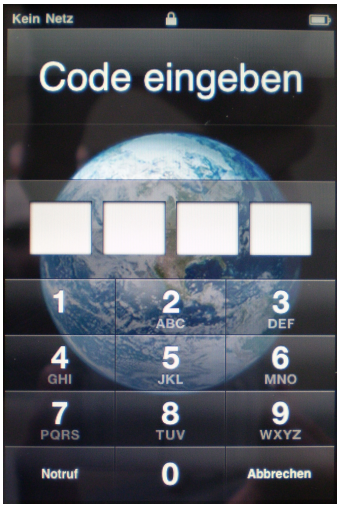
Mobile Security
Data Security on the iPhone

Companies use smartphones like the Apple iPhone more and more frequently for their field staff. But what about the safety of the data?



Christian Eichelmann
Dipl.-Inform. Sebastian Schreiber

February 3, 2010

1 Introduction

The number of smartphones on the market constantly increases. The devices are more and more frequently used by field workers in a number of companies. The reasons for their use can easily be comprehended. These mobile companions enable field workers to check their e-mails in a convenient way and help immensely to synchronize data and appointments - thus they save a tremendous load of work. In order to be able to do this, the devices have to be able to store sensitive data which should not fall into the hands of unauthorized people. Therefore, modern smartphones – like the iPhone 3GS, for instance – offer hardware encryption and the use of a lock code which deletes all data on the device after repeated invalid code insertion.

2 The Scenario

An iPhone 3GS with the current firmware version 3.1.2 serves as the exemplary configuration. On the device an Exchange account has been set up via which both e-mails, calendar and contact data can be synchronized. The connection to the Exchange server is SSL-protected. On the device itself there is a four-digit passcode which is queried when starting the device or after one minute of inactivity. When inserting the wrong code for three times, the device is turned inoperative. It is suggested that the attacker will find the smartphone switched on and locked.

3 Switching Off the Lock Code

In order to deactivate the lock code, a *jailbreak* is executed on the device. In order to do this, the software REDSN0W by the iPhone Dev Team[1] is used. This software is freely accessible on the Internet and exploits a security hole in the USB implementation of Apple on the iPhone3GS presently in use. When pressing the `Home` and the `Power` button at the same time for a little longer, the device is put into recovery mode which also works when the lock is activated. Straight afterwards REDSN0W installs a few supplementary programmes and registers the afc2 service needed for the attack, which then makes it possible to access the entire file system of the iPhone via the USB interface. All user data remain untouched during this process.

[1] http://blog.iphone-dev.org

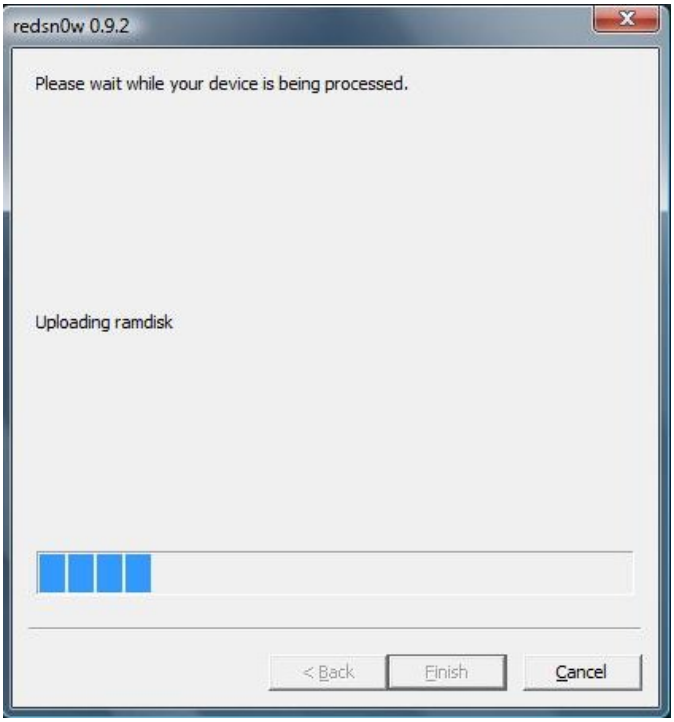


Abbildung 1: *Jail breaking* with the programme REDSN0W.

