iOS Pentesting - Reversing Jailbreak

2019-03-12 Trelis iOS iOS pentest reversing jailbreak

In this article jailbreak detection method is analyzed and bypassed doing reversing.

Introduction

iOS jailbreaking is privilege escalation for the purpose of removing software restrictions imposed by Apple. It typically does this by using a series of kernel patches. Jailbreaking permits root access to iOS, allowing the downloading and installation of additional applications, extension, and themes that are unavailable through the official Apple App Store.

A lot of applications allow its execution in jailbroken devices, even though having jailbreak detection, because they don’t want to lose users. For example, banks allow applications to execute in all devices but some fucntionalities might be restricted to be executed only in non-jailbroken devices.

Detection

File based detection

Content

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

During the jailbreaking process, some additional files are created on the device. Looking for these files is a simple way to detect a jailbreak. It's also an easy method for a malicious individual to detect and bypass. An attacker can search for a string in the application, and then simply change the file names in question to avoid detection.

- **Files or appplciations used**:

```
/private/var/lib/apt

/private/var/tmp/cydia.log

/private/var/lib/cydia

/private/var/mobile/Library/SBSettings/Themes

/Library/MobileSubstrate/MobileSubstrate.dylib

/Library/MobileSubstrate/DynamicLibraries/Veency.plist

/Library/MobileSubstrate/DynamicLibraries/LiveClock.plist

/System/Library/LaunchDaemons/com.ikey.bbot.plist

/System/Library/LaunchDaemons/com.saurik.Cydia.Startup.plist

/var/cache/apt

/var/lib/apt

/var/lib/cydia

/var/log/syslog

/var/tmp/cydia.log

/bin/bash

/bin/sh

/usr/sbin/sshd

/usr/libexec/ssh-keysign
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
/usr/sbin/sshd
/usr/bin/sshd
/usr/libexec/sftp-server
/etc/ssh/sshd_config
/etc/apt
/Applications/Cydia.app
/Applications/RockApp.app
/Applications/Icy.app
/Applications/WinterBoard.app
/Applications/SBSettings.app
/Applications/MxTube.app
/Applications/IntelliScreen.app
/Applications/FakeCarrier.app
/Applications/blackra1n.app
```

- **Directory permissions**: Like detecting a jailbroken device by looking for certain new files, certain permissions on partitions and folders can also indicate a jailbroken device. For example, during the jailbreaking process, access to the root partition is amended. If the root partition has read/write permissions, the device has been jailbroken.
- **Size of /etc/fstab file**: The /etc/fstab file contains mount points for the system. Many jailbreaking tools modify this file by adding entries to it, changing its file size. The typical iOS app isn't capable of reading the file, but it can check the size of the file.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Do note however, that the file size can change as a result of a new update from Apple.

- **Existence of symbolic links**: Some directories are originally located in the small system partition, however, this partition is overwritten during the jailbreak process. Therefore the data must be relocated to the larger data partition. Because the old file location must remain valid, symbolic links are created. The following list contains files or directories which would be symbolic links on a jailbroken device. An application could check for these symbolic links, and, if they exist, detect a jailbreak.

```
/Library/Ringtones

/Library/Wallpaper

/usr/arm-apple-darwin9

/usr/include

/usr/libexec

/usr/share

/Applications
```

- **Writing files**: On jailbroken devices, applications are installed in the /Applications folder and thereby given root privileges. A jailbroken device could be detected by having the app check whether it can modify files outside of its sandbox. This can be done by having the app attempt to create a file in, for example, the /private directory. If the file is successfully created, the device has been jailbroken.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

API-based detection

Some API calls provided by iOS behave differently if run on jailbroken devices. Detecting a jailbroken device based on API calls can be both effective and difficult for a malicious individual to recognize and bypass.

- **fork()**: The sandbox denies process forking on non-jailbroken devices. By checking the returned pid on fork(), an app can detect if it has successfully forked. If the fork is successful, the app can deduce that it is running on a jailbroken device.
- **system()**: Calling the system() function with a NULL argument on a non-jailbroken device will return 0. Doing the same on a jailbroken device will return 1. This is because the function will check whether /bin/sh exists, and it only exists on jailbroken devices.
- **dyld functions**: This detection method starts with calling functions like _dyld_image_count() and _dyld_get_image_name() to see what dylibs are currently loaded. This method is very difficult to dynamically patch due to the fact that the patches themselves are part of dylibs.

OpenSSH Service Detection

Jailbroken devices can run services that aren't normally present on non-jailbroken devices - the most common is the OpenSSH service.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Note that this detection method can be very slow. If SSH is not installed or running on the device, it can take some time for the connection to timeout. Attackers can also easily bypass this method by simply changing the port for the OpenSSH service.

Cydia Scheme Detection

Most jailbroken devices have Cydia installed. While an attacker can change the location of the Cydia app, it's unlikely they will also change the URL scheme the Cydia app is registered with.

If calling the Cydia's URL scheme (cydia://) from your application is successful, you can be sure that the device is jailbroken.

It's difficult to change the scheme for Cydia, but it is possible to simply remove Cydia during the testing process.

Objective-C

Objective-C calls from one method to another are compiled as calls to objc_msgSend(). One effect of this is that IDA Pro cross references do not reflect the actual functions being called at runtime. This function is defined with the following function signature:

```
id objc_msgSend(id self, SEL op,...)
```

This implies that for any Objective-C method call it is made, the first two arguments are the object's self pointer, and the selector, which is a string representation of the method being called on self.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

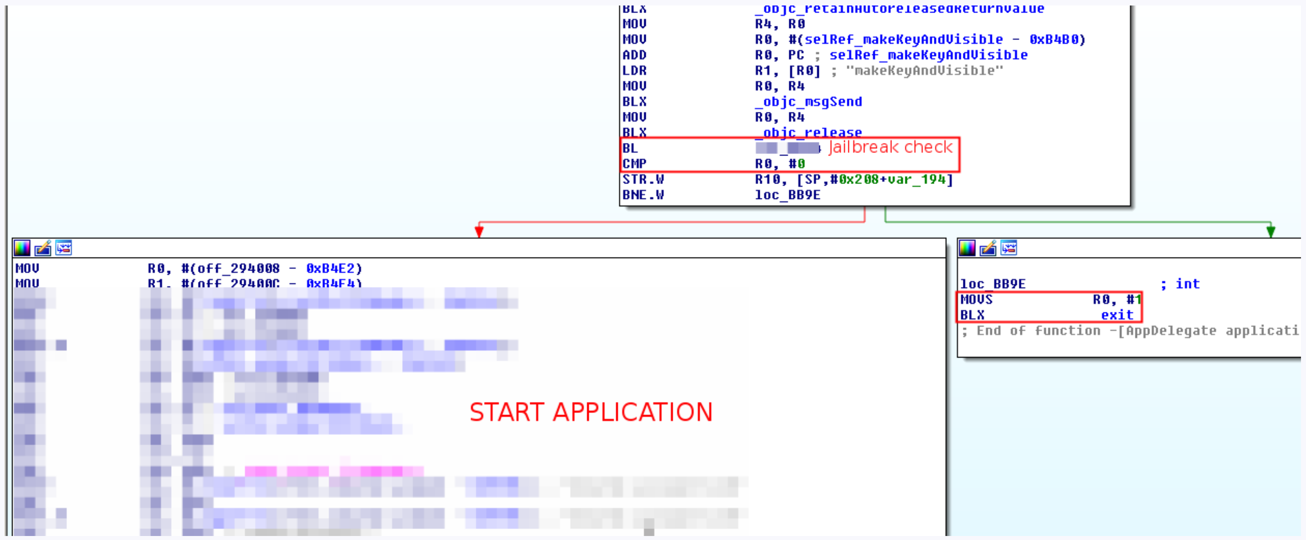
```
return_value = objc_msgSend(class_receiver, method_signature, arg1, arg2,...)
```

where:

- **return_value**: return value of the method called is stored in R0
- **class_receiver**: value should be stored in R0
- **method_signature**: value should be stored in R1
- **arg1**: value should be stored in R2
- **arg2**: value should be stored in R3

Reversing

There is a method which initialize everything so the application can start. One of the checks it does before starting is whether the application is running in a jailbroken device or not:



```
BLX             _objc_retainAutoreleasedReturnValue
MOV             R4, R0
MOV             R0, #(selRef_makeKeyAndVisible - 0xB4B0)
ADD             R0, PC ; selRef_makeKeyAndVisible
LDR             R1, [R0] ; "makeKeyAndVisible"
MOV             R0, R4
BLX             _objc_msgSend
MOV             R0, R4
BLX             _objc_release
BL              [redacted]
CMP             R0, #0
STR.W           R10, [SP,#0x208+var_194]
BNE.W           loc_BB9E
```

