A JOURNEY INTO CREATING A MOBILE APP HACKING TOOLKIT

@leonjza | NowSecure Connect '19

$ WHOAMI

- Security Researcher @ SensePost
- Write code, Hack many things, including mobile applications
- Unhealthy addiction to porting things to Frida

PENTESTING IN A PRE-FRIDA WORLD...

CONSIDER THE TYPICAL THREAT MODEL

TYPICAL APPROACH

- Where is that jailbroken device again?
- A race to get to API endpoints to test the "fun stuff"
- Use existing tooling (or manual RE if needed) to bypass SSL pinning and dump keychains
- Depending on time, check out some OWASP guidelines and whatever you can get to
- Rinse and repeat

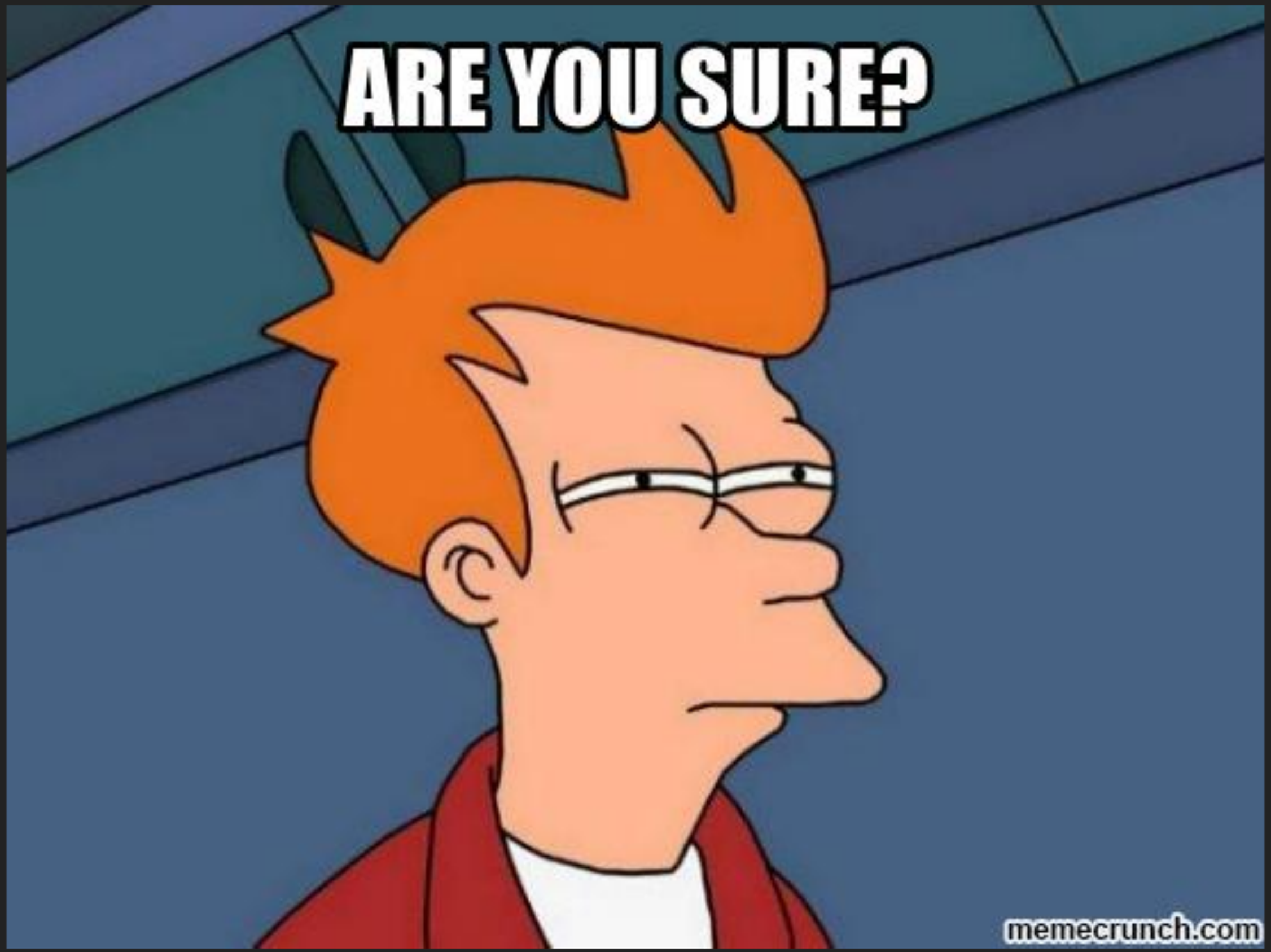
:(

EMV CARD READER PROXY

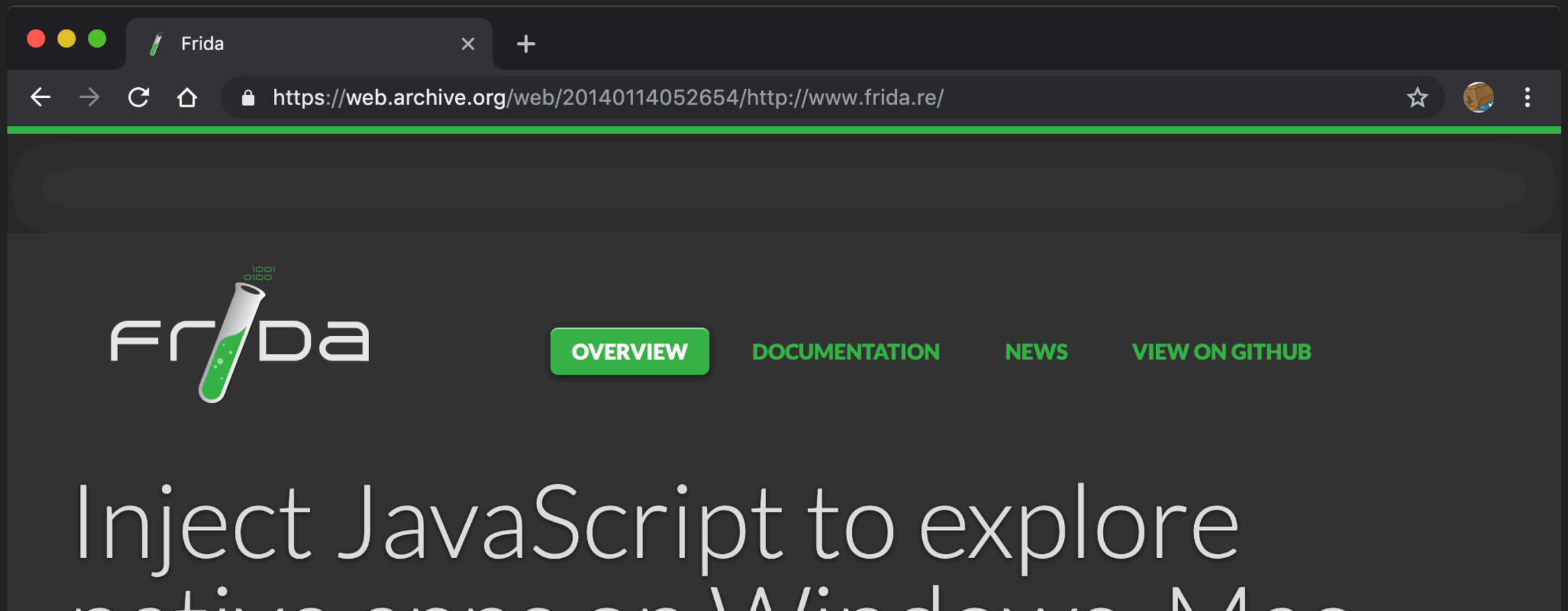

Present card
Please insert chip card or swipe
MAG card now

EMV CARD READER

- "Everything is encrypted"
- **Android app was apparently just a "proxy" between the hardware and a web server...**


ARE YOU SURE?
memecrunch.com

```
/* Access modifiers changed, original: 0000 */
public void decode(byte[] samples, int length) {
    for (int i = 0; i < length; i++) {
        byte currentInput = samples[i];
        switch (this.state) {
            case PREAMBLE_START:
                if (this.prevInput == (byte) 1 && currentInput == (byte) 0) {
                    this.baseTime = 0.0d;
                    this.currentTime = 0.0d;
                    [redacted]
                    break;
                }
