offer easiness for their users to manage their financial record. Appxy as the developer also offer the usage of Passcode that only be able for the users who buy the Pro Version of the related application.

Based on the discussion in the Abstract part, in this application, Writer finds that an Attacker can gain Passcode from Pocket Expense application which has been set by the users in order to protect their data. Of course, with using this Passcode, Attacker will be able to login to the related application or even using it to login to the user's iDevice. Why is this possible? Because generally, someone will use same Passcode as an identification in every systems that have login feature. Moreover, Passcode in Pocket Expense has only 4 digits like iDevice's which is published by Apple (if the user activates the simple Passcode on their iDevice).

For information, Writer uses Pocket Expense 4.5.1, which is one version late to the version 5.2.1 which has been published by Appxy after the Writer submitted the vulnerability of the Passcode.



Description

Pocket Expense is a full-featured finance software that you can carry in your pocket. It is a powerful and easy to use program that will help you track and understand your personal finance.

Pocket Expense brings all your financial accounts together, categorizes your transactions, tracks all your bills, lets you set budgets and helps you achieve your savings goals. In just minutes, you'll see where your money is going and where you could save money.

...

What's New in Version 5.2.1 Posted 16 Jul, 2014

- Redesigns user interface
- Supports customize categories and split transactions
- Supports recurring transactions and bills
- Adds picture for transaction is available
- Minor bug fixes

Version 4.5.1 Posted 7 Nov, 2013

- Minor bug fixes

Version 4.5 Posted 25 Oct, 2013

- Improves compatibility with iOS7
- Supports calculator when entering amount
- Supports duplicate a transaction
- Fixes crash issues

Figure 1 Pocket Expense Version for iOS

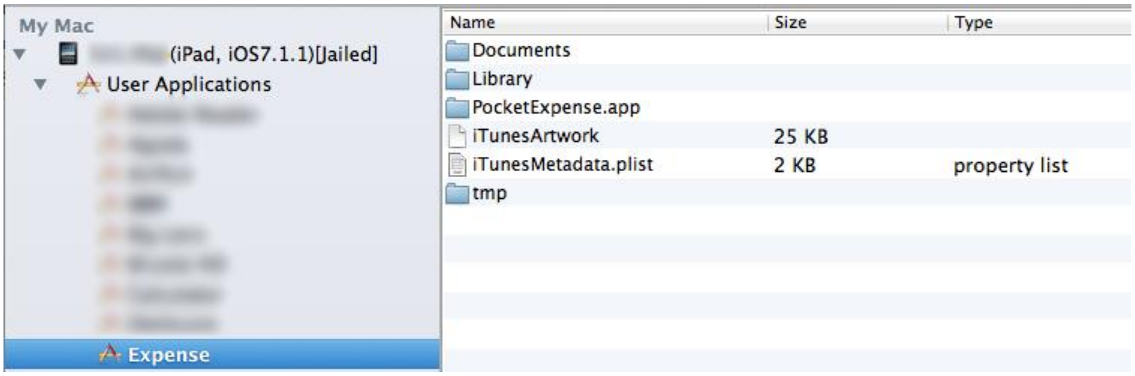
III. AFFECTED VERSION AND CONDITION

The version of Pocket Expense which is having vulnerable security is the 4.5.1 version. Here are the conditions that allow the Attacker to read the Passcode from Pocket Expense:

1. iDevice isn't need to be in jail-break condition.
2. iDevice isn't need to be in standby condition. In other words, this gap can also being used even the iDevice is in locked condition with the Passcode for that iDevice itself.
3. Writer tries it in Pocket Expense for iPhone and iPad with 6.1.3 and 7.1.1. iOS version.

