Analysis of iOS.GuiInject Adware Library

sentinelone.com/blog/analysis-ios-guiinject-adware-library

November 8, 2016

There ain't no such thing as a free $2 cracked software, especially if one went all-in buying that latest top-notch iOS device!

Our crackers *friends* know, and if one takes the bet to use an out-of-date insecure jailbroken iOS version to install cracked software, he or she is likely to be abused by the ads injector we describe in this post.

Adware masquerading already targeted Android or iOS. These cracks use poor security practices that could lead to *spyware masquerading*.

iOS Application File

Recently, an *iOS App file* got our attention on VirusTotal:

```
$ openssl dgst -sha256 com.mailtime.MailTimePro-clutch2.ipa
SHA256(com.mailtime.MailTimePro-clutch2.ipa)=
332cf0a45170d6787dcbefb086f5a5f0f6e920d485e377fe37e900a01c128c8e
```

The filename reveals we are probably dealing with a cracked software, using Clutch, an iOS cracking toolbox. Let's decompress the iOS App file (it is a Zip archive):

```
$ ditto -xk com.mailtime.MailTimePro-clutch2.ipa com.mailtime.MailTimePro-clutch2

$ cd "com.mailtime.MailTimePro-clutch2/Payload/MailTime Pro.app/"

$ find . -type f -exec file {} \; | grep "Mach-O"
(...)
./jailbreak: Mach-O universal binary with 2 architectures
./jailbreak (for architecture armv7): Mach-O dynamically linked shared library arm
./jailbreak (for architecture arm64): Mach-O 64-bit dynamically linked shared library
./MailTime Pro: Mach-O universal binary with 2 architectures
./MailTime Pro (for architecture armv7): Mach-O executable arm
./MailTime Pro (for architecture arm64): Mach-O 64-bit executable
```

That `jailbreak` shared library looks suspicious.

```
$ codesign -dvv "MailTime Pro" 2>&1 | grep Authority
Authority=iPhone Developer: nguyen tat hung (T99T9WYY54)
Authority=Apple Worldwide Developer Relations Certification Authority
Authority=Apple Root CA

$ codesign -dvv jailbreak 2>&1 | grep Authority
Authority=iPhone Developer: nguyen tat hung (T99T9WYY54)
Authority=Apple Worldwide Developer Relations Certification Authority
Authority=Apple Root CA
```

While nguyen tat hung is a known developer identifier, he is not the expected MailTime developer.

Going further, we find `jailbreak` dynamic library is added to the original binary imports:

```
$ otool -arch arm64 -L "MailTime Pro"
(...)
    @executable_path/jailbreak (compatibility version 0.0.0, current version 0.0.0)
```

Most libraries contain their installation *prefix* and *real name* in their imports:

```
$ otool -arch arm64 -L jailbreak
jailbreak:
    /usr/local/lib/libguiinject.dylib (compatibility version 1.0.0, current version 1.0.0)
(...)
```

And thanks to `assert()` macros, project-related information are leaking:

```
$ strings -arch arm64 jailbreak | grep -i guiinject
/Users/gtt/Documents/workspaceIOS/guiinject/guiinject/MBProgressHUD.m
/Users/gtt/Documents/workspaceIOS/guiinject/guiinject/SSZipArchive.m
guiinject
```

Injected Library

Browsing the symbols, we quickly isolate well-known SDKs and Cocoa Pods:

So many ads stuff for a jailbreak is unusual. Here you have the remaining classes headers.

We run most of the cracked apps on a jailbroken device without problems.

`-[Config getConfig]` loads `wrap.json` from the host program resources:

```
{
  "udid": "jailbreak",
  "wait_loop": "3",
  "is_jb": "1",
  "package_name": "com.mailtime.MailTimePro"
}
```

After several `wait_loop` launch, the injected code will contact an insecure remote host. Half of the services seem to work.

