Data sources used by spam call blocking applications

1. Introduction

Spam calls are a common irritation for many people. According to Hiya [1], on average a person receives **16 scam calls a month.** After the proliferation of mobile phones in the past decade, it is often seen as a personal device with the phone number often only being known to parties involved in the life of the person. It is hard to imagine that an incoming spam call will not be answered. A scam call does not bring any value to the callee, even the opposite, it may be a source of anxiety and expenses. What is even more frightening, **in 75% of cases, scam callers knew some personal information about their victims** [2].

2. Research question

How and where does each application store the information it uses to determine whether an incoming call can be classified as spam?

3. Methods and tools

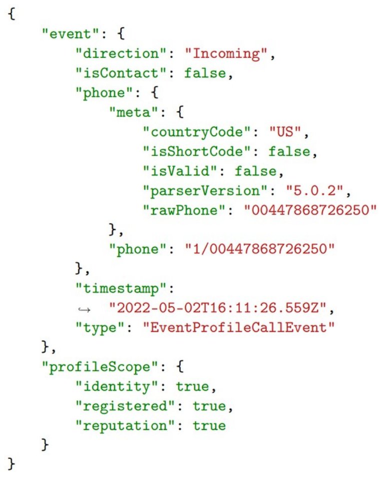
Tools used:

- Apktool
- Dex2jar
- Jadx
- Android Studio
- ADB
- SqliteBrowser
- MITMproxy
- Frida

Research methods:

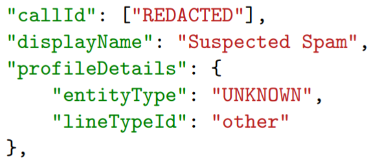
- Static analysis

- Dynamic analysis



```
{
    "event": {
        "direction": "Incoming",
        "isContact": false,
        "phone": {
            "meta": {
                "countryCode": "US",
                "isShortCode": false,
                "isValid": false,
                "parserVersion": "5.0.2",
                "rawPhone": "00447868726250"
            },
            "phone": "1/00447868726250"
        },
        "timestamp":
        ↪  "2022-05-02T16:11:26.559Z",
        "type": "EventProfileCallEvent"
    },
    "profileScope": {
        "identity": true,
        "registered": true,
        "reputation": true
    }
}
```

Figure 1: Network request for a malicious number made by Hiya



```
"callId": ["REDACTED"],
"displayName": "Suspected Spam",
"profileDetails": {
    "entityType": "UNKNOWN",
    "lineTypeId": "other"
},
```

Figure 2: Hiya's response for the malicious phone number

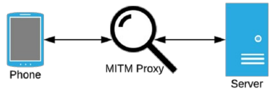
Author: Victor de Jong

Supervisors:
Dr. Apostolis Zarras
Dr. Yury Zhauniarovich

4. Setup

Using **Apktool**, **Dex2Jar** and **Jadx** source code is decompiled into readable **Java** code or **Smali**, which can be compiled back into an installable APK.

Using **Android Studio**, **MITMproxy** and **Frida**, we have a modified Android emulator which routes all network traffic through a proxy which allows us to **inspect all traffic** and access to the filesystem to **observe stored data**.



Terminology

Static (code) analysis, also called static code analysis, is a method of computer program debugging that is done by examining the code without executing the program.

Dynamic (code) analysis is the analysis of computer software that is performed by executing programs on a real or virtual processor.

Certificate pinning is a security mechanism where the certificate of the connection is verified independently of the operating system's certificate store.

6. Conclusion

There are large differences between applications and the approach they take. The amount of **offline data available** varies between **582** phone entries, to more than **130 000**.

Several applications use **online** resources to get up-to-date information about incoming phone numbers.

Some even use **both** online and offline methods in a hybrid setup.

5. Results

Application	On disk	Internet
BelControl	✓	✓
CallApp Contacts	✓	✗
CallBlocker	✓	✗
CallerID Block	✓	✓
Hiya	✗	✓
Should I answer?	✓	✗
ShowCaller	✓	✓
StopMijBellen	✓	✗
TelGuarder	✗	✓
TrueCaller	✓	✓

Figure 3: Applications and their means of storage of detection data

7. References

[1] Hiya, "State of the Phone Call: Half Yearly Report 2019" (2019), https://assets.hiya.com/public/pdf/HiyaStateOfTheCall2019H1.pdf?v=6b7b682837c56c47656c012c1da0e6a0

[2] First Orion, "Scam Call Trends and Projections Report", Summer 2019, http://firstorion.com/wp-content/uploads/2019/07/First-Orion-Scam-Trends-Report_Summer-2019.pdf

23-06-2022