IV. PROOF OF CONCEPT

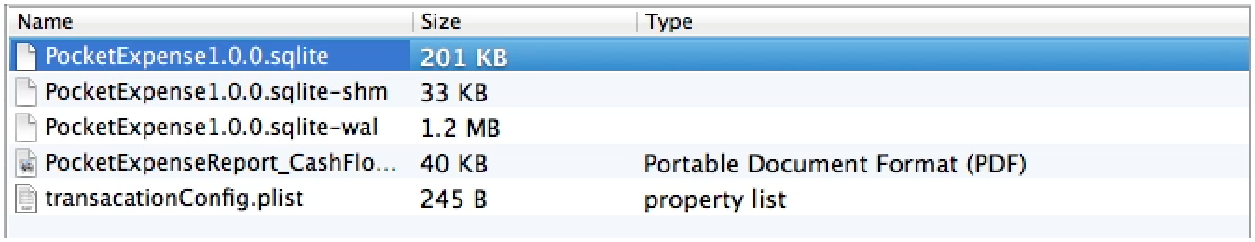
1. Connect the iDevice which has Pocket Expense in it with PC.
2. Access the "Applications" directory in iDevice with iExplorer tools, like iFunBox.
3. Go to the "Expense" application.



Name	Size	Type
Documents		
Library		
PocketExpense.app		
iTunesArtwork	25 KB	
iTunesMetadata.plist	2 KB	property list
tmp		

Figure 2 Pocket Expense Application Directory

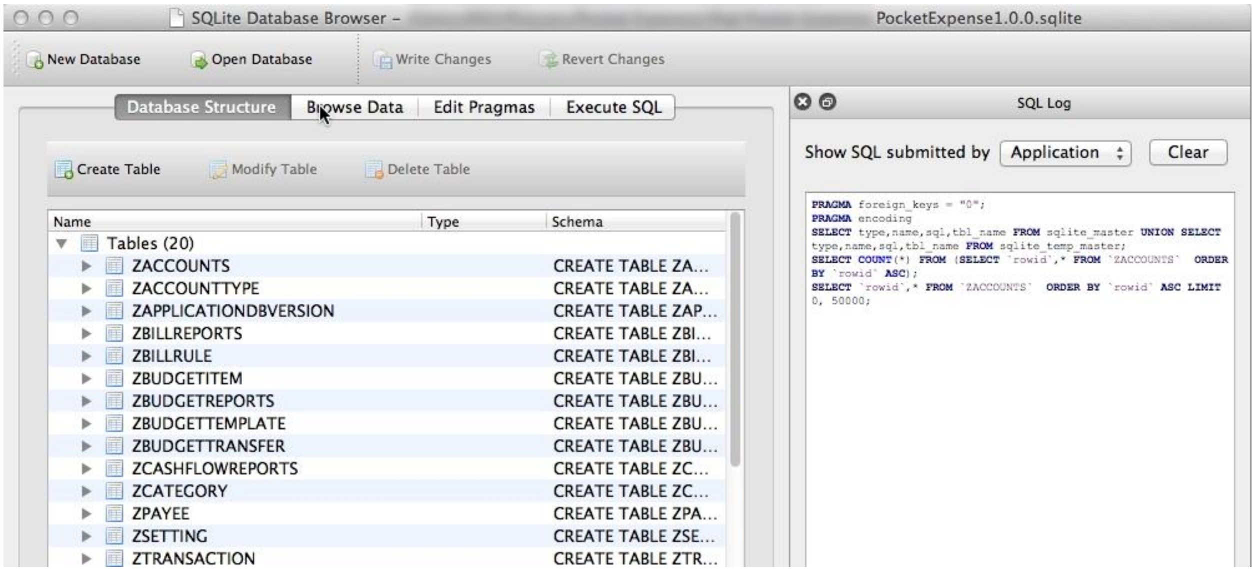
4. Go to the “Documents” directory and we’ll see some files. One of them is a file with ‘sqlite’ format.



Name	Size	Type
PocketExpense1.0.0.sqlite	201 KB	
PocketExpense1.0.0.sqlite-shm	33 KB	
PocketExpense1.0.0.sqlite-wal	1.2 MB	
PocketExpenseReport_CashFlo...	40 KB	Portable Document Format (PDF)
transacationConfig.plist	245 B	property list