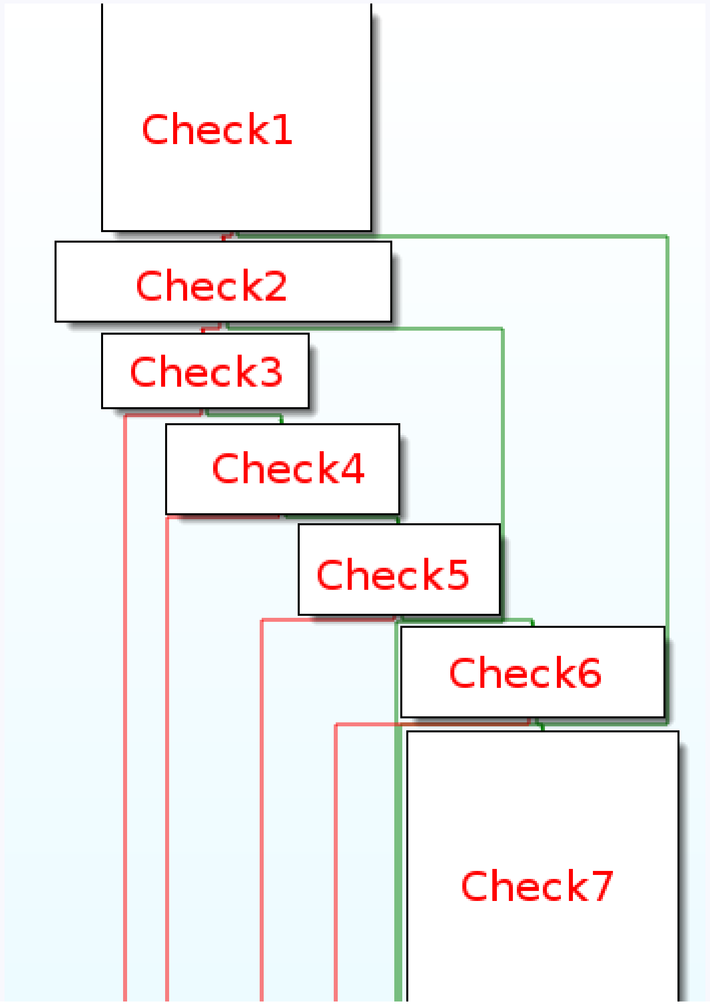
```
MOV             R0, #(off_294008 - 0xB4E2)
MOV             R1, #(off_29400C - 0xB4F4)
[redacted]
```

```
loc_BB9E                        ; int
MOVS            R0, #1
BLX             exit
; End of function -[AppDelegate applicati
```

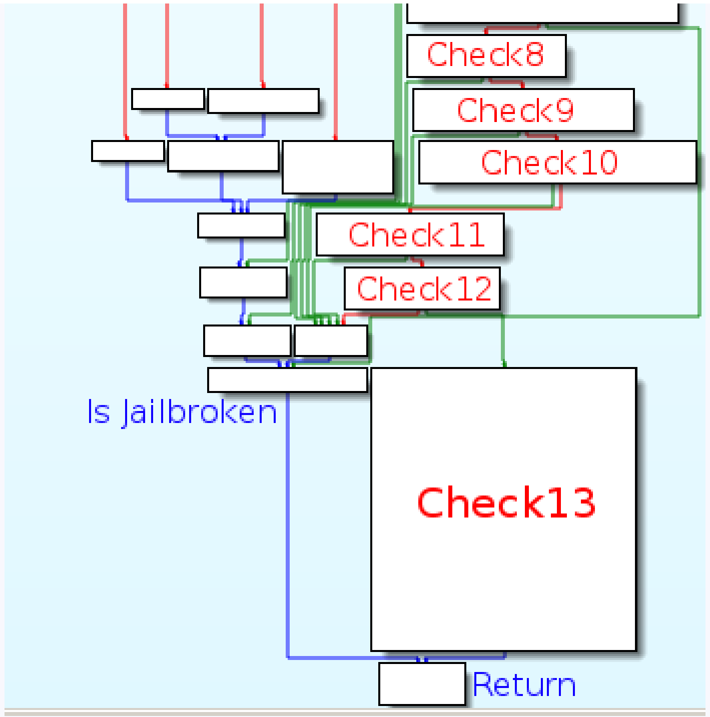
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Below we will analyze the method which is used to detect if the device is jailbroken. It has the following structure:

1. Makes N checks. The more checks it does, the more likely is to verify that the device is jailbroken. However, it is not related with the difficulty of bypassing the control.
2. If the check is negative (not jailbroken), it goes to the next check. Otherwise, it just returns that the device is jailbroken.



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

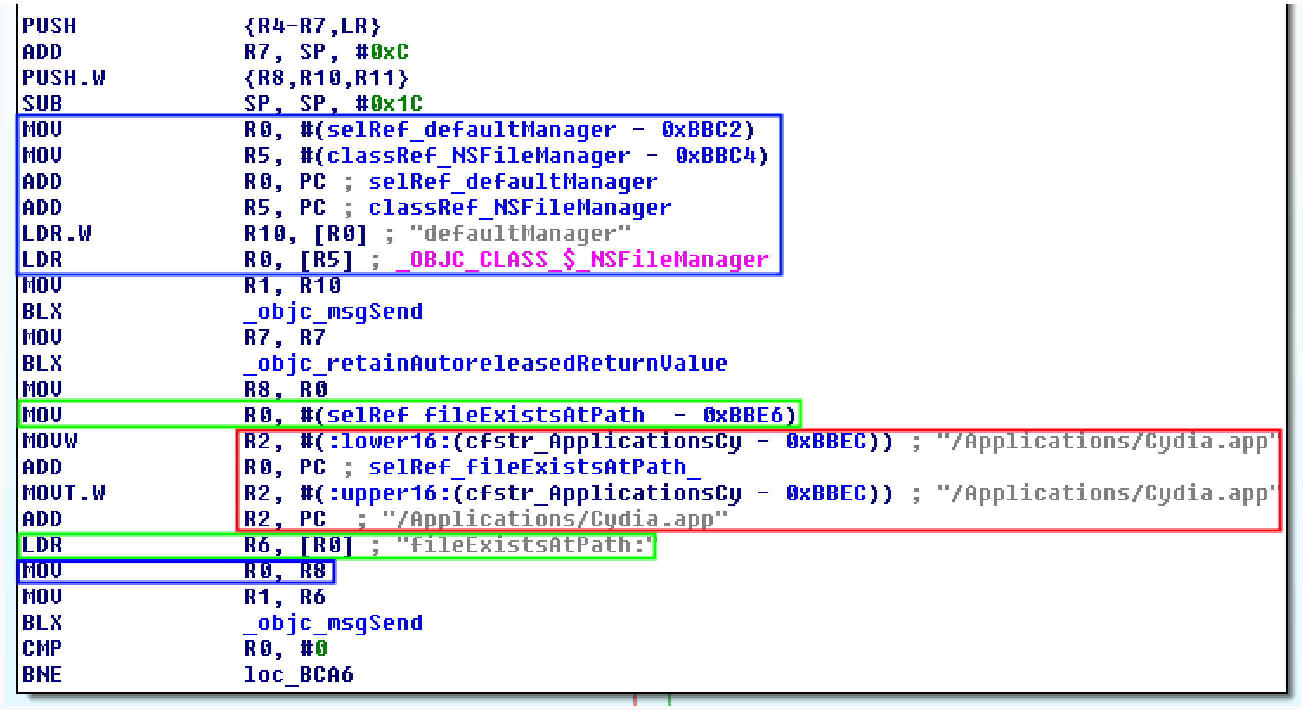


Each check will be analyze individually. They are distributed in the following categories:

- **Directory check**: check 1 to 6
- **Cydia scheme detection**: check 7
- **Files or appplciations used**: check 8 to 12
- **Writing files**: check 13

Check 1

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

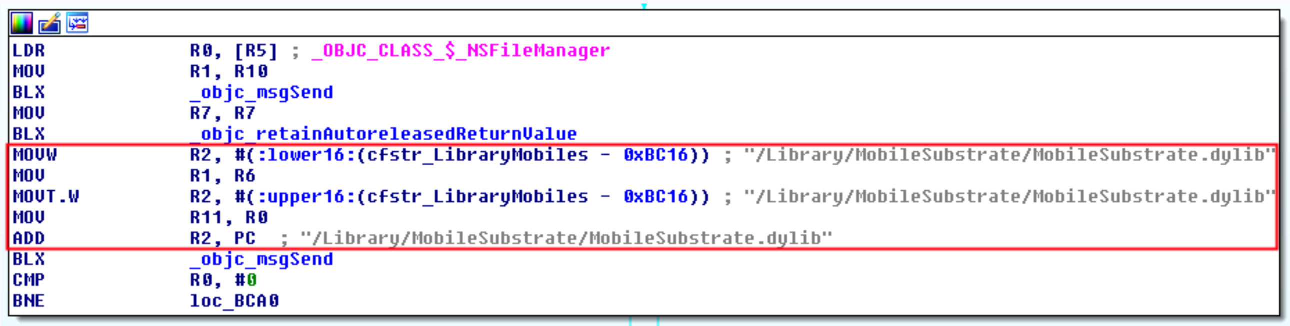


```
PUSH            {R4-R7,LR}
ADD             R7, SP, #0xC
PUSH.W          {R8,R10,R11}
SUB             SP, SP, #0x1C
MOV             R0, #(selRef_defaultManager - 0xBBC2)
MOV             R5, #(classRef_NSFileManager - 0xBBC4)
ADD             R0, PC ; selRef_defaultManager
ADD             R5, PC ; classRef_NSFileManager
LDR.W           R10, [R0] ; "defaultManager"
LDR             R0, [R5] ; _OBJC_CLASS_$_NSFileManager
MOV             R1, R10
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOV             R8, R0
MOV             R0, #(selRef_fileExistsAtPath_ - 0xBBE6)
MOVW            R2, #(:lower16:(cfstr_ApplicationsCy - 0xBBEC)) ; "/Applications/Cydia.app"
ADD             R0, PC ; selRef_fileExistsAtPath_
MOVT.W          R2, #(:upper16:(cfstr_ApplicationsCy - 0xBBEC)) ; "/Applications/Cydia.app"
ADD             R2, PC  ; "/Applications/Cydia.app"
LDR             R6, [R0] ; "fileExistsAtPath:"
MOV             R0, R8
MOV             R1, R6
BLX             _objc_msgSend
CMP             R0, #0
BNE             loc_BCA6
```