For example, the `coreapi` service seems down:

`http://wrapper.laughworld.net/coreapi/active_device.php?pk=IPANAME&is_jb=1&udid=REDACTED&signature=MD5` :

```
{
  "return": -2,
  "message": "DB operator fail!"
}
```

`http://wrapper.laughworld.net/coreapi/get_list_message.php?pk=IPANAME&is_jb=1&udid=REDACTED&libver=20160818&app_pk=IPANAME_AGAIN&app_ver=1.2.3&signature=MD5` :

```
{
  "return": 0,
  "messages": []
}
```

While the `api` service is up:

`http://wrapper.laughworld.net/api/com.mailtime.MailTimePro_ads.json` :

```
{
  "advertising_list": [
    {
      "id": 1,
      "act_type": "0",
      "b": "<body><iframe style='border:none;padding-left:0px;padding-top:0px;'
src='http://bypassfirewall.net'><p>http://bypassfirewall.net/</p></iframe></body>",
      "dp_type": "1",
      "url": "http://bypassfirewall.net",
      "hide": 1,
      "random_show": "5",
      "adsnet_name": "admob",
      "adsnet_id": "ca-app-pub-3816529472258726/8039356495"
    }
  ]
}
```

Most interesting part is the *update* request:

`-[API getUpdate:withSelector:]` connects to `http://wrapper.laughworld.net/api/com.mailtime.MailTimePro_update.json` :

```
{
  "show_ads": "YES",
  "show_message": "YES",
  "update_message_not": "",
  "update_link": "http://google.com",
  "linkfw": "http://wrapper.laughworld.net/lib/DailyUploadDownloadLib.framework.zip",
  "namefw": "DailyUploadDownloadLib.framework",
  "md5fw": "f6a51b479516f11ce503ae06f9ffff0f",
  "script_zip": "http://wrapper.laughworld.net/lib/filehost.scr.zip",
  "script_file": "filehost.scr",
  "md5_script": "a9ef52dc75ecbcfce9447237f5154417"
}
```

`script_zip`, `script_file` and `md5_script` are not implemented. The `script_zip` URL points to a password encrypted Zip file, and the `md5_script` value is not valid (last bytes are wrong).

`linkfw` points to a valid Zip archive. Once downloaded and decompressed, `-[guiinject _loadPluginAtLocation:]` will `load` the framework and send a `run` message to its `principalClass`. `md5fw` value is used for self-update.

Ads are downloaded, but so far, we don't see any of them. They are probably rendered in a hidden view.

Downloaded Framework

```
$ openssl dgst -sha256 DailyUploadDownloadLib
SHA256(DailyUploadDownloadLib)= 00ca48ebeda3d93ccf1b8b405fcf4c2062424bbc99425e27f0b65c7ee238780e

$ file DailyUploadDownloadLib
DailyUploadDownloadLib: Mach-O universal binary with 2 architectures
DailyUploadDownloadLib (for architecture armv7): Mach-O dynamically linked shared library arm
DailyUploadDownloadLib (for architecture arm64): Mach-O 64-bit dynamically linked shared library
```

The developer identifier is different:

```
$ codesign -dvv DailyUploadDownloadLib 2>&1 | grep Authority
Authority=iPhone Developer: Pham Hiep (8DYXPR6ZBP)
Authority=Apple Worldwide Developer Relations Certification Authority
Authority=Apple Root CA
```

The *User* and *Organization* names are in the framework header:

```
$ cat Headers/DailyUploadDownloadLib.h
//
// DailyUploadDownloadLib.h
// DailyUploadDownloadLib
//
// Created by GTT Media Hanoi on 9/29/16.
// Copyright © 2016 T&B. All rights reserved.
(...)
```

This file was automatically generated by Xcode. According to *ITviec* jobs website, GTT Media and T&B are outsourcing companies.

Once loaded, the framework ask a `lib` service for two lists of files, on *DailyUploads* and *FileFactory* file-hosting sites.

`http://wrapper.laughworld.net/lib/DailyUploadDownloadModule.conf`:

```
{
  "list": [
    "https://dailyuploads.net/hdo3rn24n5tg",
    "https://dailyuploads.net/udzjx12rvu0z",
    "https://dailyuploads.net/8buwsi2hk9x7",
    "https://dailyuploads.net/vgnqrv66hp4l",
    "https://dailyuploads.net/mla2ofh3c0z8",
    "https://dailyuploads.net/ud2hlgpw9dto",
    "https://dailyuploads.net/030p4rn9ll6a",
    "https://dailyuploads.net/rsjbhbc6zi0b",
    "https://dailyuploads.net/wzrqhpqa7x7w",
    "https://dailyuploads.net/dqcl45a61amy",
    "https://dailyuploads.net/rhkbzrodo6ou",
    "https://dailyuploads.net/cqrakbup91s4",
    "https://dailyuploads.net/yxskttjfo4h8",
    "https://dailyuploads.net/m6p6maeijff1",
    "https://dailyuploads.net/f15g1prokvks"
  ]
}
```

`http://wrapper.laughworld.net/lib/FileFactoryDownloadModule.conf`:

```
{
  "list": [
    "http://filefactory.com/file/ebdz39d8dex/myfile42.encrypt",
    (...)
    "http://filefactory.com/file/1ycfrbml51ox/myfile7.encrypt",
    "http://filefactory.com/file/1k97tfd8ibu5/kdiff3-0.9.98-MacOSX-64Bit.dmg",
    "http://filefactory.com/file/5difpf82yog1/Newsgroup_collection.zip",
    "http://filefactory.com/file/6mlwn7iv1mv7/docword.enron.txt.gz",
    "http://filefactory.com/file/52sg5aurgkrz/Tiny_Wings__Andreas_Illiger___v2.1_os43_-
Nitrox.rc330_84.ipa",
    "http://filefactory.com/file/6x9iujr8u6a5/php-5.6.14.tar.bz2",
    "http://filefactory.com/file/q29tth3j859/iBackupBot-Setup.dmg",
    "http://filefactory.com/file/6bcmlwfuw7wl/pokegoppsl.zip",
    "http://filefactory.com/file/4qqx4s5l36hn/iPhoneConfigUtility.dmg",
    "http://filefactory.com/file/5vqawo60iyb9/googlemobileadssdkios.zip",
    "http://filefactory.com/file/560caqad3k9h/mallet-2.0.8RC3.tar.gz",
    "http://filefactory.com/file/5ovqpwwp0w7h/609704981.ipa",
    "http://filefactory.com/file/1lpoyv8v2y73/Multiplayer_for_Minecraft_PE__v2.0_v2.012_Univ_os80_-
Locophone-ICPDA.rc333_91.ipa",
    "http://filefactory.com/file/2abv5ufb9gav/MtProtoKit-master.zip",
    "http://filefactory.com/file/5t8e4px5fod1/577499909.ipa",
    "http://filefactory.com/file/4zy55s6qayrh/intel_rst_7_mb_8.1.zip",
    "http://filefactory.com/file/12toqn6khwd3/MEAD-3.12.tar.gz"
  ]
}
```

The framework also periodically checks the external IP address of the iOS device, using *DynDNS*.

When the framework is loaded, or when the external IP address changes, a random file from both hosting sites is downloaded. It's worth noting DailyUploads and FileFactory have affiliation programs (per thousand downloads). Median file size is 15 Mb.