            case PREAMBLE_DECODE:
                this.currentTime += [redacted];
                if (this.prevInput != (byte) 1 || currentInput != (byte) 0) {
                    if (this.currentTime - this.baseTime <= this.IDLE_PERIOD) {
                        break;
                    }
                    this.state = State.PREAMBLE_START;
                    break;
                } else if (this.currentTime - [redacted]) - this.TOLERANCE ||
                    if (this.currentTime - this.baseTime > this.LOGIC_LOW_PERIOD - this.TOLERANCE && this.
                        if (this.preambleBitsReceived <= 3) {
                            [redacted]
                            break;
                        }
```


Frida
https://web.archive.org/web/20140114052654/http://www.frida.re/
OVERVIEW
DOCUMENTATION
NEWS
VIEW ON GITHUB
Inject JavaScript to explore native apps on Windows, Mac, Linux and iOS.
Scriptable
Your own scripts get injected into black box processes to execute custom debugging logic. Hook any function, spy on crypto APIs or trace private application code, no source code needed!
Stalking
Stealthy code tracing without relying on software or hardware breakpoints. Think DTrace in user-space, based on dynamic recompilation, like DynamoRIO and PIN.
Portable
Works on Windows, Mac, Linux, and iOS. Grab a Python package from PyPI or use Frida through its .NET binding, browser plugin or C API.