1. First of all it obtains the object of the NSFileManager class which is stored in R0 and defaultManager method stored in R1 (blue). Later, the result is moved again in R0 (MOV R0, R8).
2. Then it obtains the address of the function fileExistsAtPath which is stored in R1 (green).
3. "/Applications/Cydia.app" is stored in R2. (red)
4. Finally it calls _objc_msgSend with the attributes NSFileManager, fileExistsAtPath and "/Applications/Cydia.app" stored in the registers R0, R1 and R2 respectively.
5. It returns a boolean (R0) which is compared with 0x00 (BNE = branch if not equal). If the path does not exists, it jumps to check 2, otherwise the method will return that the device is jailbroken.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Check 2



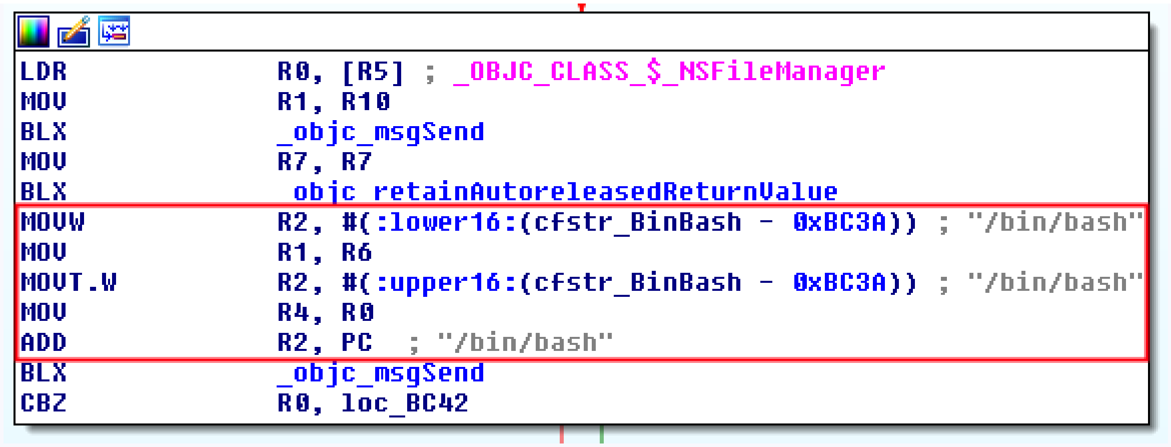
```
LDR             R0, [R5] ; _OBJC_CLASS_$_NSFileManager
MOV             R1, R10
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOVW            R2, #(:lower16:(cfstr_LibraryMobiles - 0xBC16)) ; "/Library/MobileSubstrate/MobileSubstrate.dylib"
MOV             R1, R6
MOVT.W          R2, #(:upper16:(cfstr_LibraryMobiles - 0xBC16)) ; "/Library/MobileSubstrate/MobileSubstrate.dylib"
MOV             R11, R0
ADD             R2, PC  ; "/Library/MobileSubstrate/MobileSubstrate.dylib"
BLX             _objc_msgSend
CMP             R0, #0
BNE             loc_BCA0
```

The second check if the library "/Library/MobileSubstrate/MobileSubstrate.dylib" exists. It uses the same procedure as before. Loads the object of NSFileManager class and calls the function fileExistsAtPath.

It returns a boolean (R0) which is compared with 0x00 (BNE = branch if not equal). If the path does not exists, it jumps to check 3, otherwise the method will return that the device is jailbroken.

Check 3

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



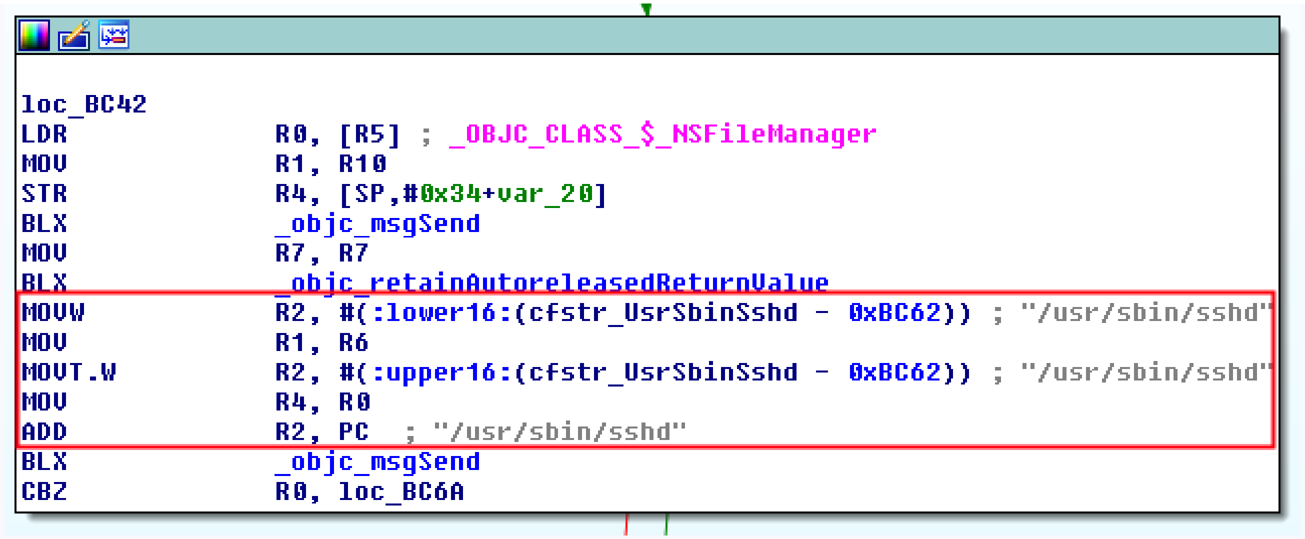
```
LDR             R0, [R5] ; _OBJC_CLASS_$_NSFileManager
MOV             R1, R10
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOVW            R2, #(:lower16:(cfstr_BinBash - 0xBC3A)) ; "/bin/bash"
MOV             R1, R6
MOVT.W          R2, #(:upper16:(cfstr_BinBash - 0xBC3A)) ; "/bin/bash"
MOV             R4, R0
ADD             R2, PC  ; "/bin/bash"
BLX             _objc_msgSend
CBZ             R0, loc_BC42
```

It checks if the path “/bin/bash” exists. It uses the same procedure as before.

It returns a boolean (R0) which is compared with 0x00 (BNE = branch if not equal). If the path does not exists, it jumps to check 4, otherwise the method will return that the device is jailbroken.

