My Wallet's Got a Pin Hole

The Google Wallet has attracted a lot of attention although most of it not too complementary. The fact there aren't very many, just on one phone the Google Nexus S (made by Samsung) and one mobile operator, Sprint, and one card representation (apart from Google's own prepaid card), the Citibank MasterCard (PayPass) doesn't stop the hype. So the story that has been circulating in the last few weeks surrounds the insecurity of the Google Wallet PIN and how it can easily be bypassed.

It is interesting that the other 3 major US network operators, AT&T, T-Mobile and Verizon don't allow the Google app on their networks. Nothing I'm sure to do with the fact that they are sponsoring ISIS to the tune of at least $100m which is a competitor to the Google wallet and NFC payments. Anyway even with this lack of adoption the pundits have now discovered that Google's claims of security are rather lacking, the PIN that controls access to your wallet and in theory all your credit and debit cards can easily be circumvented - oh dear!

First of all we need to appreciate that the Google phone is based on the Android Gingerbread V2.3 and depending on updates is currently around version 2.3.4. Conceptually each app runs in its own sandbox enforced by a security kernel that stops one app accessing the data of another.

Unfortunately developers love these Android smart phones and can't wait to take root privileges of the OS at which point you can breach the sandbox and have a look at the data used by the Google Wallet app. However and here is a fundamental point if you don't 'root' your phone and don't put malware on your phone then the discussion of the attack that follows cannot be applied.

I think ViaForensics started the ball rolling by analysing the data stored in the sqlite database managed by the Google wallet app. They discovered http://viaforensics.com/mobile-security/forensics-security-analysis-google-wallet.html that the phone database stored lots of information relating to the registered credit card including the name on the card, the expiry date and the registered eMail account. It also contained all the tracking data about the transactions that have used the Google Wallet which is what Google needs to sell to their advertisers.

But it gets worse, zvelo https://zvelo.com/ carried on analyzing the wallet data and discovered that the PIN was stored as a SHA256 encoded Hash. Since the PIN is only 4 digits it doesn't take very long to do a brute force search without having to make any attempted submissions to the app.

As if this wasn't enough in the other attack, http://thesmartphonechamp.com/second-major-security-flaw-found-in-google-wallet-rooted-or-not-no-one-is-safe-video/ details how anybody with a Google phone in hand (don't lend your phone to anybody) can clear the data associated with the Wallet from the phone's application settings menu.

The next time the wallet app is launched it resets itself and ask for a new PIN. The holder of the phone can simply create a new PIN and associate a new card to the app in order to access all the previous card information and available funds. For this attack you don't even need root access to the phone.

Although I've no doubt Google can tidy up some of these problems and in particular the residue data problem referred to above there is an intrinsic problem here that is not going to be easily solved by Google or anybody else come to that.

The fundamental weakness in any programmable terminal is that hackers can gain unauthorized access to the programs and data in that device if they are able to get their malware into that terminal.

The only absolute way to prevent malware is to not allow any executable code to be loaded onto the terminal (phone, POS, PC, game terminal, etc). In other words at the point of manufacture the device needs to be sealed and not allowed to install any additional software.

On a PC at least you will probably think about peripheral drivers, almost a fundamental way of life, a new printer, scanner, whatever and there are drivers. How about all those video codecs, the list goes on.

And then how about all those software updates to improve security, patches whatever you want to call them, can you see any security vulnerabilities here?

As mobile phones become more like PCs and certainly in the case of a tablet the lines are blurred then all these vulnerabilities equally apply.

I'm sure you can see it coming, the problems for the corporate IT department are about to increase exponentially. Their users are demanding to use these smart phones and of course they want to have their own apps........

I would like to propose that these two requirements are mutually exclusive, you cannot maintain security and allow users to load executable code onto their devices. Even the IT department can make mistakes but we all know that users are going to make many more.

Many people have been quick to point out that the secure element is the solution but as you can see here in the Google Nexus S it is not the secure element that is on risk (well it is but for a different reason).

If you can access the secure element then you can do whatever it allows, putting access control in the phone doesn't get you very far because the hacker will move his attention to the stored credentials (as happened here) in the phone memory.

For the avoidance of doubt if you have malware in the phone then any PIN typed in by the user can conceptually (and in time) be intercepted as can the data displayed on the screen which can be equally manipulated so you don't actually know what the phone is doing.

The GSMA have been pinning their hope on having a 'Trusted Execution Environment' in the phone to stop these sorts of problems, ARM created 'Trusted Zone' at least 10 years ago and lots of PCs have 'Trusted Platform Modules' on their motherboard. I equally remember singing the praises of Trusted Platform 10 years ago or more but in practice none of this is really happening in practice because it interferes with the user experience. Hands up those who remember software dongles that had to be plugged into the serial port in order for the software to run. Users of PCs, phones, tablets like the freedom to do their own thing and I can't see them giving that up in a hurry. Don't close the barn door, the horse has long since bolted!

By Dr David Everett - January 2012