Abbildung 2: The firmware of the iPhone is manipulated.

In the next step, one uses a programme like the iPhoneBrowser[2], for instance, in order to access the file system of the iPhone. By doing this, the file `/var/Keychains/keychain-2.db` can be copied from the iPhone to one's own system. This file contains the passwords stored on the iPhone including the set password in encrypted form. It is a SQLite3 database, which can be opened and executed with the respective programme.

The command `DELETE FROM genp WHERE acct='DeviceLockPassword';` deletes the set password from the database but leaves all other stored passwords untouched.

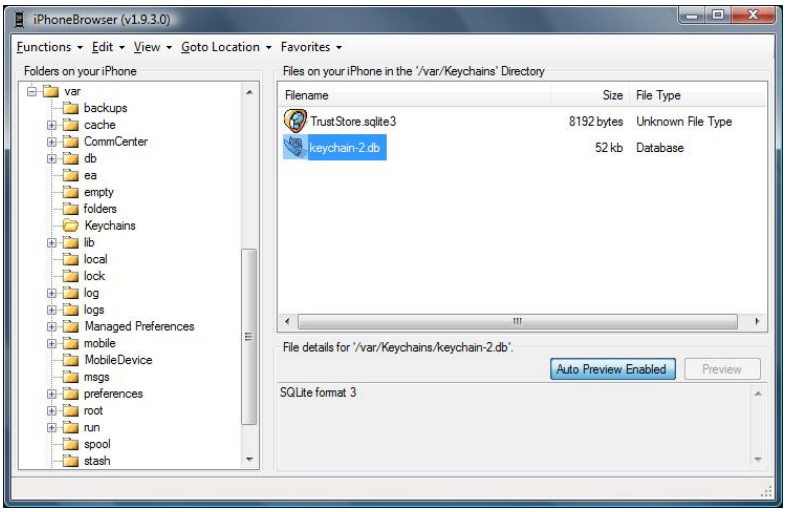


Abbildung 3: Via the iPhoneBrowser, files can be manipulated on the iPhone.

Now the manipulated database is copied back onto the iPhone. After a reboot of the device, the lock code is not requested anymore and one can access the data of the system. All synchronized e-mails, contact data and appointments can now be monitored.

4 Spying Out Access Data

The password for the Exchange server is stored in encrypted form in the database. If a perpetrator now wants to get hold of it, he can use a well-known technique: *ARP Spoofing*. First, he connects the iPhone with his own WLAN. Afterwards, he launches a *Man-in-the-Middle* attack from a computer in the respective network, for instance, with the help of the programme Ettercap[3]. Now the data traffic of the iPhone is conveyed via this computer. The attacker now starts to synchronize the Exchange data on the iPhone. When establishing the SSL connection, a faked certificate is sent from the *MitM*

[2] http://code.google.com/p/iphonebrowser
[3] http://ettercap.sourceforge.net

programme in use to the device. The respective warning of the iPhone is simply accepted by the attacker. Now, all data are encrypted with the certificate of the *MitM* programme and can in the same way be decrypted again. Thus, one can get hold of the Exchange password in use without much effort.

5 Conclusion

If an iPhone with a set passcode is lost, an attacker can access the data stored on the device with freely accessible programmes from the Internet and a relatively small amount of time. The success of such an attack depends on the firmware of the device in use and the availability of a *jailbreak* activating the afc2 service. The firmware 3.1.3 will remedy the security hole exploited by REDSN0W. We can just wait whether and when a new hole will be found enabling us to bypass the security functionality again.

Sources

[1] Current version of the *jailbreaking* tool by the iPhone Dev-Team, `http://wikee.iphwn.org/howto:rs9`

[2] Security hole on the iPhone exploited by REDSN0W, `http://www.theiphonewiki.com/wiki/index.php?title=Usb_control_msg(0x21%2C_2)_Exploit`

[3] Database format, which is used for storing passwords on the iPhone, `http://www.sqlite.org/`

[4] Windows Programme in order to manipulate files on the iPhone, `http://code.google.com/p/iphonebrowser/`

[5] Programme in order to launch *Man-in-the-Middle* attacks, `http://ettercap.sourceforge.net/`