Check 4

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



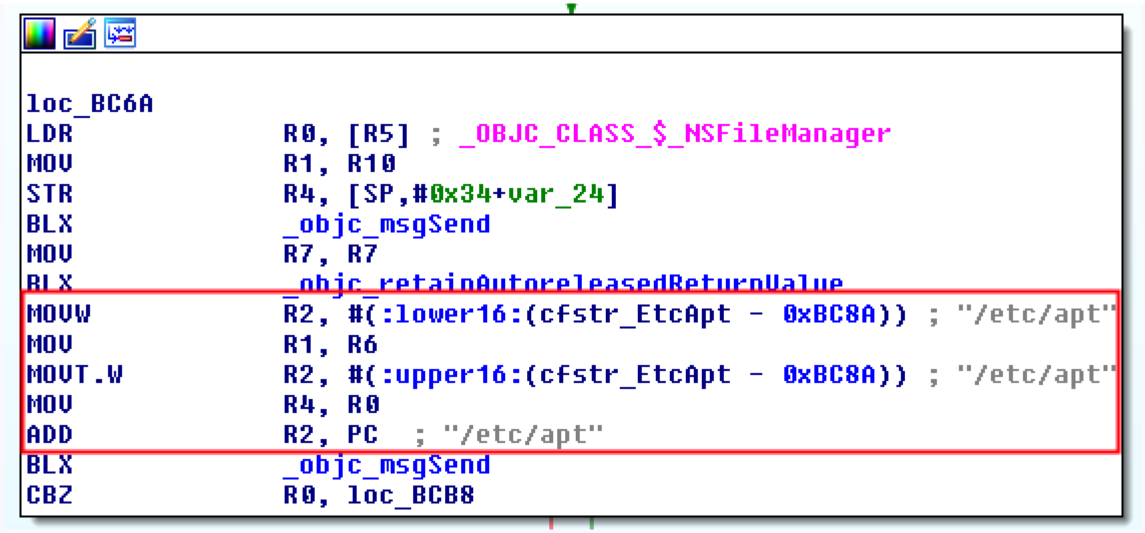
```
loc_BC42
LDR             R0, [R5] ; _OBJC_CLASS_$_NSFileManager
MOV             R1, R10
STR             R4, [SP,#0x34+var_20]
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOVW            R2, #(:lower16:(cfstr_UsrSbinSshd - 0xBC62)) ; "/usr/sbin/sshd"
MOV             R1, R6
MOVT.W          R2, #(:upper16:(cfstr_UsrSbinSshd - 0xBC62)) ; "/usr/sbin/sshd"
MOV             R4, R0
ADD             R2, PC  ; "/usr/sbin/sshd"
BLX             _objc_msgSend
CBZ             R0, loc_BC6A
```

It checks if the path “/usr/sbin/sshd” exists. It uses the same procedure as before.

It returns a boolean (R0) which is compared with 0x00 (CBZ = Compare and Branch on Zero). If the path does not exists, it jumps to check 5, otherwise the method will return that the device is jailbroken.

Check 5

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



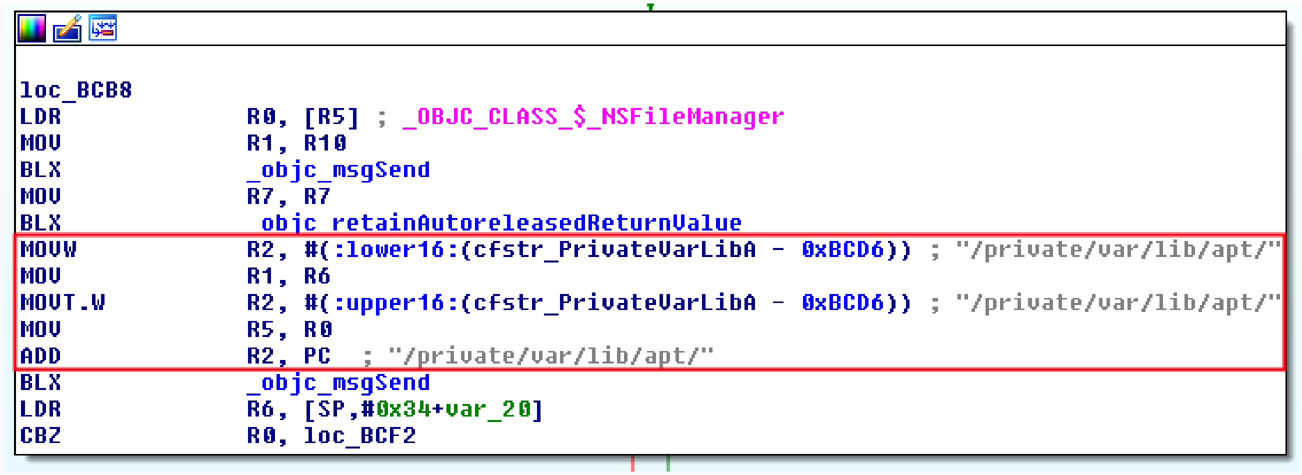
```
loc_BC6A
LDR             R0, [R5] ; _OBJC_CLASS_$_NSFileManager
MOV             R1, R10
STR             R4, [SP,#0x34+var_24]
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOVW            R2, #(:lower16:(cfstr_EtcApt - 0xBC8A)) ; "/etc/apt"
MOV             R1, R6
MOVT.W          R2, #(:upper16:(cfstr_EtcApt - 0xBC8A)) ; "/etc/apt"
MOV             R4, R0
ADD             R2, PC  ; "/etc/apt"
BLX             _objc_msgSend
CBZ             R0, loc_BCB8
```

It checks if the path “/etc/apt” exists. It uses the same procedure as before.

It returns a boolean (R0) which is compared with 0x00 (CBZ = Compare and Branch on Zero). If the path does not exists, it jumps to check 6, otherwise the method will return that the device is jailbroken.

Check 6

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



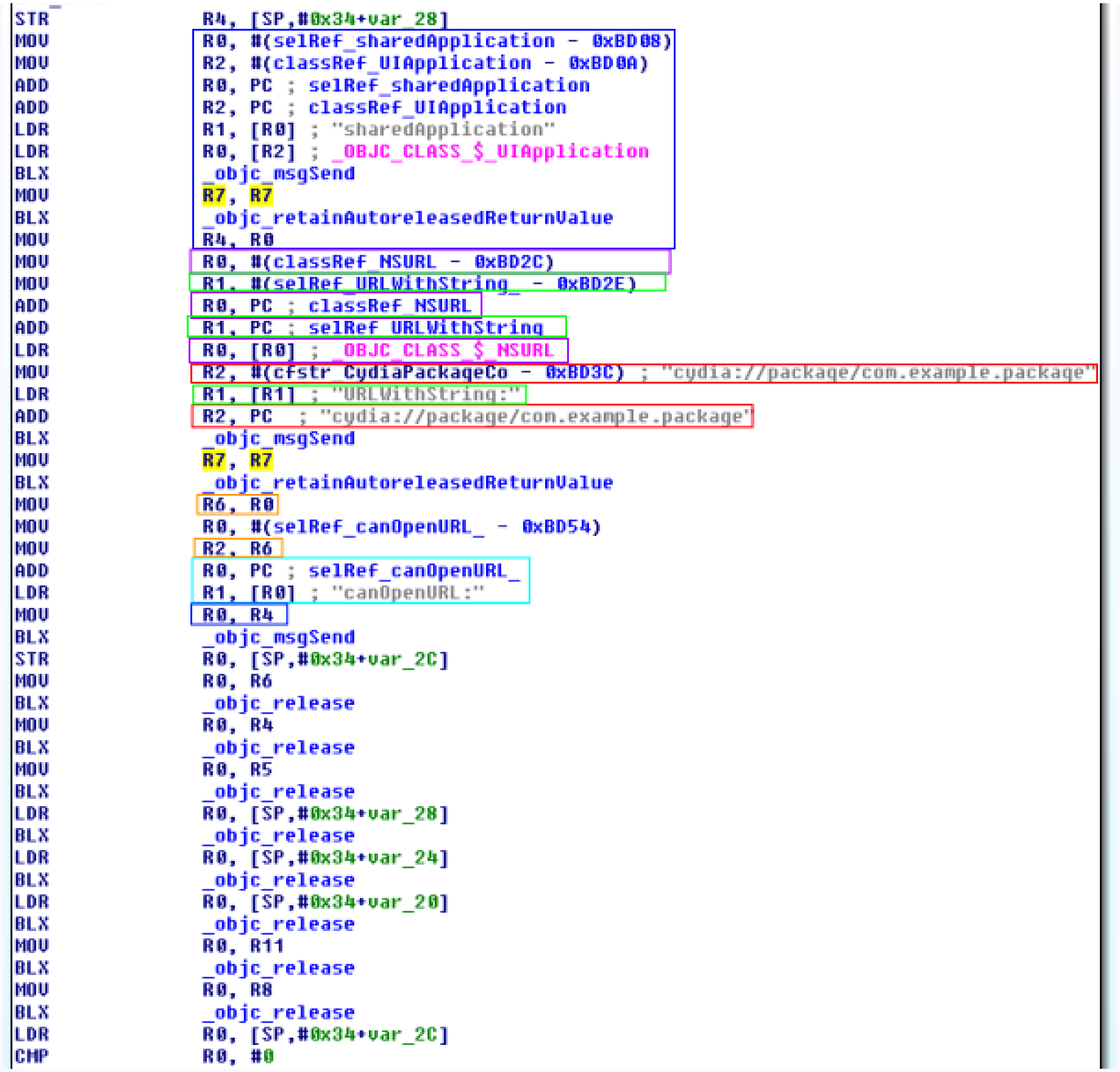
```
loc_BCB8
LDR             R0, [R5] ; _OBJC_CLASS_$_NSFileManager
MOV             R1, R10
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOVW            R2, #(:lower16:(cfstr_PrivateVarLibA - 0xBCD6)) ; "/private/var/lib/apt/"
MOV             R1, R6
MOVT.W          R2, #(:upper16:(cfstr_PrivateVarLibA - 0xBCD6)) ; "/private/var/lib/apt/"
MOV             R5, R0
ADD             R2, PC  ; "/private/var/lib/apt/"
BLX             _objc_msgSend
LDR             R6, [SP,#0x34+var_20]
CBZ             R0, loc_BCF2
```

It checks if the path “/private/var/lib/apt/” exists. It uses the same procedure as before.

