

(Game) Hacking with Frida.re

Martin Wagner – 07.12.2023

How do CTF exploits usually look like?

```
1 from pwn import *
2 r = remote('ctf.kitctf.de', 1337)
3
4 # ...
5
6 r.sendline(b'A' * cyclic_find('kaan') + rop.chain())
7 r.interactive()
```

What if the target is more complex

- (Multiplayer) games
 - Server-side messages we need to respond to
 - Traffic encryption & obfuscation
- Messenger applications
 - Whatsapp, iMessage, Zoom
 - E2E encryption, custom protocols
- Mobile apps
 - Hard to debug using regular debuggers
 - May only run on physical devices

Frida to the rescue

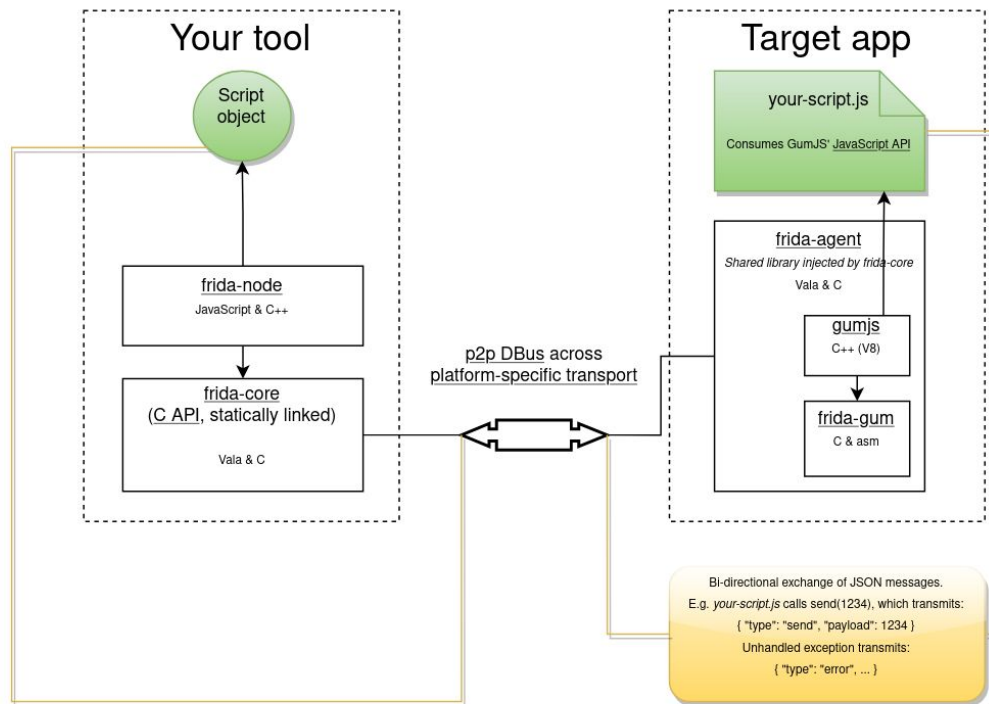
- Dynamic analysis + easy patching
- Instrument the real client for a service
 - Abuse communication routines
- Run our exploit script directly on a device
 - eg. Whatsapp exploit runs within a modified Whatsapp client
- Just for exploits?
 - No, also useful for dynamic analysis

What is Frida?

“Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers.”

- A framework for reverse-engineering tools
- A debugger with a really nice typescript API
- A way to inject custom javascript “agents” into processes
 - Greasemonkey for native apps
- Supports various platforms
 - Windows, macOS, Linux, iOS, watchOS, tvOS, Android, FreeBSD, and QNX

How does it work?



How does it look like?

```
1 const base = Module.getBaseAddress('craft')
2 const yPos = base.add(0x273c24)
3 const rot = yPos.add(12)
4
5 Interceptor.attach(Module.getExportByName(null, 'handle_movement'), {
6   onEnter() {
7     this.preY = yPos.readFloat()
8   },
9   onLeave() {
10    if (flying) {
11      yPos.writeFloat(this.preY + rot.readFloat() * .2)
12    }
13  }
14 })
```

Use Case – Mobile security

- Simple real-world problem: certificate pinning in Android
 - < 10 line script - 90% of my frida usage

```
1 Java.perform(function() {
2     var array_list = Java.use("java.util.ArrayList");
3     var ApiClient = Java.use('com.android.org.conscrypt.TrustManagerImpl');
4
5     ApiClient.checkTrustedRecursive.implementation = function(a1, a2, a3, a4, a5, a6) {
6         var k = array_list.$new();
7         return k;
8     }
9 }, 0);
```


Hooking android crypto APIs

- `com.android.org.conscrypt.TrustManagerImpl.checkTrustedRecursive`
 - Recursively build certificate chains until a valid chain is found or all possible paths are exhausted.
 - return the entire valid chain starting with the leaf certificate. This is the concatenation of `untrustedChain` and `trustAnchorChain`.
 - throws `CertificateException` If no valid chain could be constructed. [...]