COMMAND CODE FLOW

- TXT Command Invocation
  - Audio Frames Translation
    - Audio RX/TX

```
3   if (Java.available) {
4
5     Java.perform(function () {
6
7       try {
8         1
9         var DefaultCommandProcessor = Java.use("com.app.command.DefaultCommandProcessor");
10        var ProcessAnswerCommandImpltcr1 = Java.use("com.app.commands.ProcessAnswerCommandImpl");
11        var ProcessAnswerCommandImpltcr2 = Java.use("com.app.command.ProcessAnswerCommandImpl");
12        var Base16 = Java.use("com.app.util.Base16").$new();
13                                  1
14        DefaultCommandProcessor.execute
15        .overload("java.lang.String", "com.app.command.Command", "boolean")
16        .implementation = function (str, command, z) {
17                          2
18          var result = this.execute(str, command, z);
19
20          console.log("[*] Detected call to DefaultCommandProcessor.execute");
21          console.log("\t[command name] " + command.getName());          3
22          console.log("\t[responsebytes] " + Base16.encode(command.getBytes()));
23          console.log("\t[result] " + command.getResult());
24                                      3
25          return result;
26        }
```

SO IT WAS A PROXY WITH "ISSUES", BUT MOSTLY OK

- So it was mostly a proxy, with some "issues", but the important bits were encrypted
- Was pretty easy to instrument but had some repetitive work
- Essentially had a proxy for the proxy to proxy the proxy, in a few lines of code

```
DefaultCommandProcessor.doPrompt.implementation = function (getPromptCommand) {

  console.log("[*] Detected call to DefaultCommandProcessor.doPrompt");
  console.log("\t[prompt name] " + getPromptCommand.getPrompt().getName());
  console.log("\t[prompt code] " + getPromptCommand.getPrompt().getCode());
  console.log("\t[prompt strings] " + getPromptCommand.getStrings());

  this.doPrompt(getPromptCommand);
}
```

```
ProcessAnswerCommandImpltcr1.getPayloadData.implementation = function () {

  console.log("[*] Detected call to ProcessAnswerCommandImpl.getPayloadData (tcr1)");

  var response = this.getPayloadData();
  console.og("\t[payload data] " + response);

  return response;
}

ProcessAnswerCommandImpltcr2.getPayloadData.implementation = function () {

  console.log("[*] Detected call to ProcessAnswerCommandImpl.getPayloadData (tcr2)");

  var response = this.getPayloadData();
  console.log("\t[payload data] " + response);
  console.log("\t[payload prompt] " + JSON.stringify(this.prompt.value));
  console.log("\t[payload prompt type] " + JSON.stringify(this.promptType.value));
  console.log("\t[payload answer] " + JSON.stringify(this.answer.value));

  return response;
}
```

"I NEED TO GET A DATABASE FILE OUT OF AN APP CONTAINER, CAN YOU HELP?"

Colleague without a jailbroken device

Can I Jailbreak? - Home

https://web.archive.org/web/20171004163624/http://www.canijailbreak.com/

Can I Jailbreak?

by IPSW Downloads

Home Get Jailbreaking Help

My iOS device is on **iOS 10.2.1 → 11.0.2**

✖ No jailbreak yet ☹. Check back later!

*If you are already jailbroken, we recommend that you **do not upgrade** or you will lose your jailbreak!*

---

iOS 10.0 → 10.2

✔ Jailbreak using Yalu (beta, unstable) version beta7

Recommended for advanced users only. Semi-untethered. Does not work on iPhone 7. Follow RedmondPie tutorial

iOS 9.3.5

✔ Jailbreak using Phœnix version 3

32-bit devices only. Semi-untethered only. Follow the Cydia Impactor method of installing the Jailbreak.

iOS 9.1 → 9.3.4

✔ Jailbreak using Home Depot version RC3

32-bit devices only. Semi-untethered only. Follow the Cydia Impactor method of installing the Jailbreak.

https://twitter.com/Burp_Suite/status/857989397781852161

HOW FRIDA SAVED THE DAY

GADGET MODE

```
0x0000186c  cmd       51 0x26 LC_FUNCTION_STARTS
0x00001870  cmdsize      16
0x0000187c  cmd       52 0x29 LC_DATA_IN_CODE
0x00001880  cmdsize      16
0x0000188c  cmd       53 0xc LC_LOAD_DYLIB
0x00001890  cmdsize      72
0x000018a0  load_dylib   @executable_path/Frameworks/FridaGadget.dylib
0x000018d4  cmd       54 0x1d LC_CODE_SIGNATURE
0x000018d8  cmdsize      16
0x000018d8  dataoff      0x0219c9e0
0x000018dc  datasize     466544
# wtf mach0.sign 466544 @ 0x219c9e0

[0x1000adbfc]>
```

r2 "iH" header parse

```
# direct methods
.method static constructor <clinit>()V
    .locals 3

    .prologue
    const/4 v1, 0x0

    const/4 v0, 0x1

    .line 48
    sput-boolean v1, L[redacted]/ApplicationLoader;->applicationInited:Z

    .line 50
    sput-boolean v1, L[redacted]/ApplicationLoader;->isScreenOn:Z

    .line 51
    sput-boolean v0, L[redacted]/ApplicationLoader;->mainInterfacePaused:Z

    .line 52
    sput-boolean v0, L[redacted]/ApplicationLoader;->externalInterfacePaused:Z

    .line 53
    sput-boolean v0, L[redacted]/ApplicationLoader;->mainInterfacePausedStageQueue:Z

    const-string v0, "frida-gadget"

    invoke-static {v0}, Ljava/lang/System;->loadLibrary(Ljava/lang/String;)V

    return-void
.end method
```