It returns a boolean (R0) which is compared with 0x00 (CBZ = Compare and Branch on Zero). If the path does not exists, it jumps to check 7, otherwise the method will return that the device is jailbroken.

Check 7

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



```
STR                R4, [SP,#0x34+var_28]
MOV                R0, #(selRef_sharedApplication - 0xBD08)
MOV                R2, #(classRef_UIApplication - 0xBD0A)
ADD                R0, PC ; selRef_sharedApplication
ADD                R2, PC ; classRef_UIApplication
LDR                R1, [R0] ; "sharedApplication"
LDR                R0, [R2] ; _OBJC_CLASS_$_UIApplication
BLX                _objc_msgSend
MOV                R7, R7
BLX                _objc_retainAutoreleasedReturnValue
MOV                R4, R0
MOV                R0, #(classRef_NSURL - 0xBD2C)
MOV                R1, #(selRef_URLWithString_ - 0xBD2E)
ADD                R0, PC ; classRef_NSURL
ADD                R1, PC ; selRef_URLWithString
LDR                R0, [R0] ; _OBJC_CLASS_$_NSURL
MOV                R2, #(cfstr_CydiaPackageCo - 0xBD3C) ; "cydia://package/com.example.package"
LDR                R1, [R1] ; "URLWithString:"
ADD                R2, PC  ; "cydia://package/com.example.package"
BLX                _objc_msgSend
MOV                R7, R7
BLX                _objc_retainAutoreleasedReturnValue
MOV                R6, R0
MOV                R0, #(selRef_canOpenURL_ - 0xBD54)
MOV                R2, R6
ADD                R0, PC ; selRef_canOpenURL_
LDR                R1, [R0] ; "canOpenURL:"
MOV                R0, R4
BLX                _objc_msgSend
STR                R0, [SP,#0x34+var_2C]
MOV                R0, R6
BLX                _objc_release
MOV                R0, R4
BLX                _objc_release
MOV                R0, R5
BLX                _objc_release
LDR                R0, [SP,#0x34+var_28]
BLX                _objc_release
LDR                R0, [SP,#0x34+var_24]
BLX                _objc_release
LDR                R0, [SP,#0x34+var_20]
BLX                _objc_release
MOV                R0, R11
BLX                _objc_release
MOV                R0, R8
BLX                _objc_release
LDR                R0, [SP,#0x34+var_2C]
CMP                R0, #0
```

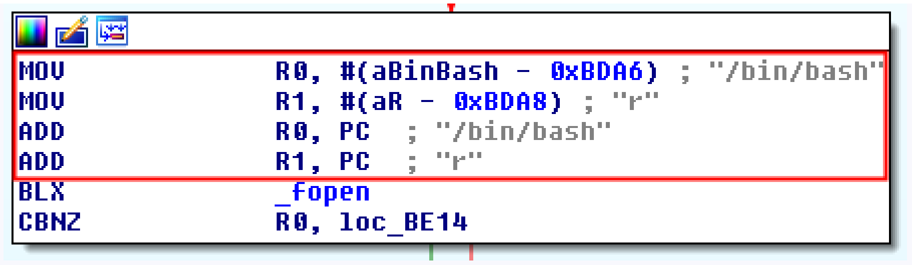
This check tries to call the Cydia’s URL scheme (cydia://). It does it in two steps: first it creates the URL object and then it opens the URL.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

1. First of all it obtains the object of the class UIApplication which is stored in R4 (blue).
2. It obtains the object of the class NSURL which is stored in R0 (purple).
3. Then it obtains the address of the method URLWithString which it is stored in R1 (green).
4. “cydia://package/com.example.package” is stored in R2 (red).
5. It calls _objc_msgSend with the attributes NSURL, URLWithString and “cydia://package/com.example.package” stored in the registers R0, R1 and R2 respectively.
6. It returns a NSURL object initialized with URLString which is stored in first in R6 and after in R2 (orange).
7. It obtains the address of the method canOpenURL which it is stored in R1 (light blue).
8. The value of R4 (UIApplication class) is moved to R0 (blue).
9. It calls _objc_msgSend with the attributes UIApplication, canOpenURL and NSURL object stored in the registers R0, R1 and R2 respectively.
10. It returns a a Boolean value indicating whether an app is available to handle a URL scheme (R0) which is compared with 0x00 (BNE = branch if not equal). If the path exists, it jumps to check 8, otherwise the method will return that the device is jailbroken.

Check 8

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

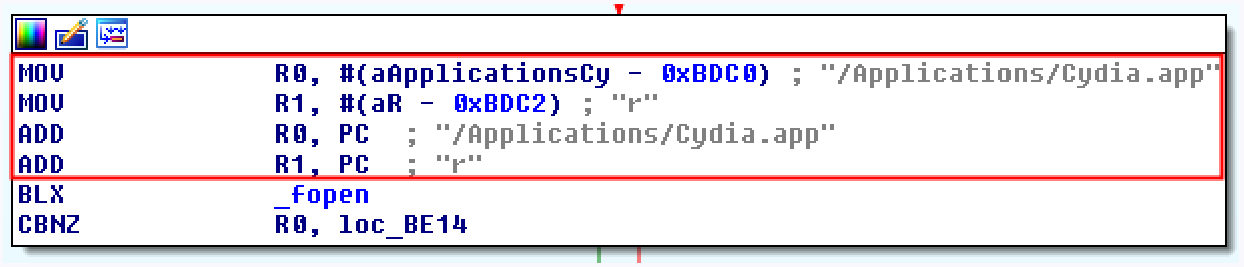


```
MOV             R0, #(aBinBash - 0xBDA6) ; "/bin/bash"
MOV             R1, #(aR - 0xBDA8) ; "r"
ADD             R0, PC  ; "/bin/bash"
ADD             R1, PC  ; "r"
BLX             _fopen
CBNZ            R0, loc_BE14
```

It tries to open the file "/bin/bash" with readonly calling the method "_fopen".

It returns a boolean stored in (R0) which is compared with 0x00 (CBNZ = Compare and Branch on Non-Zero). If the file can not be open, it jumps to check 9, otherwise the method will return that the device is jailbroken.

Check 9



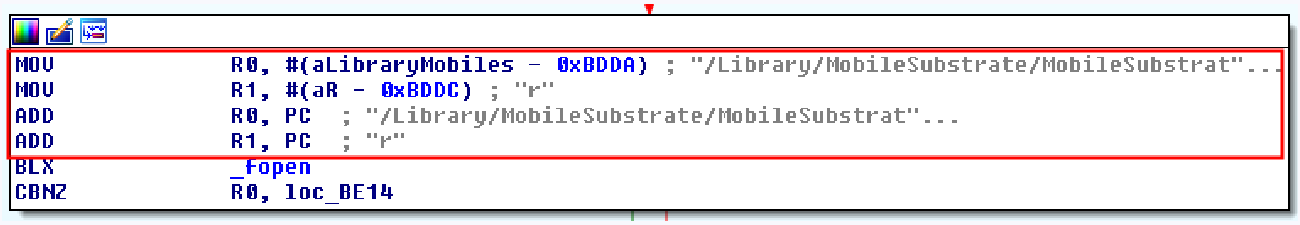
```
MOV             R0, #(aApplicationsCy - 0xBDC0) ; "/Applications/Cydia.app"
MOV             R1, #(aR - 0xBDC2) ; "r"
ADD             R0, PC  ; "/Applications/Cydia.app"
ADD             R1, PC  ; "r"
BLX             _fopen
CBNZ            R0, loc_BE14
```

It tries to open the file "/Application/Cydia.app" with readonly calling the method "_fopen".

It returns a boolean stored in (R0) which is compared with 0x00 (CBNZ = Compare and Branch on Non-Zero). If the file can not be open, it jumps to check 10, otherwise the method will return that the device is jailbroken.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Check 10

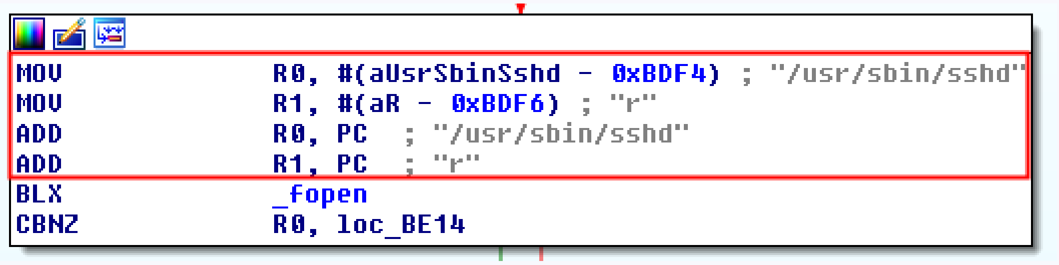


```
MOV             R0, #(aLibraryMobiles - 0xBDDA) ; "/Library/MobileSubstrate/MobileSubstrat"...
MOV             R1, #(aR - 0xBDDC) ; "r"
ADD             R0, PC  ; "/Library/MobileSubstrate/MobileSubstrat"...
ADD             R1, PC  ; "r"
BLX             _fopen
CBNZ            R0, loc_BE14
```

It tries to open the file “/Library/MobileSubstrate/MobileSubstrate.dylib” with readonly calling the method “_fopen”.

