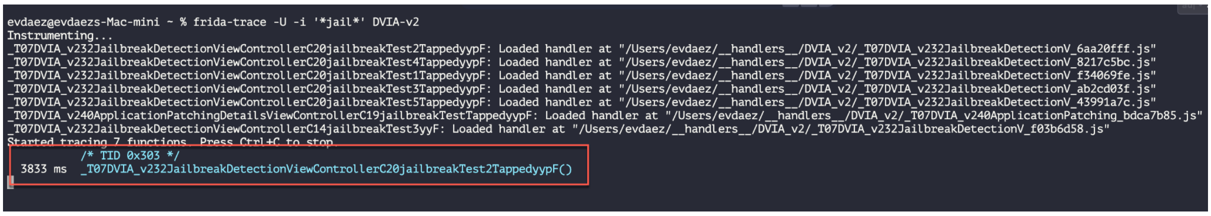
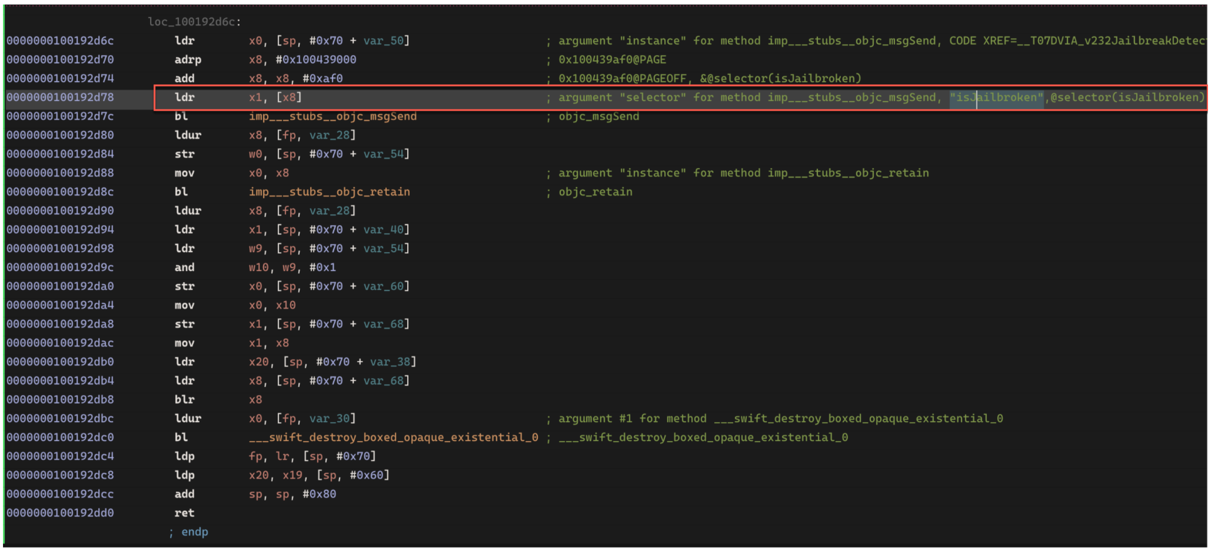
When ‘Jailbreak Test 2’ is tapped, the following swift function gets triggered:



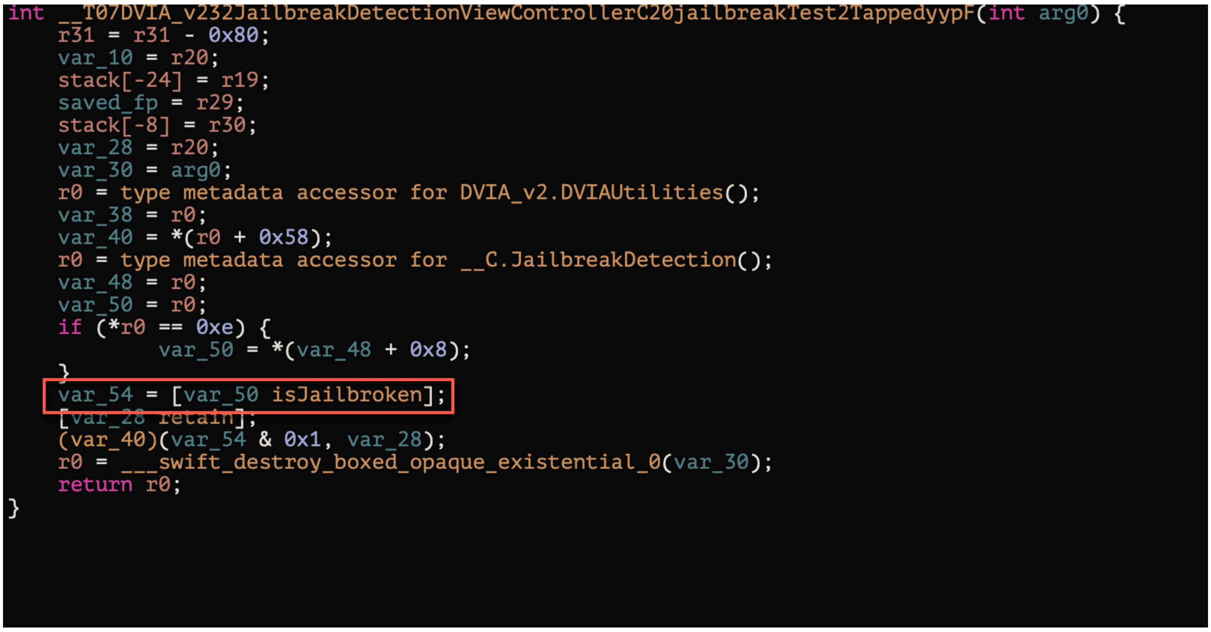
```
evdaez@evdaezs-Mac-mini ~ % frida-trace -U -i '*jail*' DVIA-v2
Instrumenting...
_T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest2TappedyypF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v232JailbreakDetectionV_6aa20fff.js"
_T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest4TappedyypF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v232JailbreakDetectionV_8217c5bc.js"
_T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest1TappedyypF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v232JailbreakDetectionV_f34069fe.js"
_T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest3TappedyypF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v232JailbreakDetectionV_ab2cd03f.js"
_T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest5TappedyypF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v232JailbreakDetectionV_43991a7c.js"
_T07DVIA_v240ApplicationPatchingDetailsViewControllerC19jailbreakTestTappedyypF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v240ApplicationPatching_bdca7b85.js"
_T07DVIA_v232JailbreakDetectionViewControllerC14jailbreakTest3yyF: Loaded handler at "/Users/evdaez/__handlers__/DVIA_v2/_T07DVIA_v232JailbreakDetectionV_f03b6d58.js"
Started tracing 7 functions. Press Ctrl+C to stop.
           /* TID 0x303 */
  3833 ms  _T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest2TappedyypF()
```

When you disassemble and trace the code, on the comments section, there is something saying isJailbroken



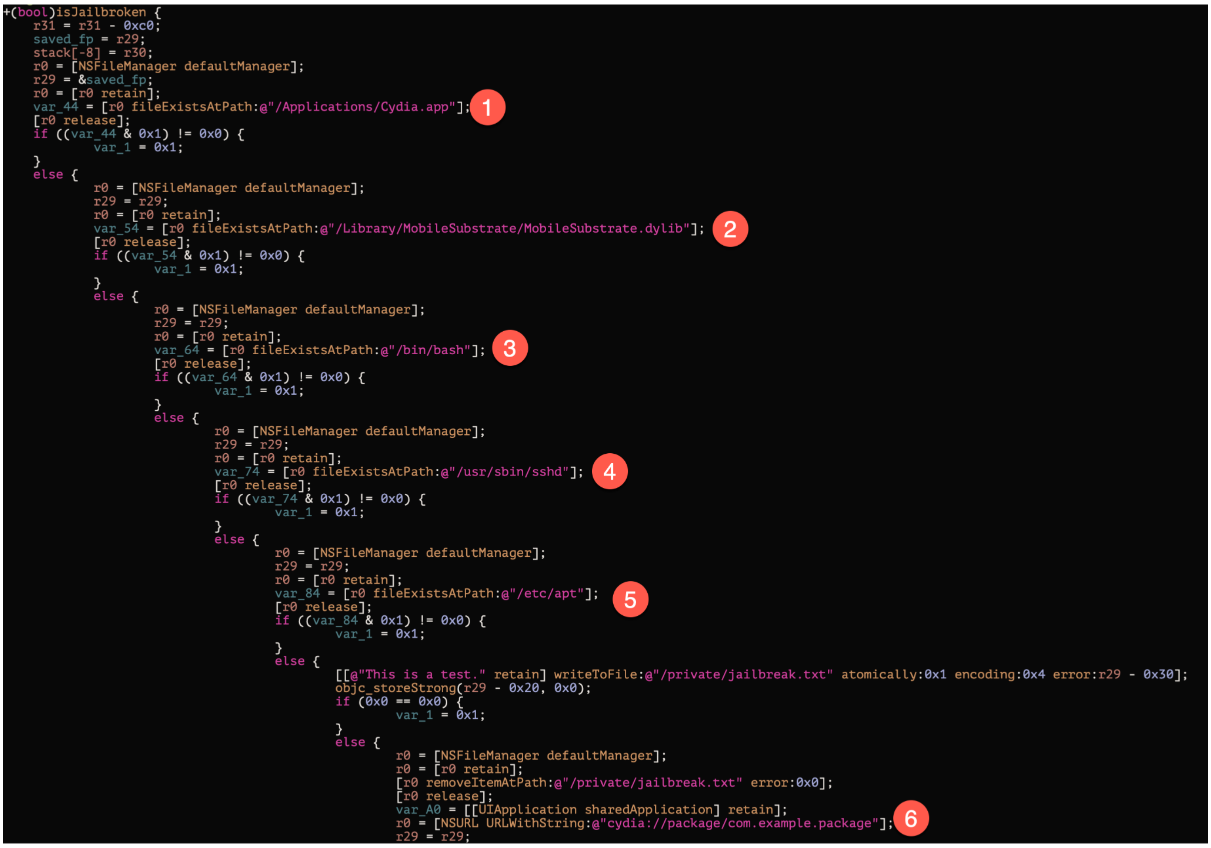
```
                        loc_100192d6c:
0000000100192d6c        ldr        x0, [sp, #0x70 + var_50]                    ; argument "instance" for method imp___stubs__objc_msgSend, CODE XREF=__T07DVIA_v232JailbreakDetec
0000000100192d70        adrp       x8, #0x100439000                            ; 0x100439af0@PAGE
0000000100192d74        add        x8, x8, #0xaf0                              ; 0x100439af0@PAGEOFF, &@selector(isJailbroken)
0000000100192d78        ldr        x1, [x8]                                    ; argument "selector" for method imp___stubs__objc_msgSend, "isJailbroken",@selector(isJailbroken)
0000000100192d7c        bl         imp___stubs__objc_msgSend                   ; objc_msgSend
0000000100192d80        ldur       x8, [fp, var_28]
0000000100192d84        str        w0, [sp, #0x70 + var_54]
0000000100192d88        mov        x0, x8                                      ; argument "instance" for method imp___stubs__objc_retain
0000000100192d8c        bl         imp___stubs__objc_retain                    ; objc_retain
0000000100192d90        ldur       x8, [fp, var_28]
0000000100192d94        ldr        x1, [sp, #0x70 + var_40]
0000000100192d98        ldr        w9, [sp, #0x70 + var_54]
0000000100192d9c        and        w10, w9, #0x1
0000000100192da0        str        x0, [sp, #0x70 + var_60]
0000000100192da4        mov        x0, x10
0000000100192da8        str        x1, [sp, #0x70 + var_68]
0000000100192dac        mov        x1, x8
0000000100192db0        ldr        x20, [sp, #0x70 + var_38]
0000000100192db4        ldr        x8, [sp, #0x70 + var_68]
0000000100192db8        blr        x8
0000000100192dbc        ldur       x0, [fp, var_30]                            ; argument #1 for method ___swift_destroy_boxed_opaque_existential_0
0000000100192dc0        bl         ___swift_destroy_boxed_opaque_existential_0 ; ___swift_destroy_boxed_opaque_existential_0
0000000100192dc4        ldp        fp, lr, [sp, #0x70]
0000000100192dc8        ldp        x20, x19, [sp, #0x60]
0000000100192dcc        add        sp, sp, #0x80
0000000100192dd0        ret
                         ; endp
```

On the decompiled code you can access the isJailbroken function which will then check for files which exist on certain paths or if it can open any cydia url:



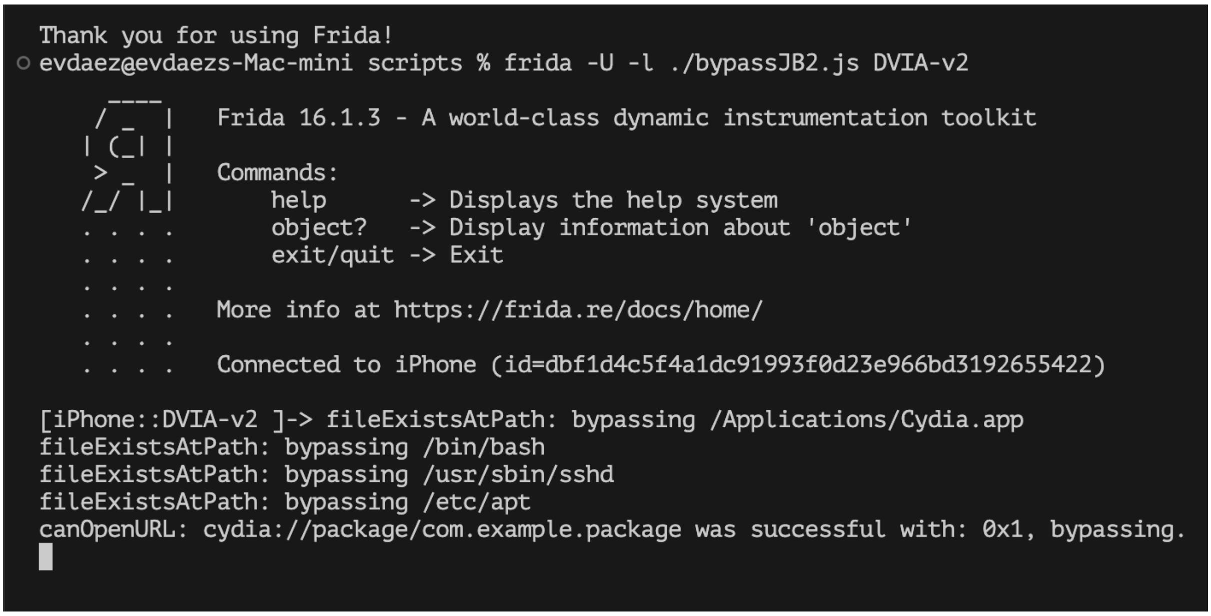
```
int __T07DVIA_v232JailbreakDetectionViewControllerC20jailbreakTest2TappedyypF(int arg0) {
    r31 = r31 - 0x80;
    var_10 = r20;
    stack[-24] = r19;
    saved_fp = r29;
    stack[-8] = r30;
    var_28 = r20;
    var_30 = arg0;
    r0 = type metadata accessor for DVIA_v2.DVIAUtilities();
    var_38 = r0;
    var_40 = *(r0 + 0x58);
    r0 = type metadata accessor for __C.JailbreakDetection();
    var_48 = r0;
    var_50 = r0;
    if (*r0 == 0xe) {
            var_50 = *(var_48 + 0x8);
    }
    var_54 = [var_50 isJailbroken];
    [var_28 retain];
    (var_40)(var_54 & 0x1, var_28);
    r0 = ___swift_destroy_boxed_opaque_existential_0(var_30);
    return r0;
}
```

So in total there are 6 things that you need to bypass:



```
+(bool)isJailbroken {
    r31 = r31 - 0xc0;
    saved_fp = r29;
    stack[-8] = r30;
    r0 = [NSFileManager defaultManager];
    r29 = &saved_fp;
    r0 = [r0 retain];
    var_44 = [r0 fileExistsAtPath:@"/Applications/Cydia.app"];
    [r0 release];
    if ((var_44 & 0x1) != 0x0) {
            var_1 = 0x1;
    }
    else {
            r0 = [NSFileManager defaultManager];
            r29 = r29;
            r0 = [r0 retain];
            var_54 = [r0 fileExistsAtPath:@"/Library/MobileSubstrate/MobileSubstrate.dylib"];
            [r0 release];
            if ((var_54 & 0x1) != 0x0) {
                    var_1 = 0x1;
            }
            else {
                    r0 = [NSFileManager defaultManager];
                    r29 = r29;
                    r0 = [r0 retain];
                    var_64 = [r0 fileExistsAtPath:@"/bin/bash"];
                    [r0 release];
                    if ((var_64 & 0x1) != 0x0) {
                            var_1 = 0x1;
                    }
                    else {
                            r0 = [NSFileManager defaultManager];
                            r29 = r29;
                            r0 = [r0 retain];
                            var_74 = [r0 fileExistsAtPath:@"/usr/sbin/sshd"];
                            [r0 release];
                            if ((var_74 & 0x1) != 0x0) {
                                    var_1 = 0x1;
                            }
                            else {
                                    r0 = [NSFileManager defaultManager];
                                    r29 = r29;
                                    r0 = [r0 retain];
                                    var_84 = [r0 fileExistsAtPath:@"/etc/apt"];
                                    [r0 release];
                                    if ((var_84 & 0x1) != 0x0) {
                                            var_1 = 0x1;
                                    }
                                    else {
                                            [[@"This is a test." retain] writeToFile:@"/private/jailbreak.txt" atomically:0x1 encoding:0x4 error:r29 - 0x30];
                                            objc_storeStrong(r29 - 0x20, 0x0);
                                            if (0x0 == 0x0) {
                                                    var_1 = 0x1;
                                            }
                                            else {
                                                    r0 = [NSFileManager defaultManager];
                                                    r0 = [r0 retain];
                                                    [r0 removeItemAtPath:@"/private/jailbreak.txt" error:0x0];
                                                    [r0 release];
                                                    var_A0 = [[UIApplication sharedApplication] retain];
                                                    r0 = [NSURL URLWithString:@"cydia://package/com.example.package"];
                                                    r29 = r29;
```

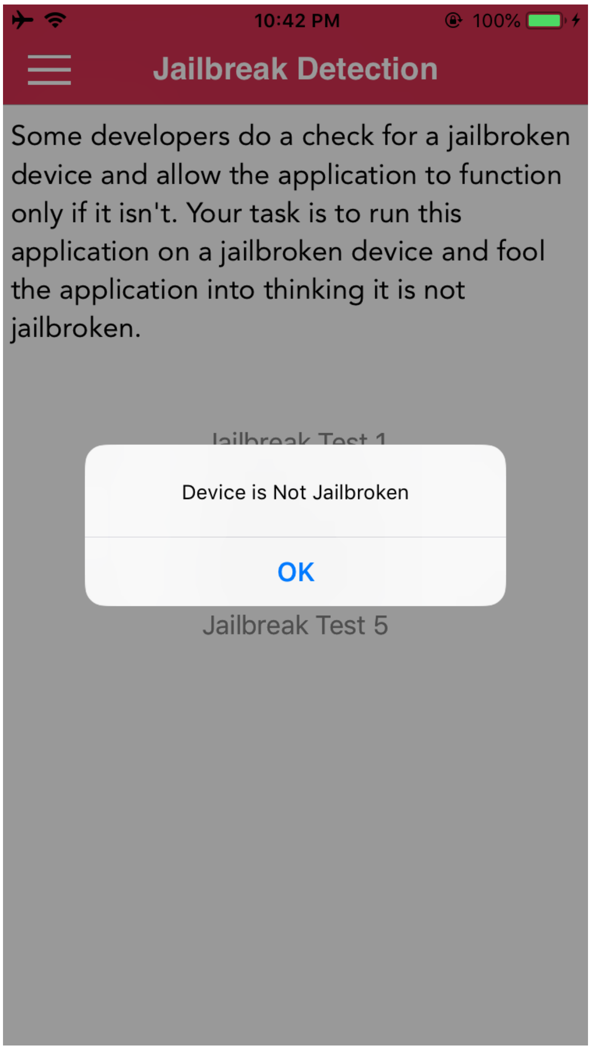
Results:



```
 Thank you for using Frida!
○ evdaez@evdaezs-Mac-mini scripts % frida -U -l ./bypassJB2.js DVIA-v2
     ____
    / _  |   Frida 16.1.3 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to iPhone (id=dbf1d4c5f4a1dc91993f0d23e966bd3192655422)

 [iPhone::DVIA-v2 ]-> fileExistsAtPath: bypassing /Applications/Cydia.app
 fileExistsAtPath: bypassing /bin/bash
 fileExistsAtPath: bypassing /usr/sbin/sshd
 fileExistsAtPath: bypassing /etc/apt
 canOpenURL: cydia://package/com.example.package was successful with: 0x1, bypassing.
```

Modified code from
https://codeshare.frida.re/@DevTraleski/ios-jailbreak-detection-bypass-palera1n/



Jailbreak Detection

Some developers do a check for a jailbroken device and allow the application to function only if it isn't. Your task is to run this application on a jailbroken device and fool the application into thinking it is not jailbroken.

Jailbreak Test 1

Device is Not Jailbroken

Jailbreak Test 5