Java class static constructor

GADGET MODE DOES NOT REQUIRE ROOT*

ABOUT THAT FILE ...

```
1   var target = 'database.sqlite';
2
3   var NSString = ObjC.classes.NSString;
4   var NSData = ObjC.classes.NSData;
5   var NSFileManager = ObjC.classes.NSFileManager;
6
7   // NSPathUtilities.h
8   var NSDocumentDirectory = 9;
9   var NSUserDomainMask = 1;
10
11  var defaultFileManager = NSFileManager.defaultManager();
12  var documentsPaths = defaultFileManager.URLsForDirectory_inDomains_(NSDocumentDirectory, NSUserDomainMask);
13  var documentsPath = documentsPaths.lastObject();
14  var path = NSString.stringWithString_(documentsPath.toString() + target);
15
16  var data = NSData.dataWithContentsOfFile_(path);
17  var bytes = Memory.readByteArray(data.bytes(), data.length());
18
19  send('incoming', bytes);
20
21  // -- Sample Objective-C
22  //
23  // NSData *data = [NSData dataWithContentsOfFile:fileToRead];
24
```

1

2

3

4

SEND IT

DOWNLOAD A FILE, IN A CONTAINER, IN A SANDBOX

```
(venv3) ------------------------------------------------------------
~/scratch » cat download.py
```

File: **download.py**

```
1   import frida
2
3   def _on_message(msg, data):
4
5       if 'type' in msg and msg['type'] == 'error':
6           print('[x] An error occured: {}'.format(msg['description']))
7           print('{}'.format(msg['stack']))
8           return
9
10      if 'payload' in msg and msg['payload'] == 'incoming':
11          print('Saving incoming bytes')
12          with open(destination, 'wb') as f:
13              f.write(data)
14          print('Done writing {} bytes to {}'.format(len(data), destination))
15          return
16
17      print('Unknown payload message:
18
19  destination = 'database.sqlite'
20
21  with open('download.js', 'r') as f:
22      s = ''.join(f.readlines())
23
24  device = frida.get_device_manager().
25  session = device.attach('Gadget')
26  script = session.create_script(s)
27  script.on('message', _on_message)
28  script.load()
29  script.unload()
30
```

1

AND SAVE IT

```
(venv3) ------------------------------------------------------------
~/scratch » python download.py
Saving incoming bytes
Done writing 12288 bytes to database.sqlite
(venv3) ------------------------------------------------------------
~/scratch »
```

2

WAIT A MINUTE ...

... did we just ... add a new feature not hooking existing methods ... ?

HOOK EXISTING CODE ...

&

... AND INJECT NEW CODE?

"PORT" SOME EXISTING TOOLING ...

- iOS SSL "Unpinner" (nabla-c0d3/ssl-kill-switch2)
- iOS Keychain Dumper (ptoomey3/Keychain-Dumper)
- Jailbreak "Simulator"
- Biometrics Authentication (LocalAuthentication) Bypass
- ... and more

... AND WRITE SOME NEW ONES !

- Binary plist dumper
- iOS Shared cookie storage (NSHTTPCookieStorage) enumerator
- Fully featured file manager to upload & download files
- Inline sqlite3 interface

INSTEAD OF HAVING LOTS OF SCRIPTS LYING AROUND, LETS BUNDLE THEM

ORIGINAL DESIGN

- CLI using python-prompt-toolkit (just like frida-tools uses)
- Reuse parts of JavaScript snippets by implementing them as Jinja2 templates
- Scripts would be single shot (by default); (aka: create session -> **load()** -> send() -> **unload()**)
- Longer running jobs would just be scripts that don't unload unless asked to

TYPICAL TOOL WRITING / PORTING PROCESS

```
51
52      NSString *t = [[NSBundle mainBundle] bundlePath];
53      NSString *infoPlist = [t stringByAppendingString:@"/Info.plist"];
54      NSLog(@"%@", infoPlist);
55      NSMutableDictionary *plistData = [[NSMutableDictionary alloc] initWithContentsOfFile:infoPlist];
56
57      NSLog(@"Plist: %@", plistData);
58
59      return YES;
60  }
```