It returns a boolean stored in (R0) which is compared with 0x00 (CBNZ = Compare and Branch on Non-Zero). If the file can not be open, it jumps to check 11, otherwise the method will return that the device is jailbroken.

Check 11



```
MOV             R0, #(aUsrSbinSshd - 0xBDF4) ; "/usr/sbin/sshd"
MOV             R1, #(aR - 0xBDF6) ; "r"
ADD             R0, PC  ; "/usr/sbin/sshd"
ADD             R1, PC  ; "r"
BLX             _fopen
CBNZ            R0, loc_BE14
```

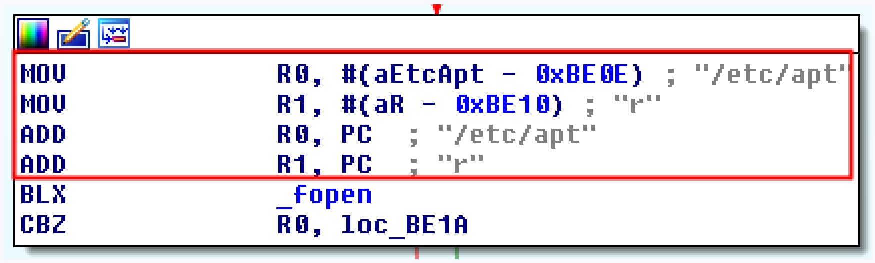
It tries to open the file “/usr/sbin/sshd” with readonly calling the method “_fopen”.

It returns a boolean stored in (R0) which is compared with 0x00 (CBNZ = Compare and Branch on Non-Zero). If the file can not be open, it jumps to check 12, otherwise the

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

method will return that the device is jailbroken.

Check 12



```
MOV             R0, #(aEtcApt - 0xBE0E) ; "/etc/apt"
MOV             R1, #(aR - 0xBE10) ; "r"
ADD             R0, PC  ; "/etc/apt"
ADD             R1, PC  ; "r"
BLX             _fopen
CBZ             R0, loc_BE1A
```

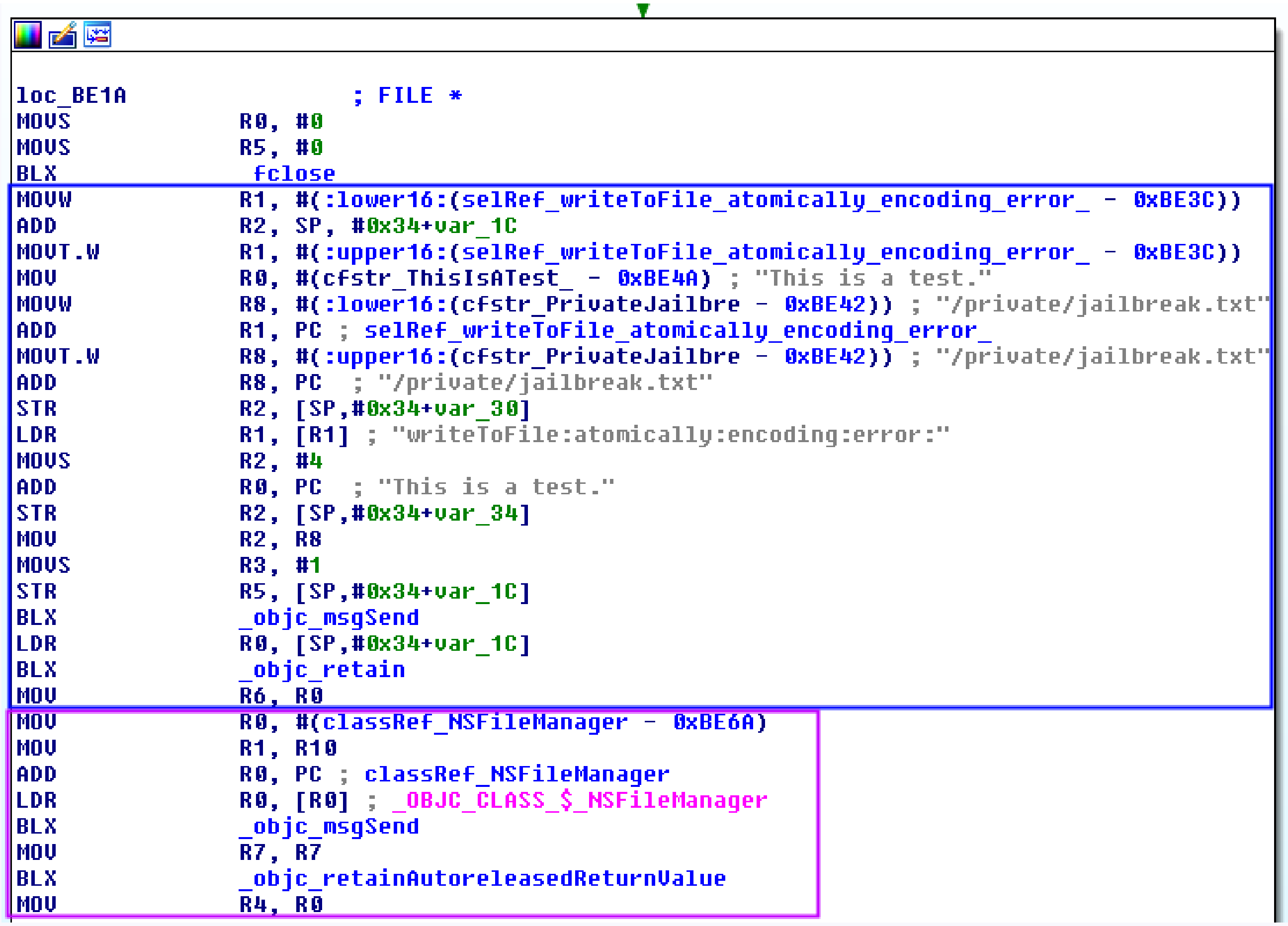
It tries to open the file “/etc/apt” with readonly calling the method “_fopen”.

It returns a boolean stored in (R0) which is compared with 0x00 (CBNZ = Compare and Branch on Non-Zero). If the file can not be open, it jumps to check 13, otherwise the method will return that the device is jailbroken.