<https://github.com/google/conscrypt/blob/master/common/src/main/java/org/conscrypt/TrustManagerImpl.java#L521>

Debugging on a phone???

```
$ adb push frida-server-* /data/local/tmp/frida-server  
$ adb shell /data/local/tmp/frida-server &  
$ frida --codeshare sowdust/universal-android-ssl-pinning-bypass-2 -U -F --no-pause
```

Use Case – Mobile security

- Inject code into real world application
 - Stdlib provides access to high level languages like Java, Swift, Objective-C
- Easy to run frida on mobile devices
 - Root is helpful but often not needed
 - Emulators work as well
- Hooking a single function in a complex lib is something frida is perfect for

Demo – Game Hacking

- “Craft” challenges from KITCTF internal CTF 2022 - by ju256
 - Source was provided during CTF, we ignore this for now
- Networking
 - Analyzing the network traffic of a game is always a good place to start. So, here you go...
- Broken Wings
 - If you get them back you should checkout the clouds. I'm pretty sure there is something hidden there.
- Treasury
 - Use the teleport in the tower to get to the treasury.

Demo – Game Hacking

The screenshot displays a game hacking demonstration. On the left, a Minecraft game window is visible, showing a player character standing in front of a large, grey, brick-like structure. The game's console output shows the player's coordinates and a welcome message. On the right, a Frida JavaScript console window is open, displaying the code for a network traffic interceptor. The code is designed to intercept and parse incoming network data, specifically looking for a flag. The console output shows the interceptor successfully parsing a message and logging the flag.

```
steam-run ~/client.sh Craft
FP, 0, (0.00, 10.75, 0.00) [1, 441, 503768] 9am 60fps
Welcome to Craft! (Credit: https://github.com/fogleman/Craft)
Visit kitcraft.me for challenge descriptions!
Type "/help" for a list of commands.
guest4 has joined the game.

frida -l _agent.js craft
0x3e1ef : send(P,0.00,16.75,0.00,2.59,0.01)
0x3e1ef : send(P,0.00,16.75,0.00,2.59,-0.29)
0x3e1ef : send(P,0.00,16.75,0.00,2.57,-0.39)
0x3e1ef : send(P,0.00,16.75,0.00,2.51,-0.33)
0x3e1ef : send(P,0.00,16.75,0.00,2.40,-0.17)
0x3e1ef : send(P,0.00,16.75,0.00,2.35,-0.15)
0x3e1ef : send(P,0.00,16.75,0.00,2.15,-0.10)
0x3e1ef : send(P,0.00,16.75,0.00,1.72,0.01)
0x3e1ef : send(P,0.00,16.75,0.00,1.68,0.03)
```

```
10   log(`${this.ret}` : send(`${this.buffer.readUtf8String().trim`
11   });
12   };
13   };
14   let flag:string = ''
15   };
16   Interceptor.attach(Module.getExportByName(moduleName:null, exportN
17   onEnter(args:InvocationArguments):void {
18     this.ret = this.returnAddress.sub(base)
19     this.buffer = args[0]
20   };
21   const msg = this.buffer.readCString().toString().trim()
22   };
23   const newFlag = msg.split(separator:'\n')
24     .filter((line:string) => line.startsWith('F'))
25     .map((l:string) => l.substring(1))
26     .join('')
27   };
28   if (newFlag.length > 0) {
29     flag += newFlag
30     log(`Flag: ${flag}`)
31   }
32   };
33   });
34   }
```

5:34:55 PM - File change detected. Starting incremental compilation...
5:34:55 PM - Found 0 errors. Watching for file changes.

Demo – Recap

- **Interceptor API**
 - Attach: breakpoints / logpoints
 - Replace: replace function with our own implementation
- **Memory & Module APIs**
 - Get runtime offsets
 - Read & write arbitrary memory
- **NativeFunctions**
 - Call existing native functions from our agent
- **Stalker**
 - Trace applications

What else is there to know?

- CModules
 - If we have a hot function in our agent we can just throw some inline C at it
- Modes of operations
 - Frida gadget can be LD_PRELOADED to run from the start of the program
- For Unity challenges: il2cpp offsets
 - Unity can compile .net IL to cpp
 - Tools can extract method names + offsets
 - Perfect starting point for some Frida hooks
- Frida for fuzzing
 - AFL++ can use frida instead of qemu for fuzzing
 - Some tools exist for in-process fuzzing, attaching a fuzzer to a running application
- Control scripts
 - Manage how your agent runs using a script outside of the target process

Hands on challenges

- Gamehacking
 - Craft challenges
 - CSRunner (unity, offline) from Rumble 2021
- Android
 - Click me from NahamCon 2022
 - Secure Notes, same event
- Links in Slack
 - <https://mawalabs.de/stuff/csranner.tar.gz>
 - <https://mawalabs.de/stuff/frida.tar.gz>

Learning resources

- <https://learnfrida.info/>
- <https://frida.re/docs/javascript-api/>
- <https://github.com/dweinstein/awesome-frida>
 - Lots of awesome tools. Mileage may vary but often a good source for example code.

How can I use it

- Install Frida
- Write agent script
- Attach to process
 - Can be on external device or run locally
- ???
- Profit
 - Knowledge
 - Working exploits
 - 100.000 HP