Figure 3 SQLite Files on Directory Folder

5. Open “PocketExpense1.0.0.sqlite” file with SQLiteBrowser or such.



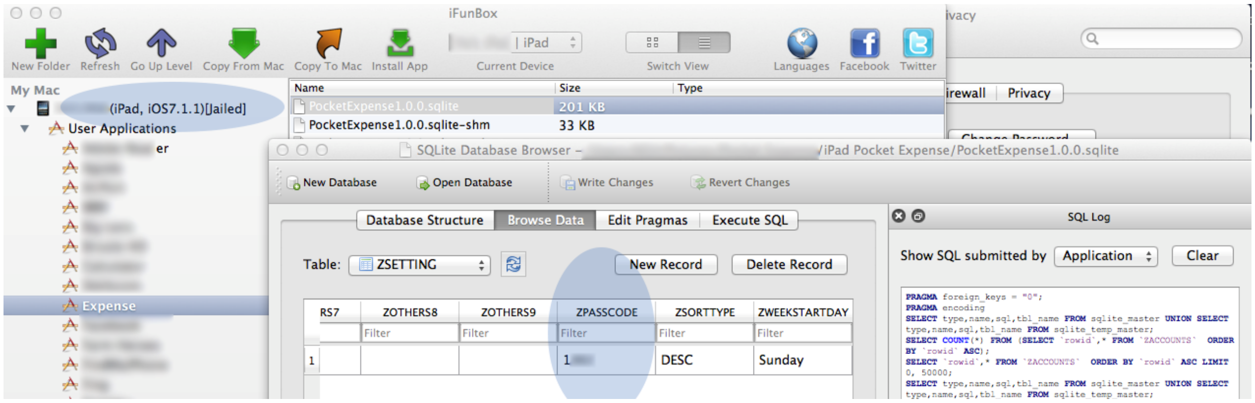
SQLite Database Browser – [redacted] PocketExpense1.0.0.sqlite

Name	Type	Schema
Tables (20)		
ZACCOUNTS		CREATE TABLE ZA...
ZACCOUNTTYPE		CREATE TABLE ZA...
ZAPPLICATIONDBVERSION		CREATE TABLE ZAP...
ZBILLREPORTS		CREATE TABLE ZBI...
ZBILLRULE		CREATE TABLE ZBI...
ZBUDGETITEM		CREATE TABLE ZBU...
ZBUDGETREPORTS		CREATE TABLE ZBU...
ZBUDGETTEMPLATE		CREATE TABLE ZBU...
ZBUDGETTRANSFER		CREATE TABLE ZBU...
ZCASHFLOWREPORTS		CREATE TABLE ZC...
ZCATEGORY		CREATE TABLE ZC...
ZPAYEE		CREATE TABLE ZPA...
ZSETTING		CREATE TABLE ZSE...
ZTRANSACTION		CREATE TABLE ZTR...

```
PRAGMA foreign_keys = "0";
PRAGMA encoding
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER BY `rowid` ASC LIMIT
0, 50000;
```

Figure 4 PocketExpense Database Structure

6. After that, access the “ZSETTING” table and pay attention to the “ZPASSCODE” part. You will see that this “Passcode” which is used to protect the data from Pocket Expense is being saved in plaintext and can be accessed even the iDevice isn’t in jail-break condition.



Name	Size	Type
PocketExpense1.0.0.sqlite	201 KB	
PocketExpense1.0.0.sqlite-shm	33 KB	

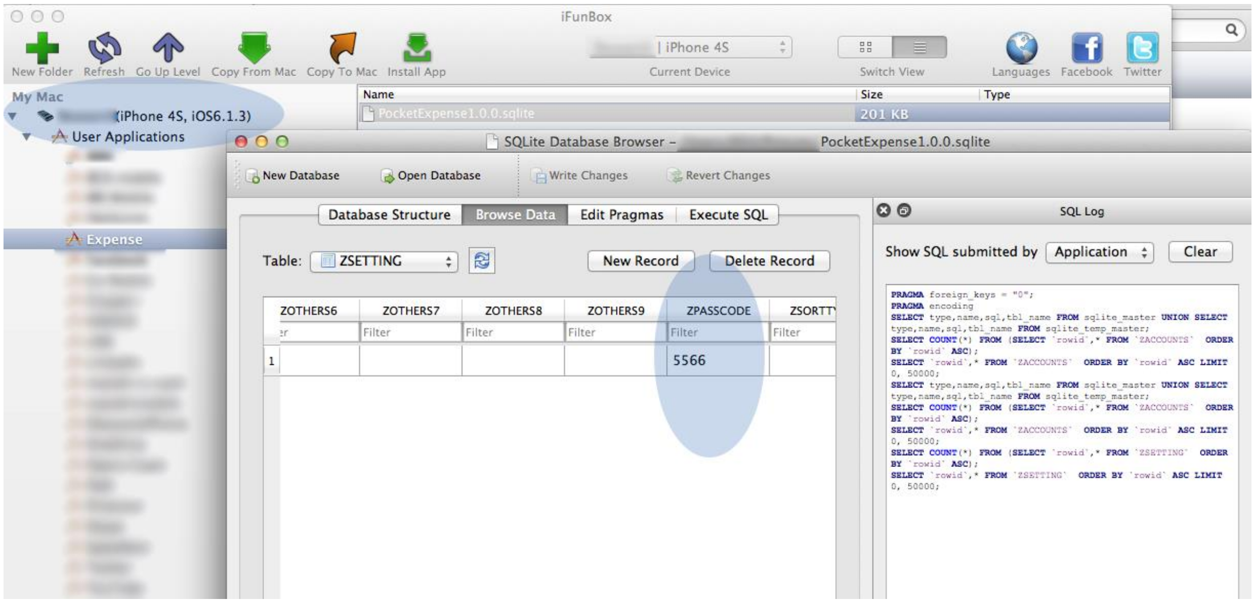
SQLite Database Browser – [redacted]/iPad Pocket Expense/PocketExpense1.0.0.sqlite

Table: ZSETTING

	RS7	ZOTHERS8	ZOTHERS9	ZPASSCODE	ZSORTTYPE	ZWEEKSTARTDAY
	Filter	Filter	Filter	Filter	Filter	Filter
1				1[redacted]	DESC	Sunday

```
PRAGMA foreign_keys = "0";
PRAGMA encoding
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER BY `rowid` ASC LIMIT
0, 50000;
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
```

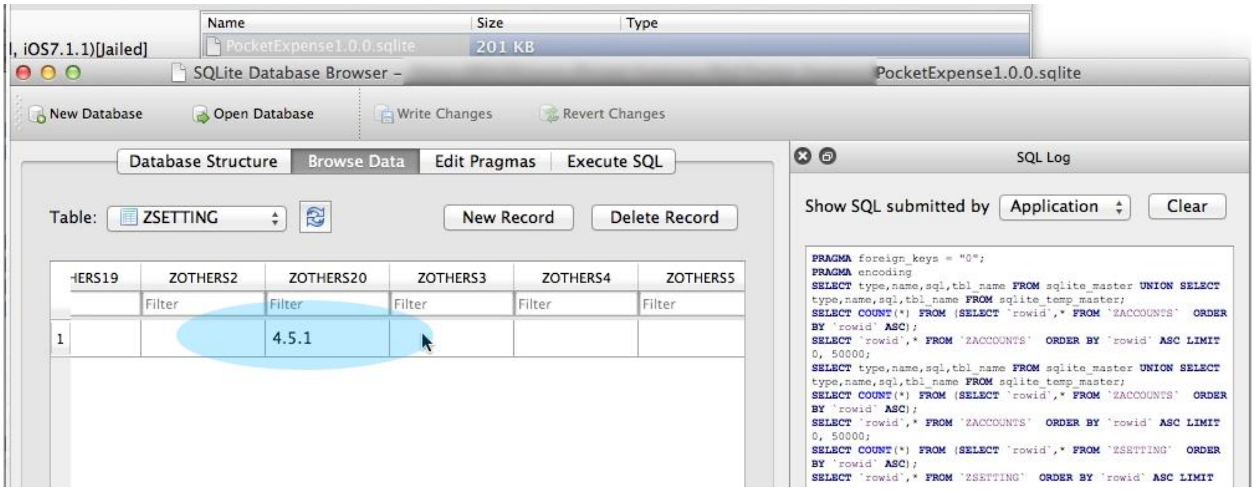
Figure 5 Passcode on iOS 7 - iPad Mini Retina Display



	ZOTHERS6	ZOTHERS7	ZOTHERS8	ZOTHERS9	ZPASSCODE	ZSORTT
	Filter	Filter	Filter	Filter	Filter	Filter
1					5566	

```
PRAGMA foreign_keys = "0";
PRAGMA encoding
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER BY `rowid` ASC LIMIT
0, 50000;
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER BY `rowid` ASC LIMIT
0, 50000;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZSETTING`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZSETTING`  ORDER BY `rowid` ASC LIMIT
0, 50000;
```

Figure 6 Passcode on iOS 6 - iPhone 4s



	HERS19	ZOTHERS2	ZOTHERS20	ZOTHERS3	ZOTHERS4	ZOTHERS5
		Filter	Filter	Filter	Filter	Filter
1			4.5.1			

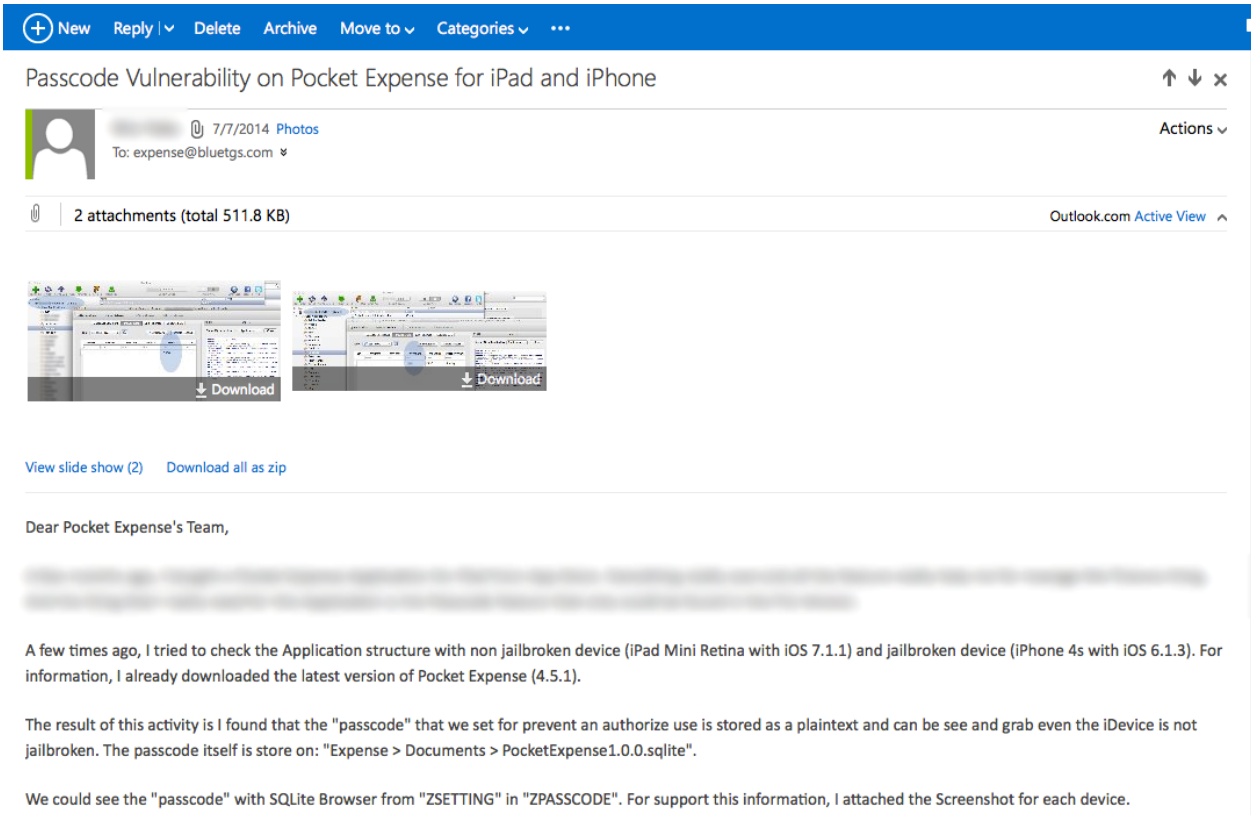
```
PRAGMA foreign_keys = "0";
PRAGMA encoding
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER BY `rowid` ASC LIMIT
0, 50000;
SELECT type,name,sql,tbl_name FROM sqlite_master UNION SELECT
type,name,sql,tbl_name FROM sqlite_temp_master;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZACCOUNTS`  ORDER BY `rowid` ASC LIMIT
0, 50000;
SELECT COUNT(*) FROM (SELECT `rowid`,* FROM `ZSETTING`  ORDER
BY `rowid` ASC);
SELECT `rowid`,* FROM `ZSETTING`  ORDER BY `rowid` ASC LIMIT
```

Figure 7 Pocket Expense v.4.5.1

V. RESPONSE FROM DEVELOPER

There has been no responses from the developer side to the e-mail that contains vulnerability from Pocket Expense application which is sent by The Writer on July 7th, 2014. However, uniquely the developer side has fixed and updated their both applications (for iPhone and iPad) in App Store on July 16th, 2014. The most disappointing thing is, the user which has paid for 4.5.1 version doesn't get the update to the related vulnerability and even this version has been removed by the developer. In other words, user has to buy the application that has been updated to the 5 version one more time.

However, the positive thing of this is the user doesn't need to worry about the problem anymore.



Passcode Vulnerability on Pocket Expense for iPad and iPhone

[redacted] 7/7/2014 Photos

To: expense@bluetgs.com

2 attachments (total 511.8 KB)

View slide show (2) Download all as zip

Dear Pocket Expense's Team,

[redacted]

A few times ago, I tried to check the Application structure with non jailbroken device (iPad Mini Retina with iOS 7.1.1) and jailbroken device (iPhone 4s with iOS 6.1.3). For information, I already downloaded the latest version of Pocket Expense (4.5.1).

The result of this activity is I found that the "passcode" that we set for prevent an authorize use is stored as a plaintext and can be see and grab even the iDevice is not jailbroken. The passcode itself is store on: "Expense > Documents > PocketExpense1.0.0.sqlite".

We could see the "passcode" with SQLite Browser from "ZSETTING" in "ZPASSCODE". For support this information, I attached the Screenshot for each device.

Figure 8 Bug Report to Developer

VI. SUMMARY AND RECOMMENDATION

Generally, with using these vulnerabilities, the Attacker will absolutely gain access to the data which is saved in the related application. Besides that, these vulnerabilities can also harm the iDevice user that is accustomed to equate the value of the Passcode.

As example, Passcode that is used by user in this Pocket Expense application is 5566. It's not impossible for the users to use the same simple Passcode for their iDevice, which means it can be resulted in misuse and brings greater impacts.

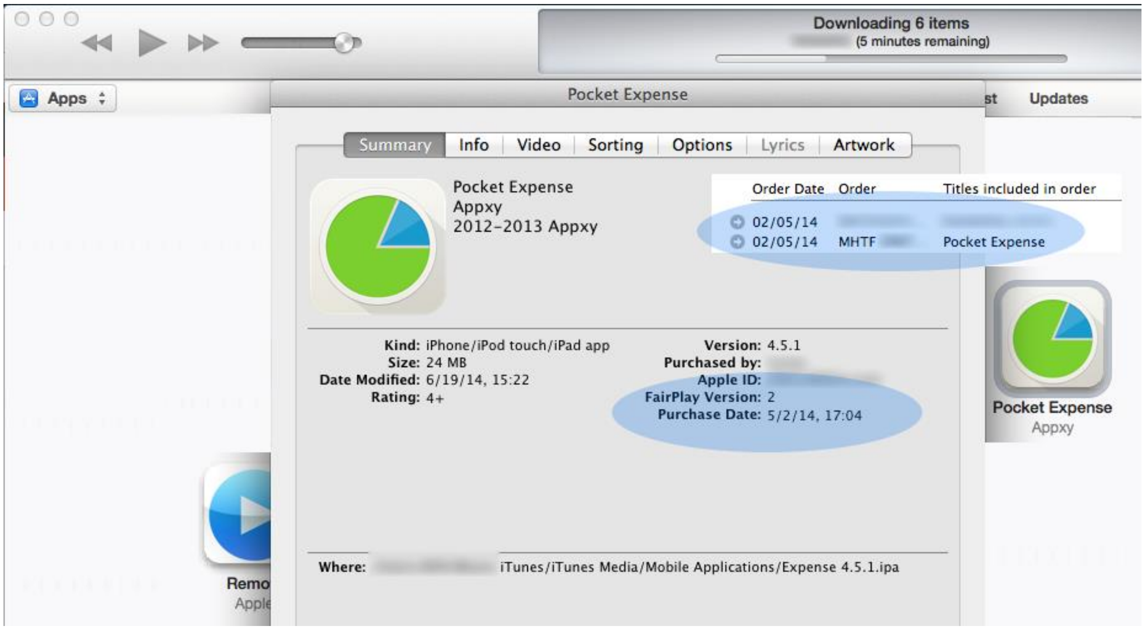
With seeing the status related to this, so the recommendations which can be implemented by the users if they won't buy the newest version of this application, are:

1. Turning off the "simple passcode" feature if the users are more accustomed to use one kind of combination for every identities.
2. If the users still want to use the "simple passcode", so the users are recommended to differentiate the Passcode that is used to login to the iDevice with the Passcode that is used to login to the Pocket Expense application.

And if the users don't mind to buy the newest version, of course the vulnerabilities aren't being worries anymore, considering that the Pocket Expense no longer use the "Passcode" value according to the description in this simple paper.

VII. ADDITIONAL INFORMATION

1. Writes used the Paid- Pocket Expense with 4.5.1 version for iPad and iPhone that the Writer bought from App Store back in the May, 2014.



Pocket Expense
Appxy
2012-2013 Appxy

Order Date	Order	Titles included in order
02/05/14	[redacted]	[redacted]
02/05/14	MHTF [redacted]	Pocket Expense

Kind: iPhone/iPod touch/iPad app
Size: 24 MB
Date Modified: 6/19/14, 15:22
Rating: 4+

Version: 4.5.1
Purchased by: [redacted]
Apple ID: [redacted]
FairPlay Version: 2
Purchase Date: 5/2/14, 17:04

Where: [redacted] iTunes/iTunes Media/Mobile Applications/Expense 4.5.1.ipa

Figure 9 Purchase Date and Purchase History

2. The logo color of the Paid- Pocket Expense is differentiated to the free one. If the free one has red logo, the paid one has blue logo.

3. Paid- Pocket Expense for iPad (v.4.5.1) hasn't got update anymore. This thing can be seen by opening our software list in iTunes software and accessing Pocket Expense in App Store through the downloaded list.

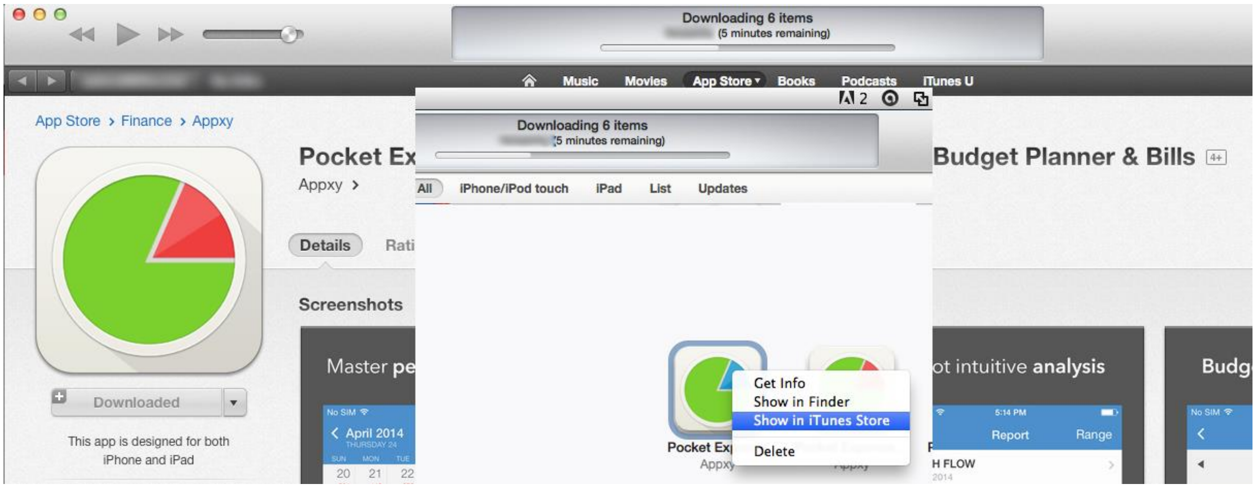


Figure 10 Show in iTunes Store

4. Writer doesn't know the exact release schedule of the 5th version, because according to the 5th paid version in App Store, the release schedule was in May, 2014. However the reparation for the free version is shown on July 16th, 2014 which is updated from the v.4.5.1 to the 5th version.