Check 13

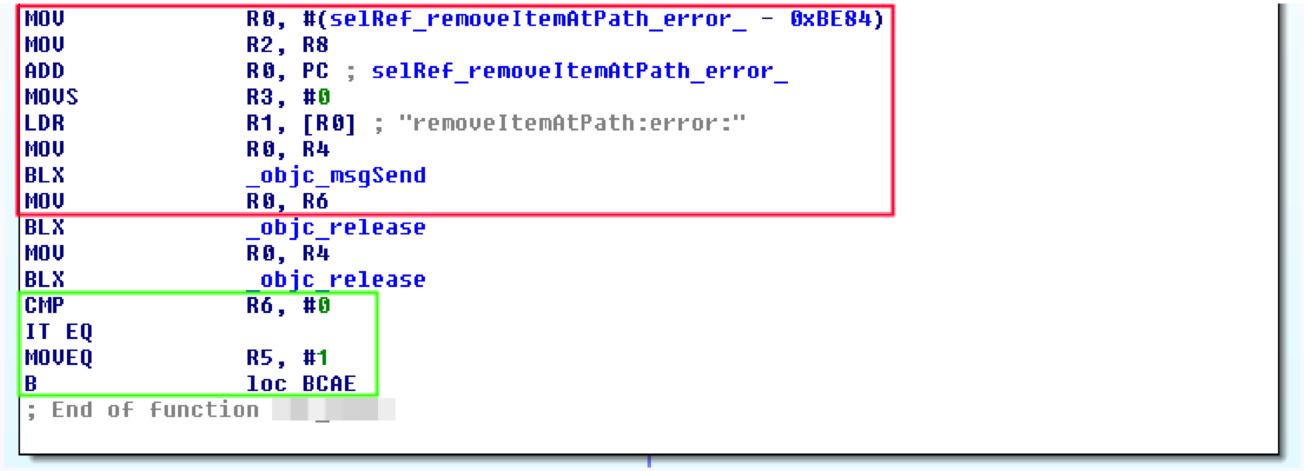
This check tries to create a file called “jailbreak.txt” in /private.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



```
loc_BE1A                        ; FILE *
MOVS            R0, #0
MOVS            R5, #0
BLX              fclose
MOVW            R1, #(:lower16:(selRef_writeToFile_atomically_encoding_error_ - 0xBE3C))
ADD             R2, SP, #0x34+var_1C
MOVT.W          R1, #(:upper16:(selRef_writeToFile_atomically_encoding_error_ - 0xBE3C))
MOV             R0, #(cfstr_ThisIsATest_ - 0xBE4A) ; "This is a test."
MOVW            R8, #(:lower16:(cfstr_PrivateJailbre - 0xBE42)) ; "/private/jailbreak.txt"
ADD             R1, PC ; selRef_writeToFile_atomically_encoding_error_
MOVT.W          R8, #(:upper16:(cfstr_PrivateJailbre - 0xBE42)) ; "/private/jailbreak.txt"
ADD             R8, PC  ; "/private/jailbreak.txt"
STR             R2, [SP,#0x34+var_30]
LDR             R1, [R1] ; "writeToFile:atomically:encoding:error:"
MOVS            R2, #4
ADD             R0, PC  ; "This is a test."
STR             R2, [SP,#0x34+var_34]
MOV             R2, R8
MOVS            R3, #1
STR             R5, [SP,#0x34+var_1C]
BLX             _objc_msgSend
LDR             R0, [SP,#0x34+var_1C]
BLX             _objc_retain
MOV             R6, R0
MOV             R0, #(classRef_NSFileManager - 0xBE6A)
MOV             R1, R10
ADD             R0, PC ; classRef_NSFileManager
LDR             R0, [R0] ; _OBJC_CLASS_$_NSFileManager
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOV             R4, R0
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



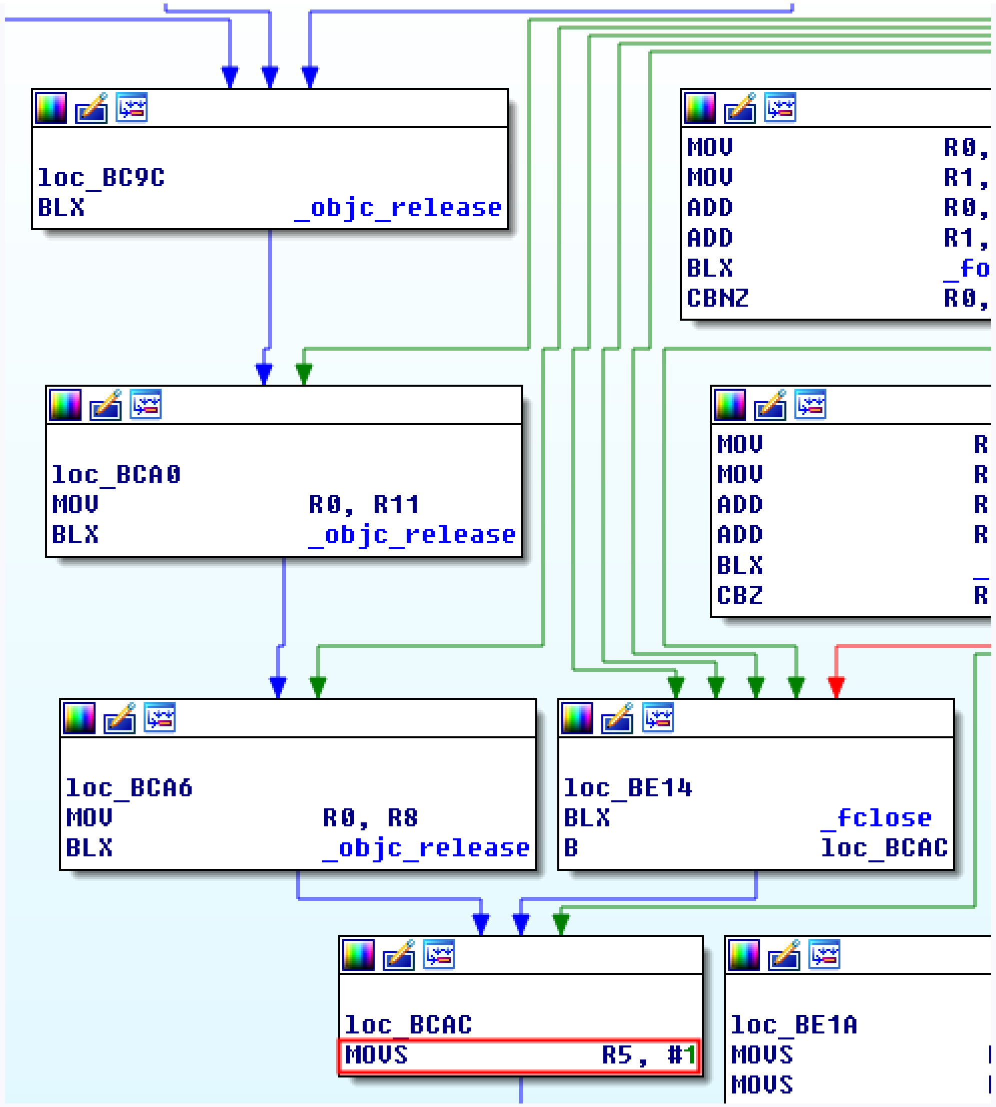
```
MOV             R0, #(selRef_removeItemAtPath_error_ - 0xBE84)
MOV             R2, R8
ADD             R0, PC ; selRef_removeItemAtPath_error_
MOVS            R3, #0
LDR             R1, [R0] ; "removeItemAtPath:error:"
MOV             R0, R4
BLX             _objc_msgSend
MOV             R0, R6
BLX             _objc_release
MOV             R0, R4
BLX             _objc_release
CMP             R6, #0
IT EQ
MOVEQ           R5, #1
B               loc_BCAE
; End of function [redacted]
```

It can be divided in three parts:

1. **Create file**: uses the method writeToFile, stored in R1, to create the file. In R0 there is the text inside the file and in R2 the path "/private/jailbreak.txt". If the file has been successful created, it returns a 1 and it is stored in R6. (blue)
2. **Remove file**: First obtains the object of NSFileManager by calling _objc_msgSend with NSFileManager in R0 and defaultManager in R1, moved from R10 (purple). The obtained object is used to call the method removeItemAtPath, stored in R1, with the path passed as argument stored in R2. (red)
3. **Comparision**: Moves the result of the file creation stored in R6 to R0. It makes a comparission between R0 and 0 and stores 1 to R5 if R0 is 0. (green)

Is Jailbroken

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



```
loc_BC9C
BLX                 _objc_release
```

```
MOV                 R0,
MOV                 R1,
ADD                 R0,
ADD                 R1,
BLX                 _fo
CBNZ                R0,
```

```
loc_BCA0
MOV                 R0, R11
BLX                 _objc_release
```

```
MOV                 R
MOV                 R
ADD                 R
ADD                 R
BLX                 _
CBZ                 R
```

```
loc_BCA6
MOV                 R0, R8
BLX                 _objc_release
```

```
loc_BE14
BLX                 _fclose
B                   loc_BCAC
```

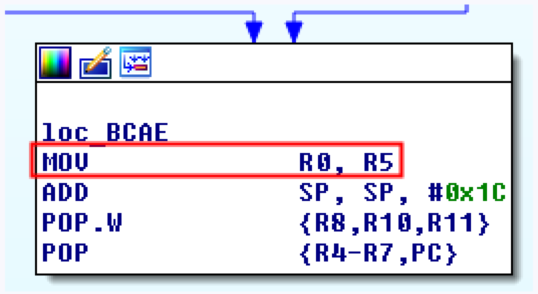
```
loc_BCAC
MOVS                R5, #1
```

```
loc_BE1A
MOVS
MOVS
```

If any of the checks fail, except check 13 which jumps to the return method, it ends here. Integer 1 is moved to R5 (device is jailbroken).

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Return