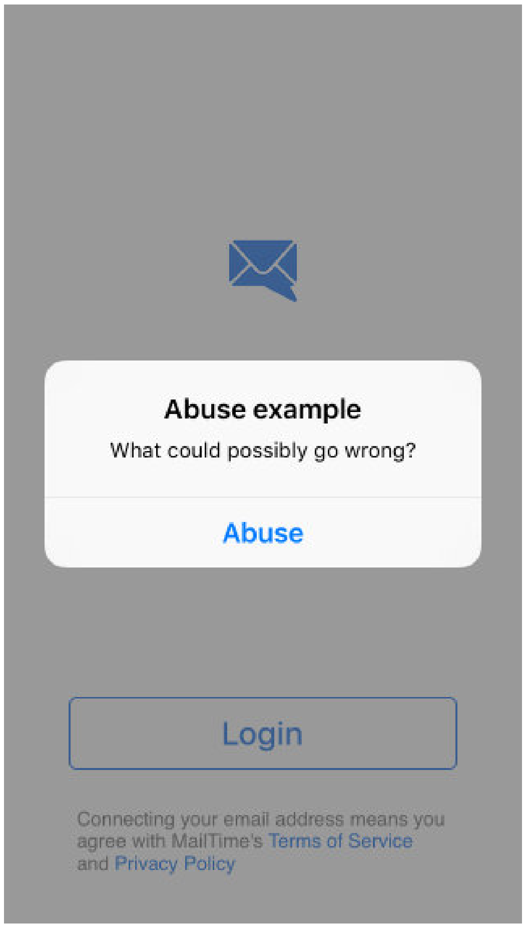
DailyUploads links point to other iOS applications. They are also signed by *nguyen tat hung* and injected:

```
$ yara -r iOS.GuiInject.yara DailyUploads
ipa_jb DailyUploads/com.infinear.call-clutch2.ipa
ipa_jb DailyUploads/com.axidep.polyglotvoicereader-clutch2.ipa
ipa_jb DailyUploads/com.contrast.mileagelog-clutch2.ipa
ipa_jb DailyUploads/com.kymatica.AUFX-Space-clutch2.ipa
ipa_jb DailyUploads/co.qapps.calcpro-clutch2.ipa
ipa_jb DailyUploads/com.pixiapps.ecoutemobile-clutch2.ipa
ipa_jb DailyUploads/com.jhnd.blender-clutch2.ipa
ipa_jb DailyUploads/com.jackadam.darksky-rc.ipa
ipa_jb DailyUploads/com.markelsoft.Text2Speech-clutch2.ipa
ipa_jb DailyUploads/com.giacomoballi.FindTower-clutch2.ipa
ipa_jb DailyUploads/com.venderbase.dd-wrt-clutch2.ipa
ipa_jb DailyUploads/com.vincenzoaccardi.itracking-clutch2.ipa
ipa_jb DailyUploads/com.realvnc.VNCViewer-clutch2.ipa
ipa_jb DailyUploads/com.yacreader.yacreader-clutch2.ipa
ipa_jb DailyUploads/com.plumamazing.iWatermark-clutch2.ipa
```

Abusing the Injected Adware Library

So, a dynamic library asks a remote host using an insecure protocol for code to execute... What could possibly go wrong?

By intercepting and modifying the `update` response, we successfully load our code:


Abuse example

What could possibly go wrong?

Connecting your email address means you agree with MailTime's Terms of Service and Privacy Policy

This is Arbitrary Code Execution using Man in the Middle attack. We should probably thank crackers, or developers, for explicitly allowing `NSAllowsArbitraryLoads` (insecure http protocol) in *App Transport Security*.

Hundreds of Samples

Using VirusTotal Retrohunt and some Yara rules, we find hundreds of samples.

Conclusion

The incentive for **jailbreak is research, not piracy**. Production devices must stay away from jailbreak and piracy.

Real **crackers never monetize** their findings, and most developers provide free versions of their products, with their own ads. Those ad revenues go to the author instead of some random selfish cracker.

And while we are on it, for the price of that latest iOS device, people could get hundreds of programs and improve their current device experience. They won't even have to feel guilty for Apple as the mothership takes its cut on App Store sales.

Conclusions regarding crackers *earned money* using hidden ads and fake downloads, users *wasted data* plans, or *potential abuses* of the injected adware library are left as an exercise to the reader.

For other deep analyses on hacks and attacks, check out SentinelOne's other Security Research blogs.

SentinelOne unifies next-generation prevention, detection, and response in a single platform to protect user endpoints and critical servers from all types of advanced attacks. Learn more about the SentinelOne Endpoint Protection Platform's technology.