PewPew

```
[Frida INFO] Listening on 127.0.0.1 TCP port 27042
2019-05-27 22:29:31.182252+0100 PewPew[79767:12560626] libMobileGestalt MobileGestalt.c:890: MGIs
2019-05-27 22:29:31.284848+0100 PewPew[79767:12560626] We are up scotty!
2019-05-27 22:29:31.284969+0100 PewPew[79767:12560626]
    /Users/leonjza/Library/Developer/CoreSimulator/Devices/08F85C2C-D009-4AB6-BA01-61E6C2B0CB67/d
    B9BE-C04D73825A46/PewPew.app/Info.plist
2019-05-27 22:29:31.285483+0100 PewPew[79767:12560626] Plist: {
    BuildMachineOSBuild = 18E226;
    CFBundleDevelopmentRegion = en;
    CFBundleDisplayName = "i-pewpew";
    CFBundleExecutable = PewPew;
    CFBundleIcons =     {
        CFBundlePrimaryIcon =         {
            CFBundleIconFiles =             (
                AppIcon40x40
            );
            CFBundleIconName = AppIcon;
        };
};
```

```
[Remote::Gadget]-> ObjC.classes.NSMutableDictionary.alloc().initWithContentsOfFile_('/Users/
4AB6-BA01-61E6C2B0CB67/data/Containers/Bundle/Application/B89B61E7-C21E-4E85-B9BE-C04D73825A
"{
    BuildMachineOSBuild = 18E226;
    CFBundleDevelopmentRegion = en;
    CFBundleDisplayName = "i-pewpew";
    CFBundleExecutable = PewPew;
    CFBundleIcons =     {
        CFBundlePrimaryIcon =         {
            CFBundleIconFiles =             (
                AppIcon40x40
            );
            CFBundleIconName = AppIcon;
        };
};
```

```python
runner = FridaRunner()
runner.set_hook_with_data(ios_hook('plist/get'), plist=plist)
runner.run()

response = runner.get_last_message()

if not response.is_successful():
    click.secho('Failed to get plist with error: {0}'.format(response.error_reason), fg='red')
    return

click.secho(response.data, bold=True)
```

ADD HELPERS FOR COMMON ANALYSIS TASKS

- Dump method arguments
- Dump method return values
- Dump stack traces for when entering a method
- Generic method return value overrider. (for BOOL cases)

AND AN APPLICATION PATCHER*

- Add the ability to patch a decrypted iOS application with a Frida gadget in a single command.

* App needs to already be decrypted

```
sensepost's iPad on (iPad: 10.2.1) [usb] # ls
Read Access
No Write Access
Type                 Perms  Read   Write  Owner          Group          Size  Creation                  Name
-------------------  -----  -----  -----  -------------  -------------  ----  ------------------------  ----------------------
NSFileTypeRegular      420  True   False  _installd (33) _installd (33)  8063  2017-07-05 14:29:34 +0000  AppIcon40x40@2x.png
NSFileTypeRegular      420  True   False  _installd (33) _installd (33) 191955  2017-07-05 14:29:35 +0000  Assets.car
NSFileTypeDirectory    493  True   False  _installd (33) _installd (33)   102  1970-01-01 00:00:00 +0000  Base.lproj
NSFileTypeDirectory    493  True   False  _installd (33) _installd (33)   102  1970-01-01 00:00:00 +0000  Frameworks
NSFileTypeRegular      420  True   False  _installd (33) _installd (33)  1386  2017-07-05 14:29:35 +0000  Info.plist
NSFileTypeDirectory    493  True   False  _installd (33) _installd (33)    68  1970-01-01 00:00:00 +0000  META-INF
NSFileTypeRegular      493  True   False  _installd (33) _installd (33) 936656  2017-07-06 16:32:37 +0000  PewPew
NSFileTypeRegular      420  True   False  _installd (33) _installd (33)     8  2017-07-05 14:29:35 +0000  PkgInfo
NSFileTypeDirectory    493  True   False  _installd (33) _installd (33)   102  1970-01-01 00:00:00 +0000  _CodeSignature
NSFileTypeRegular      420  True   False  _installd (33) _installd (33)  9498  2017-07-06 16:32:37 +0000  embedded.mobileprovision
NSFileTypeRegular      420  True   False  _installd (33) _installd (33)  1673  2017-07-04 19:09:42 +0000  swapi.co.der
sensepost's iPad on (iPad: 10.2.1) [usb] #
```

JULY 2017

OBJECTION 0.0.1 GOES PUBLIC


android studio
Powered by the IntelliJ® Platform

AUGUST 2017

OBJECTION 1.0.0 GOES PUBLIC WITH ANDROID SUPPORT

MAJOR OS VERSIONS NO LONGER BOTHER US, WE CAN STILL HACK APPS.

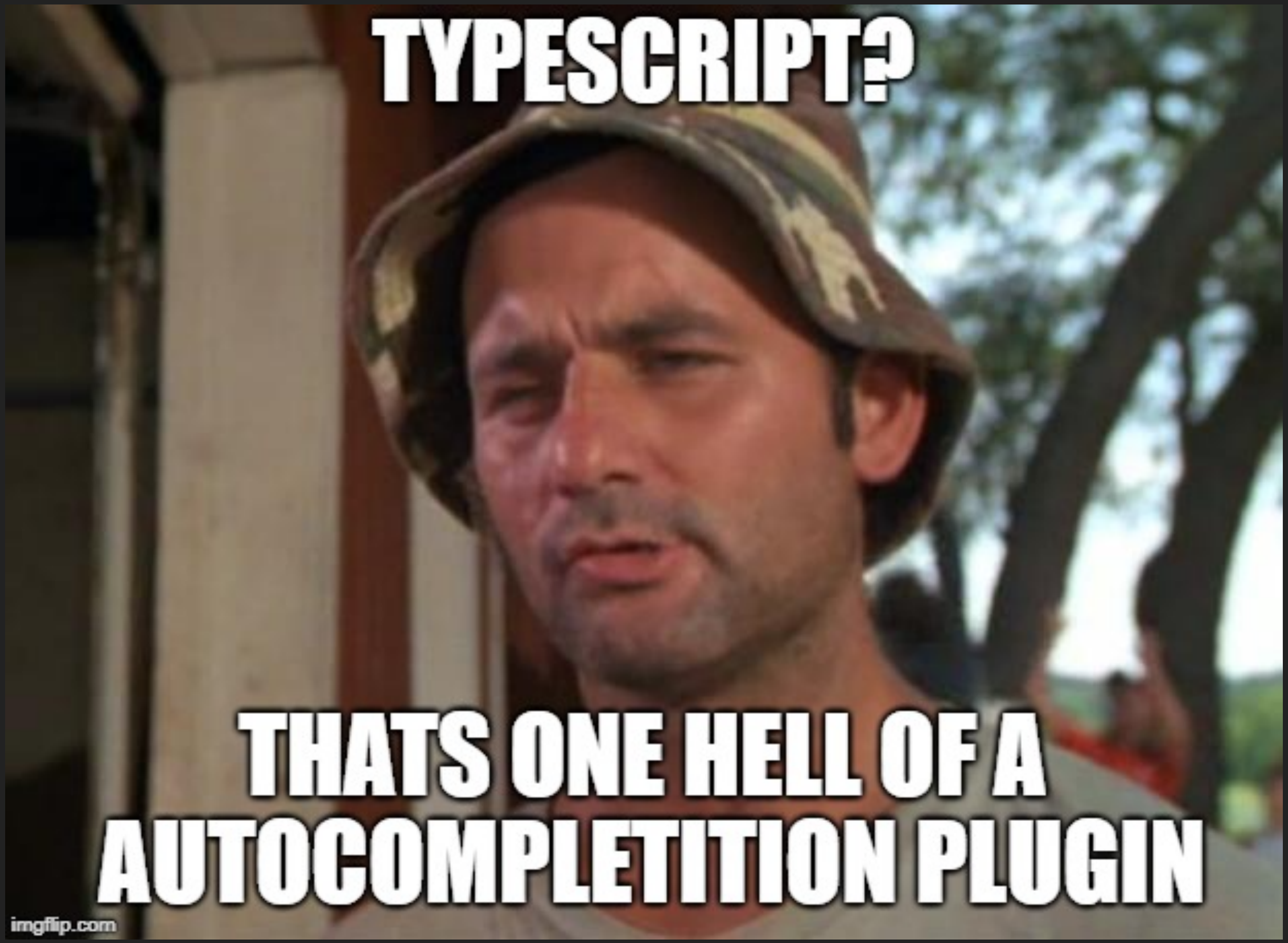
THERE WERE ... BUGS.

```
~ » node
Welcome to Node.js v12.3.1.
Type ".help" for more information.
> 3 - 1
2
> 3 + 1
4
> '3' - 1
2
> '3' + 1
'31'
>
```

Also, JavaScript ... >:|

ISSUES

- The load() -> unload() loop was using ~80 separate scripts (~2300 ES5 JavaScript)
- Some bugs were pretty silly, typical 2am dev stuff :/
- Even though Jinja2 offered some snippet reuse, it was not great
- Importing NPM modules... lol.


TYPESCRIPT?
THATS ONE HELL OF A AUTOCOMPLETITION PLUGIN
imgflip.com

FRIDA-COMPILE && TYPESCRIPT

- Project skeleton exists letting you compile an agent to ES5 compatible JavaScript
- frida-gum-types ! (<3 with VSCode)
- Opens up the NPM ecosystem to our agent
- TypeScript ! (Static Typing, saving my sanity)

```
const demo = (a: number, b: number): number => {
  return "3" + a + b;
};
```

```
25
26  const foo = Module.ge
27
28  export namespace iosk
29
30    // clean out the ke
31    export const empty = (): void => {
32    const searchDictionary: NSMutableDictionaryType = NSMuta
33    itemClasses.forEach((clazz) => {
34
35      // set the class-type we are querying for now & delete
36      searchDictionary.setObject_forKey_(clazz, kSec.kSecClass);
37      libObjc.SecItemDelete(searchDictionary);
38    });
```

```
getBaseAddress
getExportByName
arguments
length
```

```
(method) Module.getBaseAddress(nam
e: string): NativePointer

Looks up the base address of the name
module. Throws an exception if the module isn't
loaded.

@param name — Module name or path.
```

ARCHITECTURE CHANGES FOR AN AGENT

- Move to Frida RPC Usage only (no more send())
- Job "tracking" (long running hooks) needed to move from Python to JavaScript
- Single agent, with a single load() and unload() on exit.

```
1  import { ping } from "./generic/ping";
2  import { android } from "./rpc/android";
3  import { env } from "./rpc/environment";
4  import { ios } from "./rpc/ios";
5  import { jobs } from "./rpc/jobs";
6  import { memory } from "./rpc/memory";
7
8  rpc.exports = {
9    ...android,
10   ...ios,
11   ...env,
12   ...jobs,
13   ...memory,
14   ping: (): boolean => ping(),
15 };
16
```

FEBRUARY 2019

OBJECTION 1.5.0 RELEASED

FEATURES TODAY (V1.6.3)

- Still does not require a rooted/jailbroken[1] device
- Many SSL pinning bypasses for both iOS & Android
- Fast runtime analysis helpers and manipulators
- Community contributed plugin loader for private hooks [2]
- HTTP API to call any agent RPC method

[1] but decrypted iOS app
[2] By SpeedyFireCyclone @ Github

UNDER THE HOOD

KEYCHAIN DUMPER

```
NSMutableDictionary *dict = [self prepareDict:key];
[dict setObject:(__bridge id)kSecMatchLimitOne forKey:(__bridge id)kSecMatchLimit];
[dict setObject:(id)kCFBooleanTrue forKey:(__bridge id)kSecReturnData];
[dict removeObjectForKey:(__bridge id)kSecAttrAccessible];

CFTypeRef result = NULL;
OSStatus status = SecItemCopyMatching((__bridge CFDictionaryRef)dict, &result);
```

KEYCHAIN DUMPER

```
// the base query dictionary to use for the keychain lookups
const searchDictionary: NSMutableDictionaryType = NSMutableDictionary.alloc().init();
searchDictionary.setObject_forKey_(kCFBooleanTrue, kSec.kSecReturnAttributes);
searchDictionary.setObject_forKey_(kCFBooleanTrue, kSec.kSecReturnData);
searchDictionary.setObject_forKey_(kCFBooleanTrue, kSec.kSecReturnRef);
searchDictionary.setObject_forKey_(kSec.kSecMatchLimitAll, kSec.kSecMatchLimit);

return [].concat.apply([], itemClasses.map((clazz) => {

  const clazzItems: IKeychainItem[] = [];
  searchDictionary.setObject_forKey_(clazz, kSec.kSecClass);

  // prepare a pointer for the results and call SecItemCopyMatching to get them
  const resultsPointer: NativePointer = Memory.alloc(Process.pointerSize);
  const copyResult: NativePointer = libObjc.SecItemCopyMatching(searchDictionary, resultsPointer);

  // without results (aka non-zero OSStatus) we just move along.
  if (!copyResult.isNull()) { return; }

  // read the resultant dict of the lookup from memory
  const searchResults: NSDictionary = new ObjC.Object(resultsPointer.readPointer());
```

ACL parsing contributed by @_dotvader

JAILBREAK "SIMULATOR"

```
const jailbreakPaths = [
  "/Applications/Cydia.app",
  "/Applications/FakeCarrier.
  "/Applications/Icy.app",
  "/Applications/IntelliScree
  "/Applications/MxTube.app",
  "/Applications/RockApp.app"
  "/Applications/SBSetttings.
```

```
return Interceptor.attach(
  ObjC.classes.NSFileManager["- fileExistsAtPath:"].implementation, {
    onEnter(args) {…
    },
    onLeave(retval) {

      // stop if we dont care about the path
      if (!this.is_common_path) {
        return;
      }

      // depending on the desired state, we flip retval
      switch (success) {
        case (true):
          // ignore successful lookups
          if (!retval.isNull()) {
            return;
          }
          send(
            c.blackBright(`[${ident}] `) + `fileExistsAtPath: check for ` +
            c.green(this.path) + ` failed with: ` +
            c.red(retval.toString()) + `, marking it as successful.`,
          );

          retval.replace(new NativePointer(0x01));
          break;
```

TRICKIER HOOKS — BIOMETRICS BYPASS & OBJC BLOCK

```
- (void)evaluatePolicy:(LAPolicy)policy
       localizedReason:(NSString *)localizedReason
                 reply:(void (^)(BOOL success, NSError *error))reply;
```

https://developer.apple.com/documentation/localauthentication/lacontext/1514176-evaluatepolicy?language=objc

TRICKIER HOOKS — BIOMETRICS SAMPLE IMPLEMENTATION

```
[myContext evaluatePolicy:LAPolicyDeviceOwnerAuthentication
          localizedReason:myLocalizedReasonString
                    reply:^(BOOL success, NSError *error) {
        if (success) {

            dispatch_async(dispatch_get_main_queue(), ^{
                    [self performSegueWithIdentifier:@"LocalAuthSuccess" sender:nil];
            });

        } else {


            dispatch_async(dispatch_get_main_queue(), ^{
                UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Error"
                                                                    message:error.desc
                                                                   delegate:self
                                                          cancelButtonTitle:@"OK"
                                                          otherButtonTitles:nil, nil];
                [alertView show];
            });
        }
}];
```

TRICKIER HOOKS — BIOMETRICS FRIDA BYPASS

```
const lacontext: InvocationListener = Interceptor.attach(
  ObjC.classes.LAContext["- evaluatePolicy:localizedReason:reply:"].implementation, {
    onEnter(args) {

      const originalBlock = new ObjC.Block(args[4]);
      const savedReplyBlock = originalBlock.implementation;

      originalBlock.implementation = (success, error) => {
        if (!success === true) {
          // Change the success response from the OS to true
          success = true;
        }

        savedReplyBlock(success, error);
      };
    },
});
```

AN API IN 4 LOC

```
# invoke the method based on the http request type
if request.method == 'POST':
    response = getattr(rpc, method)(*post_data.values())

if request.method == 'GET':
    response = getattr(rpc, method)()
```

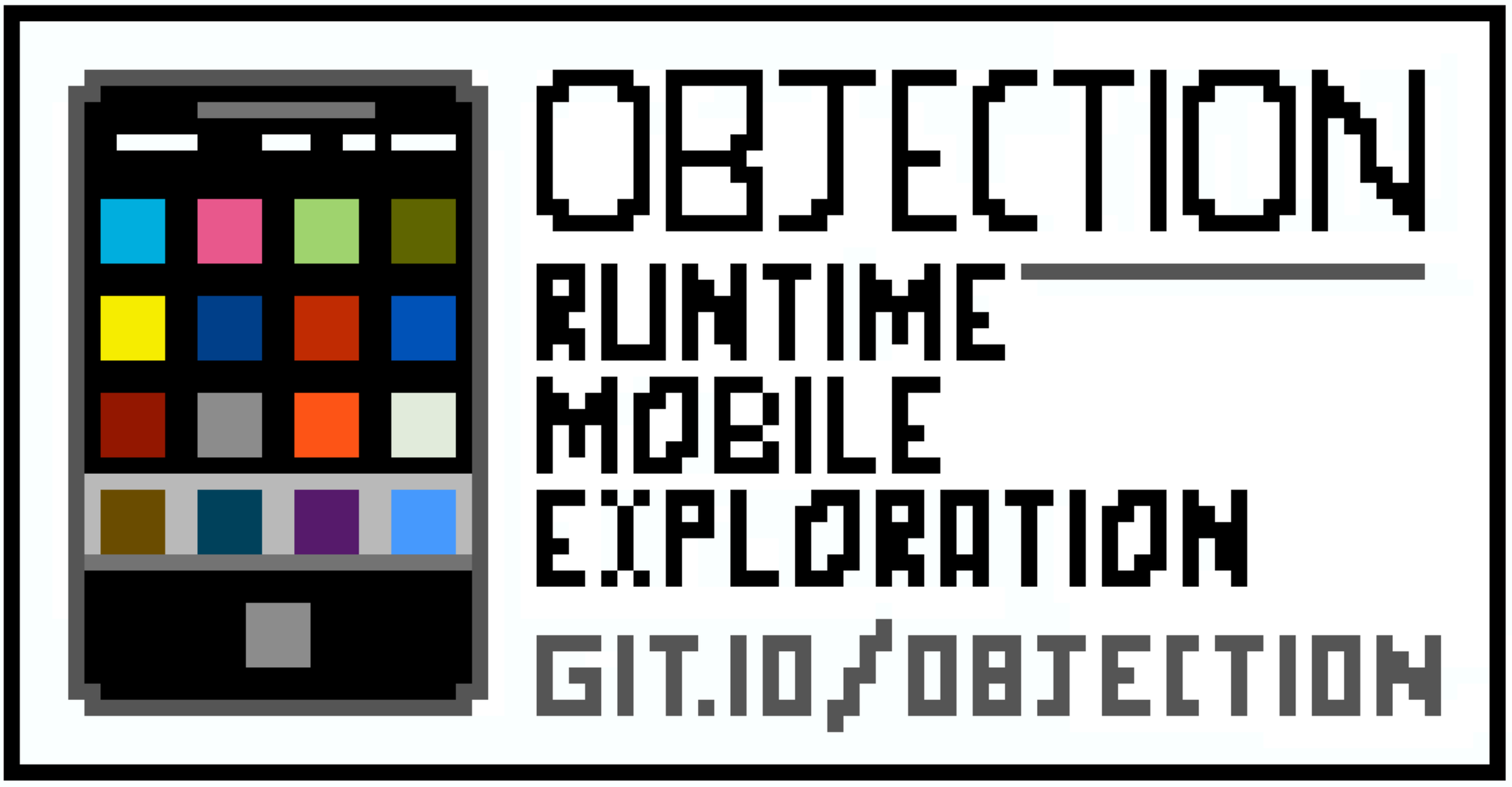
```
~ » curl -s -H "Content-Type: text/javascript" http://127.0.0.1:8888/rpc/invoke/iosBundlesGetFrameworks | jq
[
  {
    "bundle": "com.apple.CorePhoneNumbers",
    "executable": "CorePhoneNumbers",
    "path": "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/CoreSim
nts/Resources/RuntimeRoot/System/Library/PrivateFrameworks/CorePhoneNumbers.framework",
    "version": "1.0"
  },
  {
    "bundle": "com.apple.LocalAuthentication",
    "executable": "LocalAuthentication",
    "path": "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/CoreSim
nts/Resources/RuntimeRoot/System/Library/Frameworks/LocalAuthentication.framework",
    "version": "1.0"
  },
  {
    "bundle": "com.apple.CoreAuthentication.SharedUtils",
    "executable": "SharedUtils",
    "path": "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/CoreSim
nts/Resources/RuntimeRoot/System/Library/Frameworks/LocalAuthentication.framework/Support/SharedUtils.framewo
    "version": "1.0"
  },
```

DEMO

CLOSING THOUGHTS

- Realise that jailbreak free pentesting is possible
- We can test apps on the latest mobile operating systems
- Encourage you to play more with Frida!

Questions?



@leonjza | git.io/objecton