```
loc_BCAE
MOV                 R0, R5
ADD                 SP, SP, #0x1C
POP.W               {R8,R10,R11}
POP                 {R4-R7,PC}
```

There are two possible ways to get here:

1. From the “Is Jailbroken” step, so R5 will be 1.
2. From the “Check 13”, so R5 can be either 0 or 1.

The method will move R5 to R0 and will return 1 if the device is jailbroken or 0 if not.

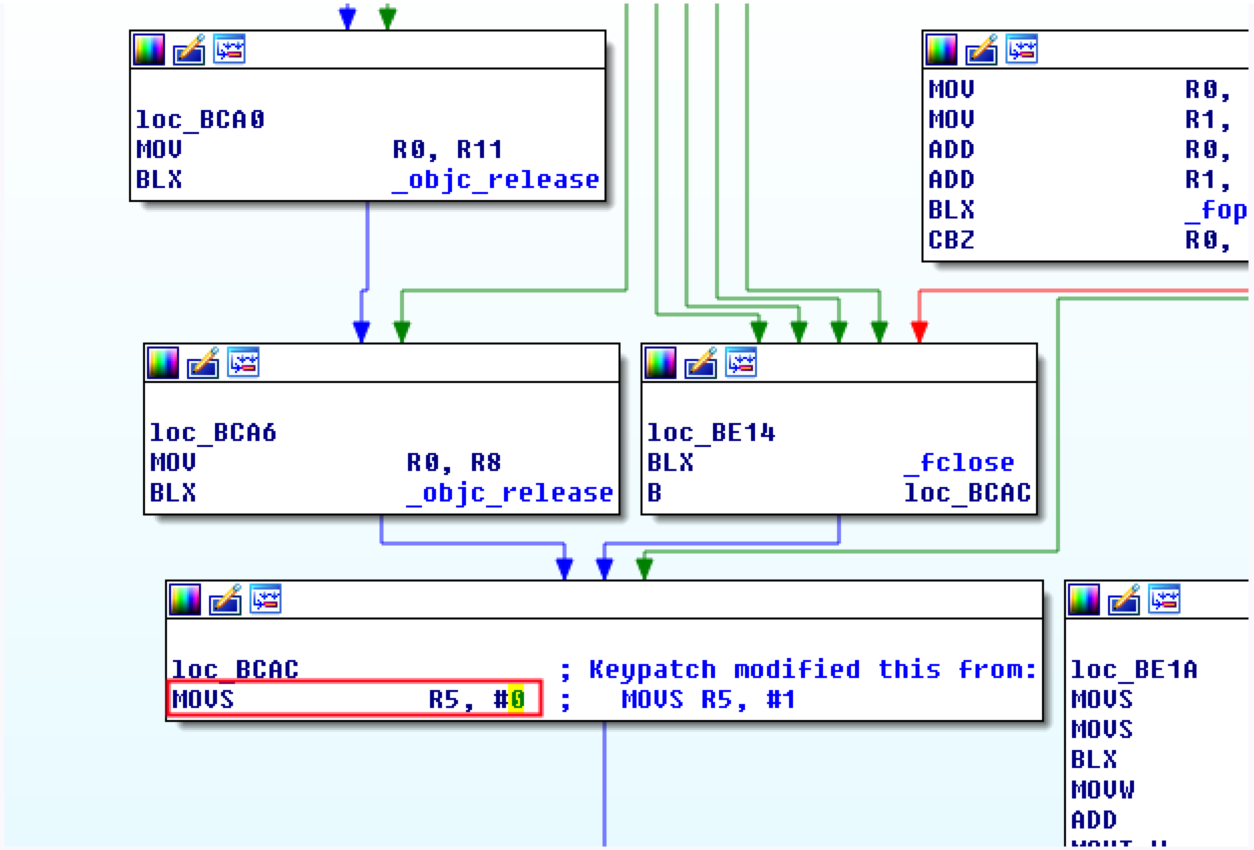
Patching

Jailbreak method

Makeing the method which checks if the device is jailbroken return always 0. There are some ways to do it:

1. Modifying all jumps from all the checks so they jump always to the next check and then make check 13 to return always 0 by modifying the comparision.
2. Modifying is_jailbroken because is where the checks jump if they fail.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD


```
loc_BCA0
MOV                R0, R11
BLX                _objc_release
```
```
MOV                R0,
MOV                R1,
ADD                R0,
ADD                R1,
BLX                _fop
CBZ                R0,
```
```
loc_BCA6
MOV                R0, R8
BLX                _objc_release
```
```
loc_BE14
BLX                _fclose
B                  loc_BCAC
```
```
loc_BCAC                  ; Keypatch modified this from:
MOVS               R5, #0 ;   MOVS R5, #1
```
```
loc_BE1A
MOVS
MOVS
BLX
MOVW
ADD
```

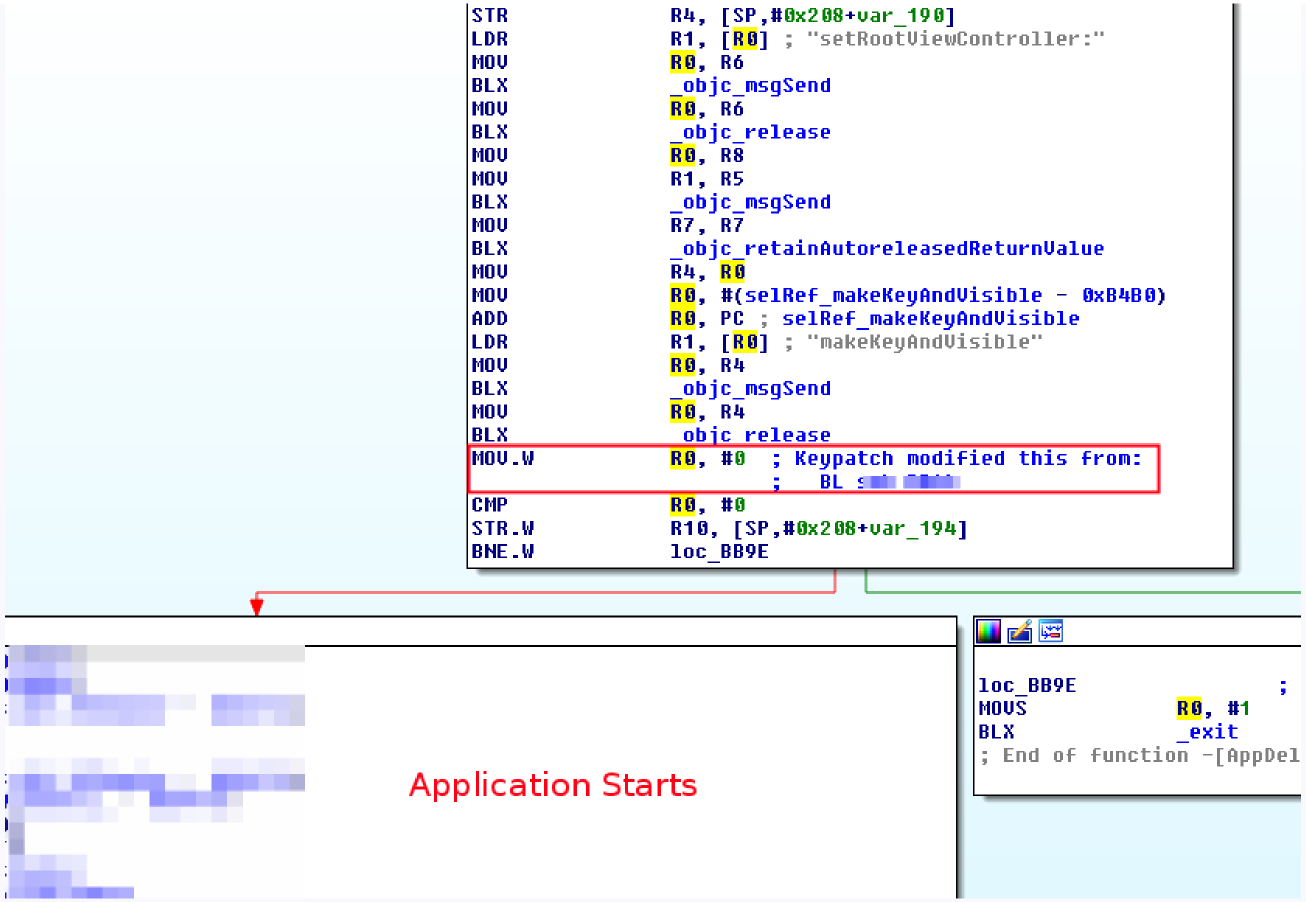
1. Modify R0 before the jailbreak method ends, so it always return 0.

Parent method

An other way is to modify the behaviour of the method which calls the jailbreak check:

1. Modifying the register used in the comparission by repleacing the call to the jailbreak check method:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



```
STR             R4, [SP,#0x208+var_190]
LDR             R1, [R0] ; "setRootViewController:"
MOV             R0, R6
BLX             _objc_msgSend
MOV             R0, R6
BLX             _objc_release
MOV             R0, R8
MOV             R1, R5
BLX             _objc_msgSend
MOV             R7, R7
BLX             _objc_retainAutoreleasedReturnValue
MOV             R4, R0
MOV             R0, #(selRef_makeKeyAndVisible - 0xB4B0)
ADD             R0, PC  ; selRef_makeKeyAndVisible
LDR             R1, [R0] ; "makeKeyAndVisible"
MOV             R0, R4
BLX             _objc_msgSend
MOV             R0, R4
BLX             _objc_release
MOV.W           R0, #0  ; Keypatch modified this from:
                        ;   BL s[redacted]
CMP             R0, #0
STR.W           R10, [SP,#0x208+var_194]
BNE.W           loc_BB9E
```

[redacted]

```
loc_BB9E                        ;
MOVS            R0, #1
BLX             _exit
; End of function -[AppDel
```

1. Modifying the comparission in a way that starts the application however the result of the jailbreak check is negative.

Similar Posts

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

- Frida iOS
- Information gathering
- iOS Pentesting - Introduction
- iOS Pentesting - Static analysis
- iOS Pentesting Tools

Previous post Basic Buffer Overflow

Next post Certificate Pinning and Mutual Authentication

Contact me at:

This site has been visited: times, Number of visitors: , This post has been viewed times

Site powered by Jekyll & Github Pages. Theme designed by HyG.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD